# Surrogate models for particle and power exhaust in divertor tokamak simulations

Inaugural-Dissertation

zur Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf

vorgelegt von

**Stefan Dasbach**
aus Leverkusen

Düsseldorf, August 2024

aus dem Institute of Fusion Energy and Nuclear Waste Management
des Forschungszentrums Jülich

Gedruckt mit der Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Berichterstatter:

1. Prof. Dr. Yunfeng Liang

2. Prof. Dr. Sebastijan Brezinsek

Tag der mündlichen Prüfung:

## Eidesstattliche Versicherung

Ich versichere an Eides Statt, dass die Dissertation von mir selbständig und ohne unzulässige fremde Hilfe unter Beachtung der „Grundsätze zur Sicherung guter wissenschaftlicher Praxis an der Heinrich-Heine-Universität Düsseldorf“ erstellt worden ist.

Stefan Dasbach

Ort, Datum

The work presented in this thesis is in parts based on the following publications:

**Towards fast surrogate models for interpolation of tokamak edge plasmas**
Stefan Dasbach and Sven Wiesen
Published in Nuclear Materials and Energy, [1]
The author of this thesis conducted the simulations and data analysis as well as the training, optimization and evaluation of the machine learning models shown in the publication. This includes the creation of all figures and writing of the manuscript. The results presented in the publication were obtained on an intermediate version of the dataset, while the corresponding Sections 6, 8.2, 8.4 in this thesis present an updated version on the final dataset with an extended scope and more machine learning models.

**Data-driven models in fusion exhaust: AI methods and perspectives**
Sven Wiesen, Stefan Dasbach, Adam Kit, Aaro E. Järvinen, Andreas Gillgren, Aaron Ho, Alex Panera, Dirk Reiser, Martin Brenzke, Yoeri Poels, Egbert Westerhof, Vlado Menkovski, Gijs F. Derks, Par Strand
Published in Nuclear Fusion, [2]
Mainly contents from Section 2 of the publication contribute to this thesis. The author of this thesis conducted the simulations and data analysis as well as the training, optimization and evaluation of the machine learning models shown in that section. This includes the creation of all figures and writing of the manuscript.
Also parts of the introduction and conclusion of the publication contribute to the introduction and conclusion in this thesis. All authors contributed to the writing of these sections in the publication.

**Surrogate models for divertor tokamak simulations**
Stefan Dasbach and Sven Wiesen
Prepared for submission to Nuclear Fusion
The author of this thesis conducted the simulations and data analysis as well as the training, optimization and evaluation of the machine learning models shown in the publication. This includes the creation of all figures and writing of the manuscript.

Both text and figures from these publications are used either in their original or adapted form throughout this thesis without explicit citation.

# Abstract

The heat exhaust constitutes one of the most critical operational limits in tokamak fusion reactors. Unmitigated, the expected heat fluxes in future reactors will exceed what is sustainable for known materials. The scrape-off layer connects the plasma and reactor components. It plays a crucial role for limiting the heat fluxes to reactor components while maintaining desirable plasma conditions in the confined region. Accurate models of the scrape-off layer are required for the design and during the operation of tokamak reactors.

Simplified analytical descriptions of the scrape-off layer lack the required predictive capabilities due to a lack in the included physical processes. This often necessitates a calibration against experimental measurements, which introduces uncertainties for the planning of future larger reactors. While more complex simulations include all necessary physical processes, these simulations are computationally expensive, difficult to operate and suffer from numerical instabilities. This prevents their application in rapid design studies, algorithmic optimization or integrated modeling. A potential remedy comes in using machine learning models trained on simulations for fast and easy to use predictions. This thesis is devoted to provide a proof-of-concept of such a surrogate model and to provide recommendations for the methods to construct it.

To this end, a large dataset of several thousand SOLPS-ITER simulations using a reduced fidelity fluid neutral gas description was created. The dataset uses a conformal size scaling to encompass cross-machine scenarios across an eight dimensional parameter space. Besides the reactor size, the varied simulation parameters include the input power, deuterium throughput, impurity seeding rate and strength of anomalous cross-field transport. Analysis shows that the reduced fidelity procedure introduced some simulation artifacts, but all expected physical regimes and trends are recovered.

Based on this dataset, a variation of machine learning models with differing architectures and scopes are tested. Among these are different variants of Gradient Boosted Regression Trees and fully connected feedforward neural networks. The evaluation shows that the neural networks are the most accurate. Further, it has no benefit to develop models for specific parts of the scrape-off layer, but the conditions in the whole domain can be predicted at once. It is easier to achieve high accuracy by employing independent models for different observables rather than by using a combined model. For the tested models it is difficult to preserve the small temperature gradients in low density regimes. This leads to drastic errors in heat fluxes deduced from the surrogate predictions. Using independent models the heat fluxes can be predicted accurately but the predictions are then not consistent with the other plasma properties. So far it was not possible to create a model that predicts accurate heat fluxes self-consistent with all plasma quantities in all regimes. Analysis shows that the surrogate model accuracy drops drastically when less than 1000 training simulations are available.

Using the developed surrogate model some potential applications are demonstrated and the impurity concentrations necessary to achieve detachment are predicted for multiple tokamaks. These predictions show similar functional relations as previous scaling laws.

Finally, a small dataset of higher fidelity ITER simulations is used to train surrogate models. The smaller scope of the dataset allows for achieving much more accurate predictions. Further analysis shows that transfer learning from the previous surrogate model has no benefits over training a new model from scratch. But due to the the small number of high fidelity test simulations, no final evaluation is possible. Therefore, future efforts should focus on discovering the potential and the methods for models utilizing simulations with mixtures of fidelity.

## Zusammenfassung

Die Abwärme stellt eine der kritischsten Betriebsgrenzen in Tokamak-Fusionsreaktoren dar. Ungehemmt werden die zu erwartenden Wärmeströme in zukünftigen Reaktoren das für bekannte Materialien tragbare Maß überschreiten. Die Plasmarandschicht verbindet das Plasma und die Reaktorkomponenten. Sie spielt daher eine entscheidende Rolle die Wärmeströme zu den Reaktorkomponenten zu begrenzen und gleichzeitig die gewünschten Plasmabedingungen im eingeschlossenen Bereich aufrechtzuerhalten. Genaue Modellvorhersagen der Plasmarandschicht sind erforderlich für die Konstruktion und den erfolgreichen Betrieb von Tokamak-Reaktoren.

Vereinfachte analytische Beschreibungen der Plasmarandschicht haben aufgrund fehlender physikalischer Prozesse nicht die erforderlichen Vorhersagefähigkeiten. Dies macht häufig eine Kalibrierung anhand von experimentellen Messungen erforderlich. Dies Kalibrierung führt zu Unsicherheiten hinsichtlich der Anwendbarkeit für die Planung künftiger größerer Reaktoren. Komplexere Simulationen enthalten zwar alle erforderlichen physikalischen Prozesse, sind aber rechenintensiv, schwierig zu bedienen und leiden unter numerischen Instabilitäten. Dies verhindert ihre Anwendung für schnelle Designstudien, algorithmische Optimierung oder integrierte Modellierung. Eine mögliche Lösung besteht in der Verwendung von Modellen des maschinellen Lernens, die auf Simulationen trainiert werden, um schnelle und einfach zu handhabende Vorhersagen zu ermöglichen. In dieser Arbeit wird ein Konzeptnachweis für ein solches Surrogatmodell erbracht und es werden Empfehlungen für Methoden zur dessen Erstellung gegeben.

Zu diesem Zweck wurde ein großer Datensatz mit mehreren tausend SOLPS-ITER-Simulationen erstellt, bei denen ein vereinfachtes Fluid Modell für das neutrale Gas wurde. Der Datensatz verwendet eine konforme Größenskalierung, um Experiment übergreifende Szenarien in einem achtdimensionalen Parameterraum zu erfassen. Zu den variierten Simulationsparametern gehören neben der Reaktorgröße auch die Eingangsleistung, der Deuteriumdurchsatz, die Stickstoffkonzentration und die Stärke des anomalen Querfeldtransports. Die Analyse zeigt, dass das vereinfachte Simulationsmodell zu einigen Simulationsartefakte führt, aber alle erwarteten physikalischen Regime und Trends reproduziert werden.

Auf der Grundlage dieses Datensatzes werden verschiedene Modelle des maschinellen Lernens mit unterschiedlichen Architekturen und Anwendungsbereichen getestet. Dazu gehören verschiedene Varianten von Regressionsbäumen und neuronalen Netzen. Die Auswertung zeigt, dass die neuronalen Netze am genauesten sind. Darüber hinaus hat es keinen Vorteil, Modelle für bestimmte Teile der Plasmarandschicht zu entwickeln, sondern die Bedingungen im gesamten Bereich können auf einmal vorhergesagt werden. Eine hohe Genauigkeit lässt sich einfacher erreichen, indem unabhängige Modelle für verschiedene Beobachtungsgrößen verwendet werden, anstatt eines kombinierten Modells. Die getesteten Modelle zeigen Schwierigkeiten, die kleinen Temperaturgradienten in Fällen mit niedriger Dichte korrekt zu erhalten. Dies führt zu drastischen Fehlern bei den Wärmeströmen, die aus den Surrogatvorhersagen abgeleitet werden. Mit unabhängigen Modellen können die Wärmeströme genau vorhergesagt werden, aber die Vorhersagen sind dann nicht mit den anderen Plasmagrößen konsistent. Bisher war es nicht möglich ein Modell zu erstellen, das genaue Wärmeströme vorhersagt, die mit allen Plasmagrößen in allen Regimen konsistent sind. Die Analyse zeigt, dass die Genauigkeit des Surrogatmodells drastisch abnimmt, wenn weniger als 1000 Simulationen zum Training zur Verfügung stehen.

Anhand des entwickelten Surrogatmodells werden einige potenzielle Anwendungen demonstriert und die erforderlichen Verunreinigungskonzentrationen für mehrere Tokamaks vorhergesagt. Diese Vorhersagen zeigen ähnliche funktionale Abhängigkeiten wie frühere Skalierungsgesetze.

Schließlich wird ein kleiner Datensatz von ITER-Simulationen mit höherer Genauigkeit verwendet, um Surrogatmodelle zu trainieren. Durch den geringeren Geltungsbereich des Datensatzes lassen sich wesentlich genauere Vorhersagen treffen. Eine weitere Analyse zeigt, dass das Transfer-Lernen aus dem vorherigen Surrogatmodell keine Vorteile gegenüber dem Training von Grund auf hat. Aber aufgrund der geringen Anzahl von Simulationen ist keine endgültige Bewertung möglich. Daher sollten sich künftige Bemühungen auf die Erforschung des Potenzials und der Methoden für Modelle konzentrieren, die Simulationen mit unterschiedlicher Genauigkeit verwerten.

# Contents

# Chapter 1

# Introduction

Human caused climate change is one of the most pressing problems of society [3]. The energy sector is one of the leading contributors to global green house gas emissions [4]. Projections indicate a further increase of the worlds energy demand in the coming decades [5], [6]. The development of nuclear fusion power plants would introduce a new source of energy with low emissions. Nuclear fusion has clear advantages compared to established forms of energy production. Nuclear fusion power plants emit no greenhouses gases [7]. Compared to renewable power sources such as wind, solar or hydro power, nuclear fusion requires much less space and can provide baseload energy production [8]. In contrast to existing (fission based) nuclear power plants, fusion power is intrinsically safe due to the absence of runaway chain reactions [8]. With the right reactor materials the amount of high-level radioactive waste can be reduced [9]–[12] and the required reacting elements as fuel for fusion are widely available [13].

For readibility this introduction provides references only for particular details and any information given without explicit citation can be found in standard textbooks (such as [13]–[16]).

## 1.1 Nuclear fusion and magnetic confinement

Two types of nuclear reactions can be distinguished: Fusion combines lighter isotopes into heavier ones, and fission splits heavier nuclei into lighter ones. The reactions are exothermic if the binding energy of the reaction results is higher than of the inputs. This corresponds to a change of mass ($E = mc^2$) in the system. Figure 1.1A shows the binding energy per nucleon for different isotopes. Especially high energy can be released from fusion reactions of the different hydrogen isotopes ($^1_1H$, $^2_1D$, $^3_1T$). Besides the individual energy gain a specific reaction is only viable for power generation if it can be produced at high enough rate. The rate of fusion reactions per volume can be expressed as $r = n_1 n_2 \langle \sigma v \rangle$. It depends on the densities of the two reactants $n_1$, $n_2$, the reaction cross-section $\sigma$ and the relative velocity of the particles $v$. The cross-section and the velocity are averaged over the distribution of particle velocities [13]. The relative particle velocities correspond to the energy of particle collisions. Figure 1.1B depicts the reaction cross-sections of different possible reactions over the energy of the collisions. It is apparent that the D-T reaction

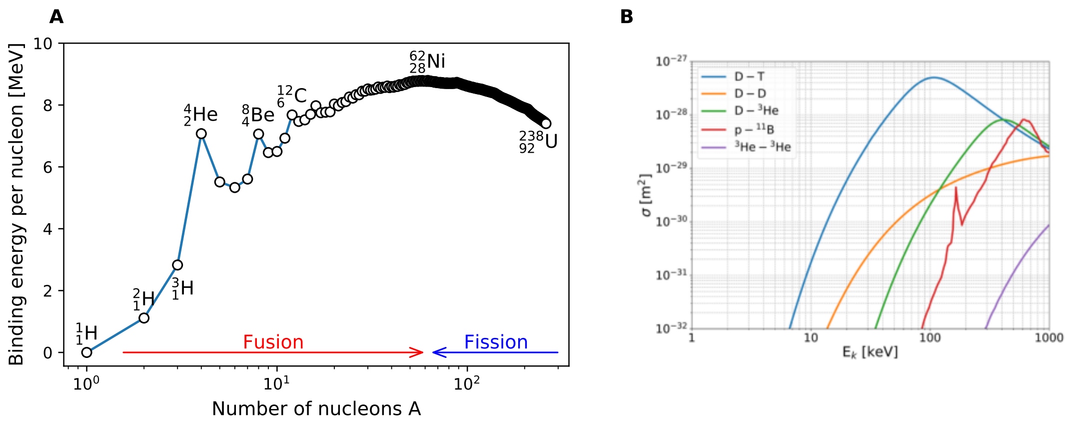


**Figure 1.1:** A: Binding energy per nucleon for different isotopes based on the data from [17], [18]. The bottom arrows highlight in which direction fusion and fission reactions release energy. B: Experimental cross-sections ($\sigma$) for the most relevant fusion reactions against the energy in the laboratory frame. Figure from [19] with data from ENDF/B-VIII.0 [20]and EXFOR [21].

of deuterium and tritium has the highest cross-section for achievable energies.

$$^2_1D + {}^3_1T \rightarrow {}^4_2He\,(3.5\,\mathrm{MeV}) + {}^1_0n\,(14.1\,\mathrm{MeV}) \tag{1.1}$$

This reaction releases an $\alpha$-particle and a neutron. Roughly 80% of the energy is carried in the latter. The neutron can not be contained in the reaction medium. Its energy will be caught as output of the reaction but can not facilitate the production of further fusion reactions. For power plant operation, the power produced in the fusion reactions needs to be larger than the power required to operate the reactor. In its most elementary form this is assessed in the power amplification factor $Q$.

$$Q = \frac{P_{fusion}}{P_{heating}} = \frac{\frac{1}{4}n^2\left<\sigma v\right>EV}{\left(\frac{3nT}{\tau_e} - \frac{1}{4}n^2\left<\sigma v\right>E_\alpha V\right)} \tag{1.2}$$

To obtain steady state conditions, the heating power needs to offset the average energy losses to the system. The energy loss can be written as the energy in the system $3nT$ divided by the energy confinement time $\tau_e$. Equation 1.2 further considers that the energy carried by the $\alpha$-particles can be contained in the system and reduces the amount of external heating required. At $Q = 1$ the external heating power has the same value as the power provided by the fusion reaction. This condition is called break even. For power generation much larger $Q$ values are required due to the energy losses in converting the fusion power to electricity and the additional power demands of the reactor components. Such aspects are considered in the engineering gain factor $Q_e$, which is the ratio of produced electricity to the electricity input [8]. The situation $Q \rightarrow \infty$ is called ignition. In this case, the heating by the $\alpha$-particles is enough to account for all energy losses, and no external heating is required. Using the D-T cross-section (for temperatures between $10 - 20\,\mathrm{keV}$), the following condition for ignition can be derived [13]:

$$nT\tau_e > 5 \cdot 10^{21}\frac{\mathrm{keVs}}{\mathrm{m}^3} \tag{1.3}$$

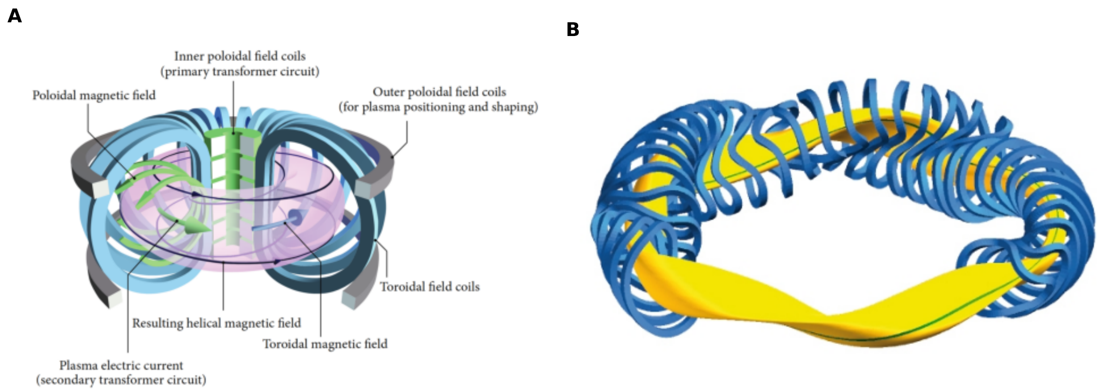


**Figure 1.2:** A: Schematic of a tokamak (Source: [25]), B: Schematic of a Stellarator (Source: Max-Planck Institut für Plasmaphysik). In a tokamak the magnetic field stems from the field coils and an induced current in the plasma, while in a stellarator the magnetic field stems solely from external coils.

It is, therefore, vital for all reactor concepts to maximize this triple product of particle density $n$, temperature $T$ and energy confinement time $\tau_e$.

At the temperatures necessary for fusion ($\approx 10\,\mathrm{keV}$ [19]), matter takes the form of a plasma. A plasma is a mixture of positively charged atomic/molecular ions and electrons. A plasma is charge neutral on length scales larger than the Debye length [14]. This behaviour is termed quasi-neutrality. Plasmas can exist with orders of magnitude varying densities and temperatures [15]. Throughout the universe, it is the most common state of matter [14], and it is used in many industrial applications such as coating, etching, welding or cutting [22]. The charged particles inside a plasma are affected by surrounding electromagnetic fields through the Lorentz force $F = q\left(\mathbf{E} + \mathbf{v} \times \mathbf{B}\right)$. This enables methods to control and contain the plasma using such electromagnetic fields.

There are several different strategies to obtain the necessary conditions for fusion energy production [8], [19]. Each aims at different ranges of particle density $n$, temperature $T$ and energy confinement time $\tau_e$ to maximize the triple product. One such strategy is called magnetic confinement. In magnetic confinement, the heated plasma is contained using magnetic fields for longer times. This aims at achieving the necessary triple product with temperatures of $\approx 10\,\mathrm{keV}$, densities of $\approx 10^{20}\,\mathrm{m}^{-3}$ and confinement times of the order $1 - 10\,\mathrm{s}$ [23], [24].

In the presence of a magnetic field, the charged particles inside the plasma travel in helical trajectories around the magnetic field lines (gyromotion). Specialized magnetic field shapes are required to keep the particles contained. Magnetic mirror configurations suffer inherently from escaping particles at the ends of the magnetic bottle [7]. Most modern approaches mitigate this loss by using closed magnetic field lines in a toroidal configuration. The magnetic field is stronger on the inside of the torus than on the outside. This gradient in magnetic field leads to a drift of particles ($\nabla\mathbf{B}$ drift) [13]. The direction of the drift is different for electrons and positive ions, which leads to a charge separation and an electric field. The electric field is perpendicular to the magnetic field. This causes another drift effect ($\mathbf{E} \times \mathbf{B}$ drift), which leads to an acceleration of electrons and ions to the outside of the torus [13]. This can be counteracted by introducing a poloidal field component (see Figure 1.2A). The full magnetic field has then a helical shape. Today, mainly two reactor designs are investigated: the tokamak and the stellarator. Both designs differ in the way they generate the poloidal field. Stellarators where initially envisioned to be figure-8 shaped devices [26]. Modern stellarators are build in toroidal shape with special magnetic field coils around the plasma torus. The field coils are specially shaped to create both the toroidal and poloidal magnetic field components (Figure 1.2B). The resulting magnetic field is not toroidally symmetric. Most notable operating experiments today are Wendelstein 7-X [27] in Greifswald, Germany and the LHD [28] in Toki, Japan. In contrast, a tokamak is toroidally symmetric, and the poloidal magnetic field is created by an induced electric current in the plasma itself. This requires a strong magnetic field coil as center column of the tokamak (Figure 1.2A). The tokamak functions like a transformer with the plasma acting as secondary winding. Due to the laws of induction, a constant current in the plasma can only be invoked if the current in this central solenoid is continuously increasing. Because the current in the solenoid can not be increased indefinitely, tokamaks need to operate in pulses. Stellarators can run continuously in steady-state. Some research projects test specific external heating systems that produce additional currents in tokamaks [29]. These currents increase the possible pulse duration and might make it possible to run tokamaks in steady-state.

The following section highlights some of the key scientific challenges for tokamaks, that are currently worked on.

## 1.2 Current tokamak research and challenges

The main objective for fusion research is the development of high performance plasmas with large $Q$ values [30]. To achieve this, the confinement of the plasma needs to be improved. In many experiments, an increase of external heating power leads to a plasma regime with significantly increased energy confinement times [31]. In this high confinement mode (H-mode), the pressure in the whole confined plasma is elevated. But, the plasma density and temperature can not increase indefinitely. Especially for the plasma density upper limits exist [32]. Exceeding these limits invokes instabilities found in tokamak plasmas [13], [33], [34]. Such instabilities lead frequently to disruptions, during which the confinement is lost. In a disruption the entire energy stored in the plasma and its electric current is released rapidly onto the reactor vessel walls [13]. For power plant scale reactors, these events can be so violent that mechanical damages would occur and disruptions need to be avoided [35]. With these limitations of the achievable plasma density, larger confinement times are required to increase the triple product. Experimental scalings have demonstrated that confinement times increase with reactor size and magnetic field strength [36]. To achieve higher confinement times, future experiments (ITER, DEMO) are larger in size [37] or contain stronger magnetic fields [38]. Both increases the costs and the engineering complexity of the reactors.

The strongest magnetic fields can be generated with recently developed high temperature superconductor (HTS) tapes [39]–[41]. On a power plant scale, these magnets have to withstand extreme forces [42], [43]. Their construction is an engineering challenge and an open field of research [44].

The high energy neutrons, which stem from the D-T reaction, are not affected by the magnetic field and hit every part of the reactor wall. Due to their high energy, they penetrate deep into the material. This affects the plasma facing and structural components and possibly even the more sensitive magnets [45], [46]. Neutron radiation has a strong impact on the mechanical and thermal properties of materials ([47] and therein). Additionally, it leads to activation of the radiated parts. Current research tests and develops new materials with high radiation resilience, longer lifetimes and desired short-lived activation [30], [48]–[50].

The required deuterium for the D-T reaction can be extracted from regular water. The amounts available on earth would last for millions of years of fusion energy [13]. However, there is hardly any tritium naturally available on earth due to its short half-life of 12.3 years [13]. The tritium available mostly comes from nuclear fission based breeding reactors. The estimated available worldwide tritium reserves would be consumed by a single fusion based power plant in a few years [51], [52]. To circumvent this bottleneck, nuclear fusion plants are planned to breed their own tritium from lithium in breeding blankets [53], [54]. These blankets are build into the reactor and produce tritium by using the neutrons from the D-T fusion in the following reactions [13]:

$$\begin{aligned} {}^{6}_{3}Li + {}^{1}_{0}n &\rightarrow {}^{3}_{1}T + {}^{4}_{2}He + 4.8\,\mathrm{MeV} \\ {}^{7}_{3}Li + {}^{1}_{0}n &\rightarrow {}^{3}_{1}T + {}^{4}_{2}He + {}^{1}_{0}n - 2.5\,\mathrm{MeV} \end{aligned} \tag{1.4}$$

The estimated lithium reserves on earth could provide enough tritium to address the global energy demand for centuries [13]. Each D-T reaction in Equation 1.1 creates only a single neutron, which itself can create only a single tritium nucleus. If not all neutrons undergo this reaction, the tritium content in the reactor decreases over time. The tritium breeding ratio (TBR) is the ratio of produced to consumed tritium. For a tritium self-sufficient reactor it needs to be larger than 1. To reach TBR $\geq 1$ the breeding blanket needs to include elements like beryllium that can increase the amount of available neutrons through additional reactions [55], [56]. But predicting the TBR for future reactors through modeling comes with large uncertainties ([19] and therein).

Additionally to the aforementioned difficulties, a "reliable solution to the problem of heat exhaust [...] is one of the main challenges in realising magnetic confinement fusion" [53]. Since the tokamak exhaust is the integral topic of this thesis, it is adressed more extensively in the following section.

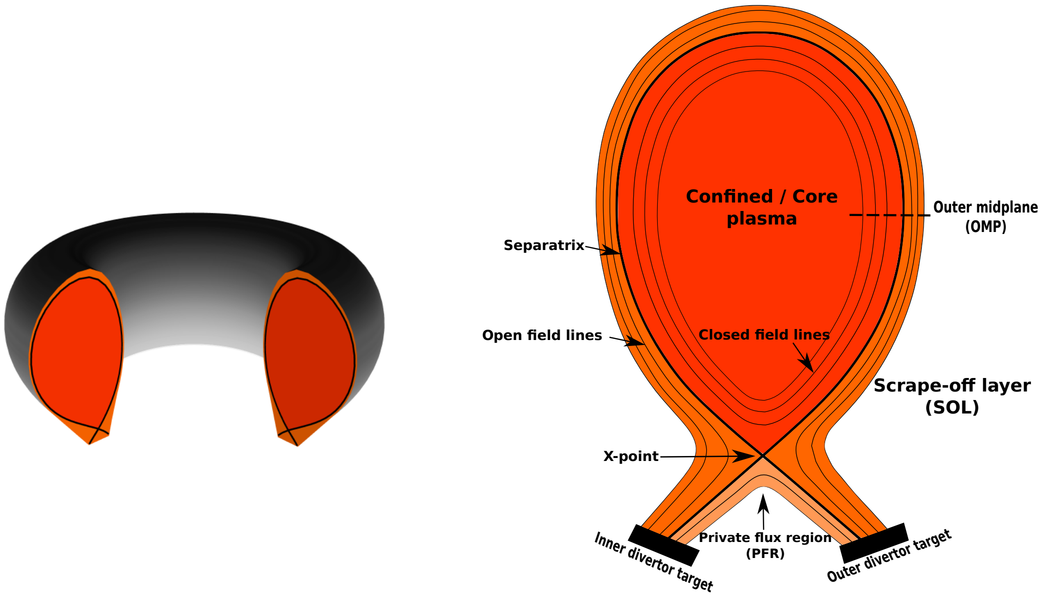


**Figure 1.3:** Schematics depicting a tokamak cross-section with divertor. In the scrape-off layer (SOL) the magnetic field lines start and end at the divertor targets. In contrast to the scrape-off layer, the private-flux region (PFR) is not in direct contact with the main core plasma.

## 1.3 The exhaust challenge

Magnetic confinement of the plasma is never perfect, and there are several mechanisms that lead to an outward flow of the plasma across the magnetic flux surfaces (see Section 2.5). This plasma will ultimately hit the vessel wall, transfer heat and erode the wall. To contain the interaction of the plasma with the vessel wall, earlier tokamak experiments included a limiter. The limiter extends from the wall and scrapes off the outermost layer of the plasma. Hence, this layer is called the scrape-off layer (SOL) and determines the properties of the tokamak exhaust [16]. The interaction of the hot plasma with the limiter surface erodes material from the limiter (called sputtering). The eroded material enters the plasma, which is undesirable as it reduces the main plasma performance (see Section 2.4). To increase the distance between the region of plasma-wall interaction (PWI) and the main plasma, the divertor was invented [16]. Due to additional benefits, the divertor configuration is now considered superior [16]. In the divertor configuration additional external poloidal field coils are introduced that create a magnetic separatrix (Figure 1.3). Inside the separatrix the magnetic field consists of nested closed magnetic flux surfaces. Outside of the separatrix the magnetic field lines start and end at the target plates of the divertor. The point on the separatrix with vanishing poloidal magnetic field is called X-point. The design of the divertor and the control of the processes in the SOL need to fulfill the following four key requirements.

The heat enters the SOL from the main plasma and is conducted along the magnetic field lines towards the divertor targets. Because of the small width of the SOL ($1-10$ mm), this can lead to extreme heat fluxes [57]. To withstand larger heat fluxes, modern divertor plates and other plasma facing components are constructed from tungsten and are actively cooled [58]. The heat flux that can be sustained in steady state without damages to the divertor is around 10 MW/m$^2$ [58]. For power plant scale reactors, like DEMO, the unmitigated heat flux could exceed 100 MW/m$^2$ [59]. Hence, the primary challenge of the tokamak heat exhaust is to reduce the heat flux to the divertors to the sustainable level [59].

The divertor plate material needs to be resilient against high temperatures and thermal stresses. Earlier tokamaks used divertors made from graphite. Graphite is less suitable for actual power plants because the tritium tends to be retained in the carbon [60], [61]. This leads to higher radiation of the reactor and difficulties to control the amount of tritium present. For future reactors tungsten is foreseen as divertor material [58]. The problem with tungsten is even if small amounts are eroded and enter the plasma this leads to drastic cooling effects [62]–[64]. The erosion depends heavily on the type of impurity ions present inside the plasma besides the hydrogen isotopes and helium (see Section 2.4). Considering a mix of argon and neon impurities, the erosion of tungsten can be mitigated to acceptable levels if the plasma temperature at the divertor targets is below 5 eV [24], [59]. The exact limit depends on the type of impurities and the necessary lifetime of the divertor. The second requirement for the tokamak exhaust is, therefore, to reduce the plasma temperatures at the divertor target low enough for an acceptable level of target sputtering.

Besides managing the PWI, the SOL has the task to remove impurities deluting the main plasma [16]. This concerns both the impurities caused by PWI as well as the helium that is generated by the D-T fusion reaction. The helium removal needs to be identical to the production otherwise the helium concentration in the plasma would rise over time. In that case a larger fraction of the heating power would be directed towards heating the unreactive helium nuclei, which lowers the overall performance. This problem is called plasma dilution and should be avoided [65]. At low temperatures in the divertor volume or at contact with the plate surfaces, the helium ions recombine with electrons to neutral atoms. The emerging neutral gas can be removed by pumps placed in the vessel. The efficiency of such pumps is tightly correlated with the pressure of the neutral gas. The exhaust needs to enable enough neutral gas pressure such that the pumps can keep up with the helium production rate [65].

Besides these requirements on the divertor side of the SOL, the SOL also acts as an important boundary condition for the confined core plasma. To achieve the necessary triple product for fusion energy production, future tokamaks are expected to operate in H-Mode [35], [38], [58]. The exact physical mechanisms behind the regime transition into H-mode are not fully understood [13]. Experimental scalings predict a necessary minimum power crossing the separatrix into the SOL in combination with certain plasma densities at the separatrix [66]. Despite the confinement advantages, in H-mode operation plasma instabilities build up. The instabilities cause periodic outbursts of heat and particles. These edge-localized-modes (ELM) lead to transient heat loads at the divertor far greater than the already difficult steady-state load. These transient heat loads pose a serious problem for power plant scale reactors [59], [67]. However, new regimes and suppression mechanisms have been discovered which can maintain the benefits of the H-mode while reducing the transient heat fluxes of ELMs [67]–[69]. The last requirement for the exhaust is therefore to provide a plasma SOL that meets the requirements of suitable conditions at the separatrix.

## 1.4 Shortcomings of current exhaust models

To achieve all the necessary plasma exhaust conditions (Section 1.3), it is impertinent to understand and control the physical processes in the SOL. Specifically for the design and control of tokamaks it is required to have suitable models that describe the phenomena accurately. Current modeling efforts rely to a large degree either on simple analytical models (see Section 2.3) or empirical scaling relations [57], [70]–[74]. But, also the analytical models require experimentally fitted corrections to incorporate the complex physical effects. Such models are fast to evaluate but not sufficiently precise for reactor design. Especially the validity of models which rely on experiment specific correction factors is uncertain for larger reactor sizes. While 1-dimensional (1D) simulations are able to capture more of the physical effects self-consistently, an accurate representation of the plasma can only be achieved by calibration against 2-dimensional (2D) simulations [75]. 2D simulations of the tokamak scrape-off layer are necessary to include cross-field transport effects sufficiently. Specifically for divertor design, the divertor shape and geometric effects on the neutral gas need to be included in the model. Such 2D plasma exhaust codes, like SOLPS-ITER (Section 2.7), simulate the multi-physics phenomena in the SOL sufficiently well but come with three key problems [76]–[78]:

1. The stability of the simulations requires timesteps around $10^{-7}\,\mathrm{s}$ [79]. With macroscopic timescales on the order of $0.1-1\,\mathrm{s}$ in the SOL [80], the simulations take exceedingly long compute times (on the order of weeks to months) [76], [81].
2. Operating such simulations is a complicated task, which requires manual fine-tuning and adjustments of numerical settings by experts [79], [82].
3. Each simulation yields only a result for a single very specific scenario. But many different parameters and their interactions are relevant. Thus, the curse of dimensionality (see e.g. [83]) leads to more scenarios than are testable by simulation.

Identification of optimal design points for future tokamaks requires large sensitivity studies, which scan across many parameters. Such an endeavour is severely bottenecked by these properties of the simulations. Such studies are especially necessary, because mechanisms of anomalous cross-field transport are not self-consistently included in these models but are prescribed [84]. The strength of this transport is still not fully known for future devices [30], [57] (Section 2.5). Additionally, the difficult nature of these codes prohibits a tight integration with models for other plasma domains. Of interest is a coupling to models of the confined plasma or PWI. Therefore, it is current standard practice to run PWI codes (e.g. DIVIMP [85] or ERO2.0 [86]) as post-processing step on simulated plasma distributions, without a self-consistent treatment of the eroded particles in the plasma. For fast design studies of future devices, coupling to other codes, the integration in system codes or model based control systems, faster and more robust models of the SOL are needed.

Surrogate models are a possible candidate for such models. Surrogate models are developed by applying machine learning techniques on datasets of simulations [87]. The surrogate is then capable of interpolating in the high dimensional parameter space and provides fast results for previously untested scenarios. It acts as a replacement for the slow and difficult simulations. Besides fast evaluation speeds, some surrogate models offer additional benefits. An example is the differentiability of neural networks, which can be used in gradient based design optimizations. Surrogate modeling is used in a vast number of fields, such as engineering [88], medicine [89], neuroscience [90], plasma physics [91] and has also been tested for SOL simulations [92]–[94]. The validity of surrogate models is coupled to the validity of the underlying simulations. This comes with the drawback that only those physical effects are included which are present also in the simulations. In comparison to developing models based on experimental data (as e.g. in [95]), surrogate models allow for an extrapolation to future scenarios (as long as these can be simulated). Also surrogate models are more free from hidden influences, which might occur only in specific experiments. The development of surrogate models requires running many of the simulations one aims to replace. Therefore, the key problem in surrogate modeling is finding models and procedures that perform well with minimal amounts of data.

## 1.5 Aim of this thesis

The goal of this study is to develop a proof-of-concept of a surrogate model for the scrape-off layer plasma. The model should be so broad in its scope that it can be applied both to existing tokamak experiments as well as for planning future power plant scale reactors. The main focus is not to develop just a single model but to test and develop methods that allow the creation of such models. These findings should allow consecutive studies to increase the models capabilities both in terms of validity and scope. This involves creating a setup and strategies for the efficient generation of a simulation database and finding suitable model architectures. In principal a large and flexible enough model should be able to predict the whole simulation domain without loss of accuracy, but achieving this in practice is not so trivial as larger models might be more difficult to optimize. Therefore, this study explicitly compares surrogates that model only specific parts of the scrape-off layer against models for the whole domain and models for individual quantities against models for all observables. Another important aspect is the evaluation of the number of simulations which is necessary to develop such models. This is accompanied by testing strategies for calibrating the models against fewer higher fidelity datapoints. This should provide guidelines how this proof-of-concept can be expanded to a fully capable scrape-off layer model.

The following chapter provides an overview of the relevant physics and models of the scrape-off layer. This also introduces the SOLPS-ITER software, which is used to perform the scrape-off layer simulations in this thesis. Basic machine learning concepts and the specific methods used in this thesis are introduced in Chapter 3. Chapter 4 presents the concepts and changes made to a SOLPS-ITER simulation which allow it to be used for the generation of a cross machine simulation database. Several preliminary tests were conducted to find a set of numerical parameters and boundary conditions that yield stable simulations in varying conditions. The results of these tests are presented in Chapter 5. Then Chapter 6 provides a short overview of the final simulation setup and the procedures that are used to construct the simulation database used in all subsequent chapters. Chapter 7 performs an analysis of this database to determine the validity of the simulations and the physical effects that are covered by it. Chapter 8 shows the development and evaluation of many surrogate models based on different machine learning methods or with different scopes. Some of the applications and results than can be obtained using these developed models are demonstrated in Chapter 9. Chapter 10 extends the surrogate models developed here to include high fidelity simulations from a pre-existing database. And finally Chapter 11 provides a summary of the central conclusions of this thesis and an outlook into future research possibilities.

# Chapter 2

# The scrape-off layer: Physics and modeling

Modeling is an integral part of the scientific process [96]. The scientific process to answer a specific question can in its briefest form be described in 3 steps. First a hypothesis is formed, which aims to explain the phenomena in question. In the most natural sciences but particularly in physics, this hypothesis is often formulated as a mathematical model in the form of a set of governing equations. The derivation of such a mathematical model usually requires assumptions or constraints, which restrict the applicability of the model. Yet such models usually attempt to generalize the rules behind phenomena that have been observed under specific conditions. The second step in the scientific process is performing an experiment or a series of experiments to test the hypothesis. The last step consists of an evaluation and comparison between the hypothesis and the experimental findings. This should either approve the validity of the hypothesis or discard the hypothesis. In the latter case a new refined hypothesis is developed and the iterative process starts again. In modern science the challenges in each of this three steps can be so difficult that they require the effort of many people for several decades. Especially in nuclear fusion research, experiments require the construction of dedicated facilities with hundreds to thousands of involved scientists, years of planning and immense monetary costs [97].

To test a hypothesis, first testable predictions need to be derived. In nuclear fusion research, but also in other domains, this can be a challenging task in itself. The challenge arises because the initial formulation of a hypothesis is usually not directly predictive but constructed as a set of governing equations. These governing equations need to be first solved to derive predictions for the behaviour of observables that can be measured in experiments. In most cases the equations are so complicated that they cannot be solved analytically. Instead the mathematical model needs to be converted into a numerical model (also called simulation, computer model, executable model) which allows to derive predictions [98]. This in itself can involve an interative process of making assumptions and performing tests to make the numerical model confirm to the mathematical model. The comparison between the numerical and mathematical model is termed ”verification” while the comparison between the final predictions to experiment is called ”validation” [98]. Plasma exhaust simulations have multiple applications [84], but most important is their ability to make predictions for settings which are not yet covered by existing experiments.

Specifically for the design of future experiments and tokamak power plants, highly validated and usable numerical models are required to test and compare different configurations and scenarios. Especially the high cost and time required for tokamaks experiments prohibits tests of all reasonable design ideas in practice.

This thesis is involved in creating replacements for the currently used numerical models of the tokamak scrape-off layer. Therefore, this section gives an introduction into the current state of modeling for the scrape-off layer and the most relevant physical effects and phenomena that occur.

This chapter follows mostly [13], [16]. Any information given in this chapter without explicit citation can be found in one of the two textbooks.

## 2.1 Kinetic and fluid equations

Modeling a plasma requires a self-consistent description of the position and motion of particles in the electromagnetic fields. In fusion plasmas the particle density is so high that not all individual particle trajectories can be tracked but instead need to be described in a statistical way. This can be expressed by the probability density $f_a(\mathbf{x}, \mathbf{v}, t)$ of particles of type $a$ in the six dimensional phase space $(\mathbf{x}, \mathbf{v})$ and time $t$. For a single species plasma (meaning it contains only electrons and one type of ions) the behaviour of this distribution function can be described by the kinetic equation (Equation 2.1). The equation describes the particle motion, the influence of the Lorentz force and particle collisions $\left(\frac{\partial f_a}{\partial t}\right)_c$.

$$\frac{\partial f_a}{\partial t} + \mathbf{v} \cdot \nabla_x f_a + \frac{q_a}{m_a}(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_v f_a = \left(\frac{\partial f_a}{\partial t}\right)_c \tag{2.1}$$

If the collision term is neglected, this corresponds to the Vlasov equation. Because the plasma particles interact through their charge rather than through hard collisions, the derivation of an appropriate collisional term needs to consider many small angle collisions. The equation than takes the form of the Fokker-Planck equation. However, in many parts of the plasma such as the SOL the collisionality is high and the mean free path is low. In these cases not the full distribution function $f_a$ is necesssary, but instead the plasma can be described through its moments (Equations 2.2-2.4). The moments are the fluid density $n_a$, fluid velocity $\mathbf{u}_a$ and temperature $T_a$. As is common in plasma physics the temperature implicitly contains the Boltzmann constant (i.e. is a measure in units of energy).

$$n_a(\mathbf{x}, t) = \int f_a(\mathbf{x}, \mathbf{v}, t) d\mathbf{v} \tag{2.2}$$

$$\mathbf{u}_a(\mathbf{x}, t) = \frac{1}{n_a} \int \mathbf{v} f_a(\mathbf{x}, \mathbf{v}, t) d\mathbf{v} \tag{2.3}$$

$$T_a = \frac{1}{n_a} \int \frac{m_a}{3} (\mathbf{v} - \mathbf{u_a})^{\mathbf{2}} f_a(\mathbf{x}, \mathbf{v}, t) d\mathbf{v} \tag{2.4}$$

The dynamics of these moments can then be described by the Braginskii fluid equations [99]:

$$\frac{dn}{dt} + n\nabla \cdot \mathbf{u} = S_{par} \tag{2.5}$$

$$mn\frac{d\mathbf{u}}{dt} + \nabla p + \nabla \boldsymbol{\pi} - nZe\left(\mathbf{E} + \mathbf{u} \times \mathbf{B}\right) = \mathbf{S}_{mom} \tag{2.6}$$

$$\frac{3}{2}n\frac{dT}{dt} + nT\nabla \cdot \mathbf{u} + \nabla \cdot \left(\boldsymbol{\pi} \cdot \mathbf{u}\right) - \nabla \boldsymbol{\pi} + \nabla \cdot \mathbf{q} = S_{ene} \tag{2.7}$$

with the substantive derivative $\frac{d}{dt} = \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla$, the pressure $p = nT$ and the stress tensor $\boldsymbol{\pi}$. The stress tensor contains the non-isotropic parts of the pressure $\pi_{\alpha\beta} = \int m\left(v_\alpha - u_\alpha\right)\left(v_\beta - u_\beta\right) f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v} - p\delta_{\alpha\beta}$. The momentum is also controlled by the forces on the plasma particles with charge $Ze$ in external electromagnetic fields $Ze\left(\mathbf{E} + \mathbf{u} \times \mathbf{B}\right)$. The energy balance equation further relies on the conductive heat flux $\mathbf{q} = \int \frac{m}{2}\left(\mathbf{v} - \mathbf{u}\right)^2 \mathbf{v} f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v}$. $S_{par}$, $\mathbf{S}_{mom}$, $S_{ene}$ are the sources for particles, momentum and energy. For a single species plasma with no external sources, the particle sources vanish and the others come down to the effects of particle collisions. It shows, the defining equation of each moment always requires the next higher order moment. The particle continuity equation requires the velocities $\mathbf{u}$, the momentum balance requires the pressure $nT$ and the energy balance equation requires the heat flux $\mathbf{q}$. Because this trend continues also to higher orders, there are always less equations than unknowns. To make the system solvable, an assumption about a higher order moment needs to be made. Usually this is an assumption on the form of the heat flux $\mathbf{q}$. Braginskii published a derivation of all relevant terms for a single species plasma [99] and an extension to multispecies plasmas was published by Zhdanov [100]. The latter defines the basic equations in SOLPS-ITER (Section 2.7). Because the heat and particle transport is highly non-isotropic and follows the magnetic field lines, the equations are usually decoupled into separate equations for the parallel and perpendicular transport. Then for the parallel-to-$\mathbf{B}$ conductive heat flux an assumption of the form $q_{\|} \propto T^{5/2}\nabla_{\|}T$ is made [16].

The fluid approximation is only valid if the velocity distributions are Maxwellian. This is only the case when the mean free path $\lambda_{\mathrm{mfp}}$ is short compared to the relevant length scales. Regions of low collisionality require a different treatment. Such is the case at the interfaces between the plasma and elements of the reactor wall. Here a plasma sheath develops [16].

## 2.2 Plasma sheath

Assuming equal temperatures for electrons and ions leads to larger thermal velocities of the electrons due their significantly lower mass. In the proximity of the surface of solid materials, this leads to an accumulation of electrons at the surface and a net negative charge. This inhibits the flux of electron and increases the flux of ions to the surface, until a potential between plasma and surface emerges which leads to an ambipolar flux. Due to the higher velocities of the electrons, this ambipolar flux requires a region of increased ion density in front of the wall $n_i > n_e$. This region is called the sheath, and it shields the remaining plasma from the wall potential due to

Debye shielding. The characteristic length scale for the sheath is the Debye length.

$$\lambda_{\text{Debye}} = \sqrt{\frac{\varepsilon_0 T_e}{n_e e^2}} \tag{2.8}$$

A fluid description of the sheath plasma is not accurate as the condition $\lambda_{\text{mfp}} < \lambda_{\text{Debye}}$ is violated. Therefore, the fluid description of the SOL plasma ends at the entrance to the sheath. Kinetic or analytical models of the sheath are required to provide the boundary conditions for the fluid description at the entrance to the sheath. Here the derivations for a single ion species plasma following [16] are presented. The shielding in the sheath is not perfect, and a small electric field extends into the plasma, which accelerates the ions to achieve the ambipolar flux. In this pre-sheath region still $n_i \approx n_e$ holds. It can be derived that the ion velocities at the entrace to the sheath $u_{se}$ must fulfill the Bohm criterion [101]:

$$u_{se} \geq c_s = \sqrt{\frac{T_e + \gamma T_i}{m_i}} \tag{2.9}$$

$\gamma$ is the adiabatic coefficient in the sheath. For isothermal conditions $\gamma = 1$ holds while for non isothermal conditions $\gamma \in \left[\frac{5}{3}, 3\right]$ depending on the exact conditions in the sheath.

With the simplifying assumption that the ion velocities follow a Maxwellian distribution with mean $c_s$, it can be derived that both the ion and electron fluxes at the solid wall are exactly the same as at sheath entrance $\Gamma^i_{se} = \Gamma^i_{\text{wall}}$; $\Gamma^e_{se} = \Gamma^e_{\text{wall}}$.

The sheath does not only act as particle sink for the plasma ions but also as an energy sink. Using the previous derivations and similar assumptions, the ion heat fluxes entering the sheath can be written as:

$$q^i_{se} = \left(\frac{5}{2} T_i + \frac{1}{2} m_i c_s^2\right) \Gamma_{se} = \gamma_i T_i \Gamma_{se} \tag{2.10}$$

This includes the ion sheath heat transmission coefficient $\gamma_i = \left( \left(\frac{5}{2} + \frac{\gamma}{2}\right) + \frac{1}{2}\frac{T_e}{T_i} \right) \approx \frac{7}{2}$. The heat flux for the electrons can be written as:

$$q^e_{se} = \gamma_e T_e \Gamma_{se} \tag{2.11}$$

The electron sheath heat transmission coefficient $\gamma_e = 2 + \frac{|eV_{\text{wall}}|}{T_e} + \frac{\left|eV_{\text{pre-sheath}}\right|}{T_e} \approx 5.5$ contains the potential drop at the target $V_{\text{wall}}$ and the potential across the pre-sheath $V_{\text{pre-sheath}}$. Without any heat sources/sinks in the sheath the total heat flux at the target surface is then $q_{\text{wall}} = q_{se} = q^i_{se} + q^e_{se}$.

The previous derivations expect a target surface that is normal to the magnetic field lines. In practice the target surfaces are usually strongly inclined compared to the magnetic field to spread the heat flux across a larger area. This requires a more complicated description, the Chodura Sheath [16], [102], which includes an additional magnetic pre-sheath between the ordinary pre-sheath and sheath. However, in terms of the boundary condition for the SOL little changes, as the derivations lead to a similar criterion for the ion velocity parallel to the magnetic field at sheath entrance $u_{\parallel B,se} \geq c_s$.

## 2.3 Regimes in the scrape-off layer

It is generally accepted that the physics of the SOL and divertor can be differentiated into three principal regimes [103]. The transitions between these regimes can be fluent and there is no uniquely used criteria for defining these regimes. This section gives an introduction to the principle physical mechanisms and characteristics of the regimes in the scrape-off layer using simple analytical models. In this the section follows mostly [16], [103]. Information given without explicit citation can be found in either of the two publications.

### 2.3.1 The Sheath-Limited Regime

One of the key drivers behind changes in the SOL regime is the plasma density at the boundary to the confined main plasma. It is a common assumption in simple SOL models that all of the plasma particles and heat from the confined plasma enter the SOL at one fixed position called upstream. For configurations with a single x-point this upstream position is often chosen as the outer midplane (OMP, see Figure 1.3), because most of the heat flux enters the SOL there [104]. The sheath-limited regime occurs for the lowest upstream plasma densities $n_u$. Here the plasma density at the divertor target $n_t$ grows linearly with the upstream density. The electron and ion temperatures are high and constant along a flux tube. This leads to relatively high plasma temperatures at the divertor target $T_t$. Because no temperature gradients (parallel to B) are present, the heat transport is mainly convective and almost no heat conduction occurs. In this regime the only relevant source of particles and heat is the transport into the SOL from the main plasma. Similarly the divertor targets act as the only sink for particles and heat. The plasma ions hitting the solid divertor material transfer their energy to the material and are emitted back as neutralized atoms. The ionization of the neutral gas inside the SOL (called recycling) is low. Because of the low plasma densities, the mean free path of the gas particles is long, such that they already moved into the main plasma before they get ionized. The fundamental property of this regime is the low plasma collisionality. Although this makes a fluid treatment actually invalid, the key properties found in a fluid approximation match more complete kinetic models [16]. With the assumptions that the plasma particles enter the SOL with zero velocity and enter the sheath at the target with exactly sound speed, it can be derived for an isothermal fluid:

$$T_t = T_u \tag{2.12}$$

$$n_t = \frac{1}{2} n_u \tag{2.13}$$

$$\mathit{\Gamma}_t = \frac{1}{2} n_u c_s \tag{2.14}$$

The plasma properties at the target always refer to the properties at entrance to the sheath.

### 2.3.2 The Conduction-Limited Regime

With increasing upstream plasma density $n_u$, the collisionality in the SOL rises. This causes conduction to dominate over convection as main heat transport mechanism parallel to the magnetic field lines. Due to the finite heat conductivity this causes the emergence of temperature gradients parallel to the field lines. These temperature gradients allow for large temperature differences between the plasma upstream and at the divertor target. Because of the dominance of conductive heat transport, this regimes is called conduction-limited regime. With the higher collisionality, the mean free-path of the neutral gas particles before ionization is reduced. The ionization of these inside the SOL becomes a significant source of plasma particles close to the divertor target. Therefore, the regime is also often described as high-recycling regime. Using the assumption that the total pressure and energy are conserved inside each individual flux tube (a flux tube is the plasma region along one magnetic field line), the Two Point Model (2PM) can be derived [103]. The model equations infer the plasma properties downstream at the divertor target in relation to the upstream quantities. Using these assumption the following can be derived:

$$T_t \propto \frac{q_{\|,u}^{10/7}}{L^{4/7} n_u^2} \tag{2.15}$$

$$n_t \propto \frac{n_u^3 L^{6/7}}{q_{\|,u}^{8/7}} \tag{2.16}$$

$$\mathit{\Gamma}_t \propto \frac{n_u^2 L^{4/7}}{q_{\|,u}^{3/7}} \tag{2.17}$$

These equations show some of the properties that are used in experiments to diagnose this regime. An increase of the plasma density at the divertor target, which is more than quadratic with the upstream density, and a quadratic dependence of the ion flux at the target. The relationship between the target temperature and upstream plasma density is generally favorable because a high fusion output requires high upstream densities, which will consequently cause low target temperatures. It is also apparent that the plasma properties depend strongly on the upstream parallel heat flux in the SOL $q_{\|,u}$, which can only be partially controlled from outside, and on the connection length $L$, which describes the distance following a magnetic field line from upstream to the divertor.

### 2.3.3 Divertor Detachment

When the upstream density is increased even more, the plasma properties at the target will divert at some point from the relations given in the Equations 2.15-2.17. The plasma density and particle flux at the target will first stagnate and then decrease with rising upstream density (not necessarily both at the same time) [105]. This phenomenon is called rollover and is caused by strong interactions with the neutral gas particles. These neutral interactions cause significant sources and sinks for the energy, momentum and particles of the plasma in the SOL. The plasma regime after the rollover is referred to as detached, while all regimes before are attached conditions. The 2PM model can be extended through empirical factors to describe the losses in momentum $f_{mom}$ and power $f_{pow}$ between upstream and target:

$$(1 - f_{mom})\, p_{tot,u} = p_{tot,t} \tag{2.18}$$

$$(1 - f_{pow})\, q_{\|,u} R_u = q_{\|,t} R_t \tag{2.19}$$

Here $p_{tot}$ is the total pressure, which is the sum of static and dynamic pressure for both electrons and ions, and $R_u$, $R_t$ are values for the tokamak major radius measured at upstream and target. With the assumptions of a purely hydrogenic plasma, sound speed at the target and $T_i = T_e$ this leads to the following relations:

$$T_t \propto \left[\frac{q_{\parallel,u}^2}{p_{tot,u}^2}\right]\left[\frac{(1-f_{pow})^2}{(1-f_{mom})^2}\right]\left[\left(\frac{R_u}{R_t}\right)^2\right] \tag{2.20}$$

$$n_t \propto \left[\frac{p_{tot,u}^3}{q_{\parallel,u}^2}\right]\left[\frac{(1-f_{mom})^3}{(1-f_{pow})^2}\right]\left[\left(\frac{R_t}{R_u}\right)^2\right] \tag{2.21}$$

$$\varGamma_t \propto \left[\frac{p_{tot,u}^2}{q_{\parallel,u}}\right]\left[\frac{(1-f_{mom})^2}{(1-f_{pow})}\right]\left[\left(\frac{R_t}{R_u}\right)\right] \tag{2.22}$$

To decrease the plasma temperature the power loss in the SOL needs to be maximized. But the sputtering of the divertor target $\varGamma_{sputtering} = Y(T_t)\varGamma_t$ depends also on the ion flux to the target $\varGamma_t$, which increases with the power loss. Therefore, a balance between momentum and power loss needs to be controlled. The volume power loss $f_{pow}$ stems from either line radiation or charge-exchange reactions with the neutral particles. The momentum loss $f_{mom}$ is caused by the viscosity of the plasma, collisions with the neutral particles and volume recombination of plasma ions to neutral gas particles. Specifically volumetric recombination plays a strong role in detachment and becomes a dominant contribution at plasma temperature below 1 eV [16], [103], [106]. Any momentum or energy transferred from the plasma to neutral particles is dissipated to a wider area because neutral particles are not bound by the magnetic field. It is difficult to formulate analytical descriptions of the loss factors $f_{pow}$ and $f_{mom}$ because these are controlled by complex non-linear atomic and molecular processes. Therefore, it is necessary to estimate these through experiments or simulation. The conditions for detachment can be reached in each fluxtube independently. Often a divertor is called detached when the critical fluxtube, which had the highest heat flux before detachment, is detached. Often the term partial detachment describes situations where not all fluxtubes at the target are detached. Generally it can be distinguished between three different types of detachment, where not necessarily all appear simultaneously:

- Energy detachment - Significantly reduced plasma temperature and heat flux at the divertor target
- Momentum detachment - Significant pressure loss in the SOL which leads to a rollover in the ion flux to the target
- Particle detachment - Reduction in the plasma density at the target, usually seen as a rollover

More strict definitions of detachment might require multiple or all of these conditions [106].

Categorical phenomena as the existence of a rollover can only be seen in designated density scans. Therefore, also different quantitative metrics are used to determine detachment. For example, the degree of detachment (DOD) [105] measures the ratio between the ion flux to the target and the expected flux based on a quadratic scaling with the upstream density. Other metrics set threshold values for the pressure or power loss in the SOL [73], [106] while even simpler classifications set threshold values for the electron temperature at the divertor [103]. These threshold values are then either motivated due their implication on the strength of volume recombination [103], sputtering yield [24] or other heuristics [68], [107]. Because the transition from conduction-limited regime to detachment is gradual, both simple and more complex metrics rely on somewhat arbitrary set threshold values for classification.

Because the detached regime offers the largest possible heat flux reduction, it is foreseen as the operational regime for future fusion reactors and experiments. The extended 2PM is not predictive in itself because it requires prior knowledge of the volume loss terms $f_{pow}, f_{mom}$, the upstream heat flux $q_{\|,u}$ and particle density $n_u$. While $n_u$ is correlated to the plasma density in the confined plasma, their exact coupling relation is in general unknown [103]. The upstream heat flux $q_{\|,u}$ strongly depends on the power fall-off length $\lambda_q$ (i.e. the width of the SOL) for which empirical scalings exist [57], [73], [108]. But it is not known how transferable these are to larger tokamaks [30]. Nevertheless, the equations of the extended 2PM offer insights into mechanism for obtaining detachment and the important aspects that need to be covered in divertor and SOL modeling:

- Power dissipation by line radiation and plasma-neutral interaction, which can be aided through the addition of impurity gases (Section 2.4).
- Increases to the plasma density/pressure.
- The scrape-off layer width as it is determined by cross-field transport (Section 2.5).
- Increasing target radius or other divertor modifications (Section 2.6).

## 2.4 Impurities

The heat flux density towards the divertor can be reduced through line radiation as the photons are not affected by the magnetic field and can distribute the energy on a larger wall surface area. To increase the line radiation rate, additional impurity gases can be inserted because these can radiate even when partially ionized (in contrast to hydrogen). The cooling rates for different impurity elements vary in their temperature dependence. However, the cooling potential at high temperatures rises for elements with higher atomic number Z. This has implications for the allowed sputtering of divertor and reactor wall materials. In graphite walled devices, the sputtered carbon atoms are an inherent source of impurities, which increase the radiation. But with metallic walls, the influx of sputtered high-Z impurities needs to be prevented because these would lead to significant cooling in the confined core plasma, which reduces the possible fusion power and possibly leads to plasma instabilities [16].

To increase line radiation in metallic devices, low-Z impurity gases are added which radiate specifically strong in the divertor and SOL. Candidate gases are Nitrogen, Neon, Argon, Krypton, but most likely a combination of impurities will be used in future reactors [78], [109]. The usage of nitrogen in actual power plants poses a safety concern as it can form ammonia with the radioactive tritium, which leads to higher tritium retention in the reactor and the gas pumping system [110]–[112]. Therefore noble gases are preferred [30]. But also low-Z impurities can lead to a reduction of fusion power due to dilution of the core plasma [16]. This is the case when a major fraction of the electrons in the plasma stem not from hydrogen but from the impurities. Furthermore, the impurities contribute dominantly to the sputtering of wall materials, reducing the lifetime of reactor components and introducing high-Z impurities in the plasma [113]. It is therefore no trivial task to determine optimum impurity levels for reactor operation.

## 2.5 Cross-field transport

Another factor determining the heat flux at the divertor targets is the width of the SOL, which is driven by the strength of plasma transport perpendicular to the magnetic field lines. Several phenomena contribute to the cross-field plasma motion. Single particle effects cause drifts of the gyromotion of ions around the magnetic field due to electric fields ($\mathbf{E} \times \mathbf{B}$), gradients in the magnetic field ($\nabla\mathbf{B}$) and the curvature of the magnetic field. Also collisions between particles of different kind (e.g. ions and electrons or ions of different elements) lead to cross-field transport. Beyond the classical diffusion also neoclassical effects that originate from particles trapped in banana orbits need to be included. However, incorporating all these effects leads to cross-field transport orders of magnitude weaker than estimated from experiments. It is assumed that plasma turbulence is the largest contributor to the cross-field transport although the underlying processes are not yet completely understood. Simulations which treat the turbulence effects self-consistently suffer from numerical complexity. Therefore, these rarely incorporate all heat dissipation mechanisms in the SOL (geometry, impurities). On the other hand, transport codes such as SOLPS-ITER (Section 2.7) assume a purely diffusive cross-field transport for both the particles $\varGamma_\perp$ and the heat $q_\perp$.

$$\varGamma_\perp = -D_\perp \frac{dn}{dr} \tag{2.23}$$

$$q_\perp = -\chi_\perp n \frac{dT}{dr} \tag{2.24}$$

This requires an empirical definition of the effective transport coefficients $D_\perp$ and $\chi_\perp$, which can vary from case to case and depend on the position in the SOL (specifically in edge transport barriers) [114].

## 2.6 Divertor shapes

An easy method to decrease the heat flux onto the divertor targets is to increase the area over which the heat is spread. This plasma wetted area can be increased by aliginig the divertor at a sharp angle compared to the intersecting magnetic field lines. In Super-X configurations the outer divertor is positioned more outwards such that the larger major radius causes the heat to be dissipated on an increased area (see Equations 2.20-2.22). Also the geometric shape of the divertor targets (e.g. vertical or horizontal), as well as how open or closed the divertor area is for the flow of neutral gas particles can impact the dynamics in the SOL [115]–[121]. More advanced geometries try to increase the plasma wetted area and divertor volume by diverting the SOL plasma to multiple divertors either at opposing reactor ends (e.g. double null) or through the introduction of higher order x-points (e.g. snowflake). However, there are significantly fewer simulations or experiments using these advanced configurations compared to the more traditional single null divertor designs (which is shown in Figure 1.3). As a result, the impact of these advanced designs is less understood, and they increase the engineering complexity of the reactors [122]–[124].

## 2.7 SOLPS-ITER

SOLPS-ITER is a state-of-art simulation software for the scrape-off layer [80]. Earlier versions played a vital part in the design of the ITER divertor [76]. It is used in a wide area of applications: to model the SOL and design divertors [125]–[127], to provide backgrounds for PWI [128], [129], for calibration of simpler models [75], [130], [131] or to provide boundary conditions for pedestal models [132]. Using its full capacity all relevant multi-physics phenomena (see the previous sections) are sufficiently accounted for such that the transition between different SOL regimes and detachment can be qualitatively modelled [78], [133]. To obtain quantitatively accurate results, a calibration against experimental results is required. SOLPS-ITER suffers from the drawbacks listed in Section 1.4, such as long computational runtimes, that expert knowledge is required to use it, that each simulation provides only a results for one highly specific scenario and that some of the underlying physical parameters are fundamentally unknown. There exists a variety of simulation codes with similar applications, that all differ slightly in their capabilities and benefits (e.g. UEDGE [134], SOLEDGE2D-EIRENE [135], EDGE2D-EIRENE [136], EMC3-EIRENE [137],...). None is so widely used and extensively validated as SOLPS-ITER [76]. Because this thesis uses SOLPS-ITER to generate the training data, this section gives an overview and introduction of SOLPS-ITER simulations. The level of detail will focus solely on the parts most relevant for this thesis. Therefore, readers are recommended to the publications [80], [84] and the official documentation for further details. Any information about SOLPS-ITER which is given in the following without explicit citation can be found in one of these three sources.

The SOLPS-ITER simulation software [80], [84] consists of the two main codes: B2.5 and EIRENE. The B2.5 transport code [138], [139] assumes a toroidal symmetry to solve 2D fluid conservation equations for multi-species plasma and neutral gas. For the plasma these equations correspond to a variant of the Braginskii equations [99]. For each plasma ion species $a$ the code solves a particle (2.25) and momentum (2.26) conservation equation together with one energy balance equation for electrons (2.27) and one energy balance equation for ions (2.28).

$$\frac{\partial n_a}{\partial t} + \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}\tilde{\mathit{\Gamma}}_{ax}\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}\tilde{\mathit{\Gamma}}_{ay}\right) = S_a^n \tag{2.25}$$

$$\begin{aligned} m_a\frac{\partial n_a u_{\parallel a}}{\partial t} + \frac{1}{h_z\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{h_z\sqrt{g}}{h_x}\mathit{\Gamma}_{ax}^{\mathrm{mom}}\right) + \frac{1}{h_z\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{h_z\sqrt{g}}{h_y}\mathit{\Gamma}_{ay}^{\mathrm{mom}}\right) \\ + \frac{b_x}{h_x}\frac{\partial n_a T_i}{\partial x} + \frac{b_x}{h_x}\frac{Z_a n_a}{n_e}\frac{\partial n_e T_e}{\partial x} = S_a^{\mathrm{mom}} \end{aligned} \tag{2.26}$$

$$\frac{3}{2}\frac{\partial n_e T_e}{\partial t} + \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}\tilde{q}_{ex}\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}\tilde{q}_{ey}\right) + \frac{n_e T_e}{\sqrt{g}}\frac{\partial}{\partial x}\left(b_x u_{\parallel e}\right) = S^e \tag{2.27}$$

$$\begin{aligned} \frac{3}{2}\frac{\partial n_i T_i}{\partial t} + \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}\tilde{q}_{ix}\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}\tilde{q}_{iy}\right) \\ + \sum_{a=0}^{n_s-1}\frac{n_a T_i}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}b_x u_{\parallel a}\right) = S^i \end{aligned} \tag{2.28}$$

Here $\sqrt{g}, h_x, h_y, h_z$ are geometric factors stemming from the x, y coordinate system, which is aligned to the local magnetic field, and $b_x = B_x/B$ is the relative strength of the component of the magnetic field in the x-direction. The code uses toroidal symmetry to solve these equations on a 2D cross-section of the torus. The coordinate system follows the magnetic field lines, such that the $x$-direction always points parallel and the $y$-direction perpendicular to the local magnetic field (inside the torus cross-section). Because the plasma fluid flow is predominantly parallel to the magnetic field lines, the simplification is made that all fluid velocities are expressed as a scalar values $u_a$ with the direction parallel to the local magnetic field. Further, $n_a$ is the particle density of each ion species, $T_e$ the electron temperature and $T_i$ the ion temperature. An individual ion species can be ions of different atom types (either different elements or isotopes) or different ionization levels of the same atom (e.g. $\mathrm{N}^+$, $\mathrm{N}^{2+}$, ...). The electron density $n_e$ and average velocity are derived from the conditions of quasi-neutrality $n_e = \sum_{a=0}^{n_s-1} z_a n_a$ (where $z_a$ denotes the charge of the ion species) and ambipolarity. While the code can also solve the electric current equation self-consistently, this is not used in this thesis but instead an electric potential of $\Phi = 3.1T_e/e$ is assumed everywhere. Because of the very large mass difference between electrons and ions $m_i/m_e \approx 2000$, these fluids are not necessarily equilibrated and therefore possess different temperatures. In contrast, the different ion species share one common temperature $T_i$. In the Equations 2.25-2.28 the terms $\tilde{\mathit{\Gamma}}_a, \mathit{\Gamma}_a^{\mathrm{mom}}, \tilde{q}_e, \tilde{q}_i$ constitute fluxes and $S^n, S^{\mathrm{mom}}, S^e, S^i$ sources of particles, momentum or energy. Another key assumption is that the transport perpendicular to the magnetic field is purely diffusive. Without drift or current effects, this yields the following relations for the fluxes:

$$\tilde{\Gamma}_{ax} = b_x u_{\parallel a} n_a - D_\perp \frac{1}{h_x}\frac{\partial n_a}{\partial x} \tag{2.29}$$

$$\tilde{\Gamma}_{ay} = -D_\perp \frac{1}{h_y}\frac{\partial n_a}{\partial y} \tag{2.30}$$

$$\Gamma_{ax}^{\mathrm{mom}} = m_a u_{\parallel a}\tilde{\Gamma}_{ax} - \eta_{ax}\frac{1}{h_x}\frac{\partial u_{\parallel a}}{\partial x} \tag{2.31}$$

$$\Gamma_{ay}^{\mathrm{mom}} = m_a u_{\parallel a}\tilde{\Gamma}_{ay} - \eta_{ay}\frac{1}{h_y}\frac{\partial u_{\parallel a}}{\partial y} \tag{2.32}$$

$$\tilde{q}_{ex} = \frac{3}{2}\tilde{\Gamma}_{ex}T_e - \kappa_{ex}\frac{1}{h_x}\frac{\partial T_e}{\partial x} \tag{2.33}$$

$$\tilde{q}_{ey} = \frac{3}{2}\tilde{\Gamma}_{ey}T_e - \kappa_{ey}\frac{1}{h_y}\frac{\partial T_e}{\partial y} \tag{2.34}$$

$$\tilde{q}_{ix} = \frac{3}{2}\tilde{\Gamma}_{ix}T_i - \kappa_{ix}\frac{1}{h_x}\frac{\partial T_i}{\partial x} \tag{2.35}$$

$$\tilde{q}_{iy} = \frac{3}{2}\tilde{\Gamma}_{iy}T_i - \kappa_{iy}\frac{1}{h_y}\frac{\partial T_i}{\partial y} \tag{2.36}$$

The heat conductivities contain both a classical and the anomalous part $\kappa_e = \kappa_e^{CL} + \chi_\perp n_e$, $\kappa_i = \kappa_i^{CL} + \chi_\perp \sum_{a=0}^{n_s-1} n_a$. Parallel to the magnetic field the particle and heat transport are dominated by the classical particle flow and heat conduction, but perpendicular to the field only the anomalous parts play a role. Here $D_\perp$ and $\chi_\perp$ constitute effective anomalous transport coefficients (see Section 2.5), which are usually chosen by testing various values and selecting those that yield similar plasma properties as measured in experiments. The source terms contain effects stemming from the interaction of particles from differing species and the surroundings. Among these are: Ionization, recombination, collisions, charge-exchange reactions and radiation. In these reactions the sources terms also capture the interaction between the plasma and the neutral gas particles.

In most SOLPS-ITER simulations the EIRENE [140] code is used to solve the kinetic Boltzmann equation for neutral gas particles. It is a fully 3D (in space) Monte-Carlo code, which samples trajectories of neutral test particles on a fixed plasma background. The test particles are produced by recombination of ions on the solid reactor surfaces and in the plasma volume or due to sputtering (physical or chemical erosion of solid surfaces). Constant particle influx and absorption coefficients can be specified on some surfaces to model gas puffing and pumping respectively. Various physical processes are taken into account in great detail: charge-exchange, elastic collisions, ionization, dissociation, as well as reflection from the walls. The usage of EIRENE comes with two downsides. The Monte-Carlo nature of the algorithm introduces stochastic noise into the source terms of the plasma equations. To reduce the level of noise, a high number test particles is necessary, which requires longer compute times. In this study a much simpler fluid description of the neutral gas is used. It is implemented as the "standard" fluid neutral model in B2.5. Details of the neutral model implementation can be found in [84]. Running the code in fluid neutral mode drastically lowers the compute time of a simulation [141]. While the results are less realistic than with the kinetic neutrals, the general trends and physical regimes are recovered in a similar fashion [142]. In this neutral model similar particle and momentum conservation equations as for the plasma are solved. As defining difference to the plasma equations, the transport coefficients are determined by neutral instead of coloumb collisions, and the diffusive flux is not density but pressure driven (e.g. $\tilde{\Gamma}_{ay} = -D_{p,a}\frac{1}{h_y}\frac{\partial p_a}{\partial y}$). In this simple model also the ion temperature is shared between the neutrals and ions.

If the mean free path of neutral particles is short a fluid description of the neutral gas becomes more approriate. But even in such conditions, the implementation of the "standard" fluid neutral model is not complex enough to cover all relevant aspects, which lead to the development of the "Advanced Fluid Neutral" (AFN) model [143]–[145] in B2.5. As the AFN model was not available during the course of this thesis, the older "standard" fluid neutral implementation is used. Nevertheless most scenarios still require a kinetic neutral description [146], [147] for quantitatively accurate results.

In addition to the changes in neutral transport, the standard fluid neutral model also neglects any molecular effects. For low plasma temperatures at the divertor targets ($T_e < 3\,\mathrm{eV}$), plasma-molecule interactions play an important role for the power and particle balance in the divertor [148]. Neglect or misrepresentation of these effects has a strong influence on the macroscopic phenomena in detached divertor conditions such as the particle flux rollover [149]. Therefore, significant deviations between the fluid neutral simulations and experiments can be expected.

Both the plasma as well as the fluid neutral model use flux limiters to constrain the appearing fluxes based on kinetic correction terms [84], [150].

The B2.5 equations are solved using a finite-volume numerical scheme on a discrete computational grid [151]. The system of equations is solved using the standard SIMPLE solver with a LU matrix pre-conditioner [152]. This uses an implicit scheme to evolve each equation forward in time. Because the implicit timestep is performed for each equation individually, a self-consistent plasma evolution over time is only possible by iterating over all equations multiple times during a single timestep or selecting very small timesteps to ensure mutual consistency. Only then the residual of the implicit matrix solver will remain small during the time evolution. This makes time-dependent simulations especially difficult and computationally expensive. Nevertheless, SOLPS-ITER is used for such applications [84], [153], [154]. In the majority of applications, such as in this thesis, SOLPS-ITER is not used to produce time-dependent results but rather to discover plasma states that are in steady state with no temporal changes. To achieve this, the code performs a similar evolution forward in time until such a state with no temporal changes appears. In that case the consistency of all equations and minimization of the solvers residuals is only important in the final state but not necessary during the (pseudo-)temporal evolution. There is no universally approved criterion to determine whether convergence to a steady state is reached, but a variety of metrics are used by researchers [80], [82]. Certainty, that the found solution is really a valid steady-state result, can only be ensured when the temporal changes and the solver residuals are minimized to the levels of floating point number precision. Specifically with a kinetic neutral model

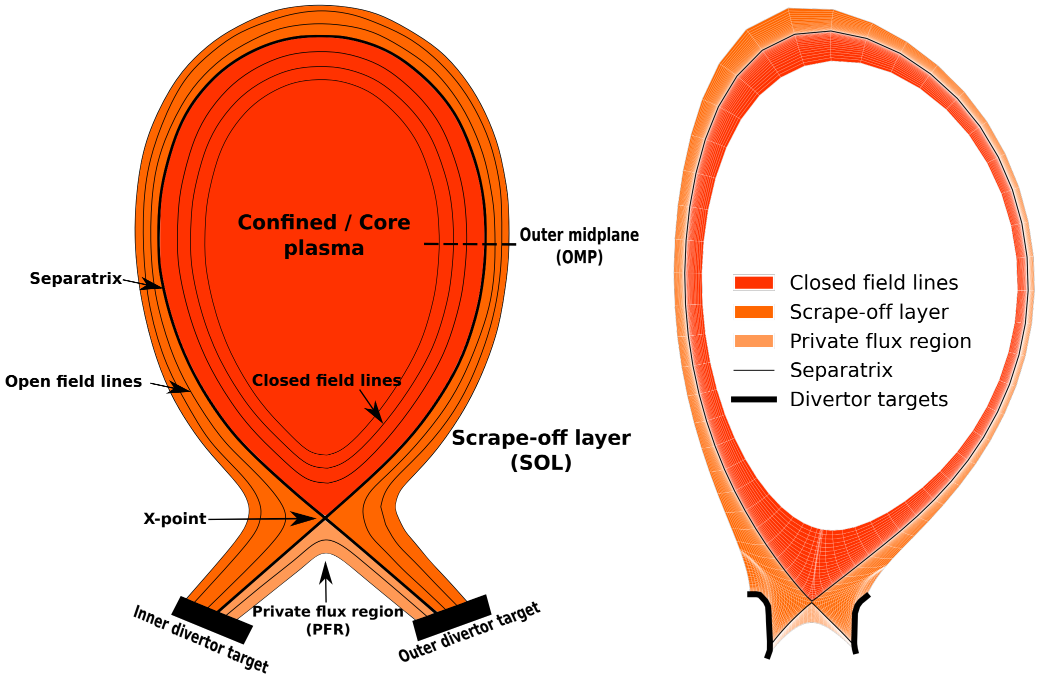


**Figure 2.1:** Schematic concept of the scrape-off layer in divertor tokamaks (left) and the respective simulation grid used in a SOLPS-ITER simulation (right).

this proves difficult due to statistical noise introduced by the Monte-Carlo solver [80]. The inclusion of cross-field drift can create similar difficulties [155]. Therefore, in practice numerical tricks are used to accelerate convergence [82], and less strict metrics are used to classify simulations as converged [141]. Once a converged solution is found, additional tests might be required to determine the sensitivity of that solution regarding the resolution of the computational grid or other numerical parameters (i.e. a solution should not change when going to finer grids or smaller timesteps) [82].

The 2D computational grid used inside B2.5 consist of quadrilaterals that are shaped such that one direction is parallel to the magnetic field and one is perpendicular (see Figure 2.1). The sizes of the quadrilaterals expand when the distance between neighbouring field lines increases. Therefore, a series of grid cells following a magnetic field line constitute a flux tube such that plasma flow parallel to the magnetic field remains entirely in this series of grid cells. The typical computational grid contains parts of the core plasma inside the separatrix where the magnetic field lines form closed loops (see Figure 2.1). This is done because it is easier to define sensible boundary conditions inside the confined region than directly at the separatrix. The grid also covers all field lines that start and end at the divertor targets. This contains the main SOL and divertor areas but also parts of the private-flux region (PFR), which is below the X-point (in single null configurations, see Figure 2.1) and has no adjacent regions to the confined plasma. For numerical reasons the computational grid cannot contain areas that are further outside, such as the far-SOL, where the magnetic field lines start and end at other reactor wall elements besides the divertor targets. In a newer version of B2.5 this option is included but comes with its own numerical difficulties [156]. Because of these limitations, the grid has artificial boundaries, which do not correspond to the actual reactor walls except at the divertor targets. Therefore, proper boundary conditions need to be set not only at the divertor targets but also at the boundaries of the computational grid in the confined plasma, at the far-SOL and inside the PFR (see Figure 2.1). The boundary conditions for the B2.5 equations are implemented by adding small boundary cells around the computational grid so called guard cells. These cells are so narrow that the plasma flow between these guard cells is negligible. The boundary conditions themselves are then implemented by enforcing predefined flow between the guard cells and the actual neighbouring grid cells. There are many predefined options in the SOLPS-ITER software which calculate the necessary fluxes based on used defined choices for common boundary conditions. Table 2.1 summarizes those boundary conditions that are used in the later parts of this thesis. When and how these conditions are used in each case can be found in Chapters 4, 5 and 6.

| Name | Formula | Description |
|---|---|---|
| Constant | $\varGamma^{\mathrm{BC}}$ | The flux is predetermined by the user. Usually by specifying a total input of particles or power that is equally distributed across the boundary area. |
| Feedback | $\varGamma^{\mathrm{Feedback}}$ | A feedback condition that adapts the flux at the boundary to achieve a pre-defined flux across a specified surface inside the computational domain. In this thesis this is only used to prescribe fluxes one row inward from the actual boundary in the confined plasma. |
| Weak | $\varGamma^{\mathrm{BC}} = h\left(\theta - \theta_{\mathrm{BC}}\right)$ | $h$ determines the strength, $\theta_{\mathrm{BC}}$ is the value for some state variable that should be obtained, $\theta$ is the state variable at the boundary. For $h \to \infty$ this corresponds to a Dirichlet condition. |
| Leakage | $\varGamma_a^{\mathrm{BC}} = \alpha c_{s,a} n_a$ | A leakage of particles out of the domain proportional to the local soundspeed $c_{s,a} = \sqrt{\gamma T_a / m_a}$ and density $n_a$. |
| Decay | $\varGamma_x^{\mathrm{BC}} = -\frac{1}{L} D_{\perp,a} n_a$<br>$\tilde{q}_{e/i} = -\frac{1}{L}\chi_{e/i} T_{e/i}$ | The length $L$ is defined by the user. Usually used at the far-SOL and PFR boundaries to model an exponential decay. |
| Sheath | $\varGamma_a^{\mathrm{BC}} = c_{s,\mathrm{avg}} n_a$<br>$\tilde{q}_e = \left(\gamma_e T_e + e\Phi\right)\Gamma_e$<br>$\tilde{q}_i = \sum_{z_a \neq 0} \gamma_i T_i \Gamma_a - \tilde{q}_{i,\mathrm{rec}}$ | The conditions at contact with the divertor targets to emulate the plasma sheath (see Section 2.2). The outflow of ions occurs with sound speed $c_{s,\mathrm{avg}} = \sqrt{\frac{\sum_{z_a \neq 0} \gamma T_a n_a}{\sum_{z_a \neq 0} m_a n_a}}$. It is usually assumed that all ions recycle to neutral gas particles, so an identical influx of neutrals occurs. With the settings used in this thesis the ions and neutrals share the same temperature thus the heat flux is affected by the recycling flux of neutrals $\tilde{q}_{i,\mathrm{rec}}$. |

**Table 2.1:** A short summary of the boundary conditions used in SOLPS-ITER in this thesis. Their exact occurances are given in Table 5.1 and Table 6.1. A complete description of all terms and their specific implementation can be found in the SOLPS-ITER documentation.

# Chapter 3

# Introduction to Machine Learning

What is today commonly coined artificial intelligence (AI) is in its core Machine learning (ML). Although the application of ML methods has seen rising popularity in the natural sciences and applications for the general public, the fundamentals of ML are not yet a standard constituent of the curriculum in university level physics programs. This section should provide readers unfamiliar with ML an introduction to the basic concepts of ML with an emphasis on the aspects used within this thesis.

This chapter relies in most parts on [157], any information given without explicit citation can be found in this or similar standard textbooks (such as [83], [158], [159]).

## 3.1 General introduction

### 3.1.1 Conceptual background

ML focuses on the development of algorithms and models that enable computers to learn from data and make predictions or decisions without explicit programming. Unlike traditional rule-based programming, where explicit instructions are provided to solve a specific task, machine learning systems learn patterns and relationships directly from data. As introduced in Section 2, traditional modeling in the natural sciences involves constructing a mathematical and/or numerical model based on assumptions about the underlying processes. The construction of these models is difficult as it requires a deep understanding of the system being modeled and in many cases it is a priori unclear which assumptions are valid or even useful, and even simple assumptions can invoke complex unforeseen outcomes. In contrast, machine learning emphasizes data-driven modeling and pattern recognition. Instead of explicitly encoding the rules or equations governing a system, machine learning algorithms automatically extract patterns and relationships from data. This enables the discovery of rules that might be challenging to articulate through traditional modeling approaches. This is advantageous in situations where the underlying processes are complex, nonlinear or not fully understood. For ML models the process of learning from data is called training. During the training process an algorithm modifies the values of free parameters in the ML model until this model shows the desired results.

However, there are also disadvantages coming with machine learning in contrast to traditional modeling approaches. Many ML models, especially complex ones like deep neural networks (Section 3.2), can be considered "black boxes". For such models it is often difficult or downright impossible to interpret or explain what patterns or rules the models rely on. Therefore, it is more difficult to anticipate how such a model might behave in changed situations. It is an ongoing research effort in the field of "Explainable AI" to develop ML model architectures and methods that can deliver an introspection into the working principles of such models. While in traditional modeling the modeler has fine control on the rules of the model, in ML the modeler has generally less control on the patterns the models learn. This can lead to problems when undesirable patterns are learned. Examples are patterns which are only specific to the examples seen in the training data but which are not general properties underlying the modeled phenomena. Another example are patterns which reproduce unwanted biases present in the training data such as the widely reported reproduction of discrimination based on ethnicity or gender [160]. Many well established (traditional) numerical algorithms (e.g. for solving differential equations) have known properties. For example, some algorithms are known to preserve certain symmetries or conservation laws. Basic ML models rely on statistical principles therefore they can easily violate these laws. It is an object of current research to design ML system that inherently preserve the intended physical symmetries [161]. Finally, ML can only be applied in fields in which data exists and the quality of the ML model is tightly coupled to the quality and quantity of the underlying data. Therefore, classical "thought experiments" that deliver new insights through induction are not possible but instead data needs to be provided either through experiments or simulations.

Even if a sufficient amount of high quality data is available, the success of a machine learning system is not guaranteed. The "No Free Lunch" (NFL) theorem suggests that there is no single algorithm that excels in every conceivable scenario [157]. Different machine learning algorithms may perform well in certain problem domains but not as effectively in others. While there is some debate whether this can be universally proven [162], in practice it is certainly the case that different problems require different models. It is the task of the scientist to make assumptions about which models are likely to perform well for the specific problem and to test, compare and improve these.

### 3.1.2 Common machine learning tasks

Generally machine learning algorithms can be categorized based on the type of supervision they get during training [157].

In supervised learning, the algorithm undergoes training on a labeled dataset. This means the data consists of pairs of input and output quantities. The objective is to establish a mapping between inputs and outputs, which allows the algorithm to make predictions for the output belonging to new unseen input data. A typical task that can be solved using supervised learning is classification. The task for the algorithm is to categorize input data into a fixed set of classes. Exemplary applications include spam detection or detecting objects on images. Another task that is typically solved using supervised learning is regression. Here the task is to predict a continuous target variable given the input data. Example applications are the prediction of prices or measurement values. This thesis is working on a regression problem using supervised learning methods.

Unsupervised learning entails training the algorithm on an unlabeled dataset where input data lacks explicit desired outputs. The objective is to unveil patterns or relationships within the data without predefined guidance. Typical tasks include clustering, where similar data points are grouped together based on inherent similarities, and dimensionality reduction, which aims to reduce the dimensionality of data while preserving essential information. Applications are the detection of anomalies/fraud or the visualization of high-dimensional data.

Reinforcement learning introduces an agent which interacts with an environment and learns to make decisions through feedback in the form of rewards or penalties. The goal of the agent is to acquire a strategy that maximizes cumulative reward over time. In this pursuit it adapts its behavior through trial-and-error interactions with the environment. Applications are the creation of controllers for robotics or the magnetic control of tokamak plasmas [163].

### 3.1.3 Overfitting and Underfitting

In machine learning in general but most predominantly for supervised learning tasks, the goal is to create models that incorporate generalized patterns. So although the model is trained on a finite set of training examples, it should be equally accurate if applied to different examples that follow similar rules. Two opposing problems can occur: underfitting and overfitting. Underfitting happens when a model is too simplistic such that it is failing to capture the underlying patterns. Overfitting occurs when a model learns the training data too well, capturing noise and peculiarities that do not generalize to new examples. To diagnose and mitigate these problems ML researchers split their data into a training, validation and testing set. The training set is used in the training algorithm to learn the model parameters. The validation set serves as an independent dataset to diagnose potential overfitting or underfitting. Overfitting can be easily recognized by a models excellent performance on the training data but poor performance on the validation set. A sign of underfitting is consistently low accuracy both on the training and validation set. The model developer can then adjust the model architecture or the training procedure to circumvent both situations (see Section 3.1.4). Underfitting usually occurs with simple models, thus it can be addressed by increasing model complexity or choosing more powerful model architectures. Overfitting occurs for highly adaptable models, and can be mitigated by constraining the adaptability of the model (called regularization), simplifying the model or using more training data. To obtain a rough estimate whether a model is more likely to over- or underfit, the number of free parameters adaptable during training can be compared to the number of training data points. Models with few parameters such as linear regression tend to underfit, while models with many parameters like neural networks tend to overfit. Finding the right level of model complexity is an optimization of the trade-off between different sources of model error (known as bias-variance trade-off, e.g [83]). Taking the accuracy on the validation data as guideline, the researcher might implicitly overfit the model also to the validation data. To obtain an unbiased evaluation of a models performance, a dedicated holdout test set is crucial. On this test set only the final models are evaluated such that the generalization error on truly unseen data can be reported for users of the model or as a reference for other researchers.

### 3.1.4 Hyperparameters

The model architecture and training procedure are defined in hyperparameters, that are not learned from the training data but set prior to training. As described in the previous section, selecting appropriate hyperparameters is vital for achieving optimal model performance. Optimizing the hyperparameters is often done through manual experimentation, domain expertise or systematic search algorithms. Two common search algorithms are grid search and random search. Grid search tests all combinations of predefined values for the hyperparameters, while random search randomly samples hyperparameters from a pre-defined distribution. With each selected hyperparameter configuration, the model is trained on the training data and then evaluated on the validation data. This process gets repeated with varying hyperparameter configurations to find the one yielding the best performance on the validation set. To avoid overfitting to the validation set, specifically for smaller dataset sizes, such searches are usually conducted using k-fold cross-validation instead of using a single dedicated validation set. In k-fold cross-validation, the dataset is randomly divided into k equally large groups, so called folds, and the model is trained and validated k times, each time using a different fold for evaluation and the remaining folds for training. The overall score is then calculated as the mean of the scores for each fold. The algorithm is visualized in Figure 3.1.

This thesis uses random search and cross-validation for hyperparameter optimization, the exact procedure, considerations and preliminary tests for this approach are presented in Section 8.1.

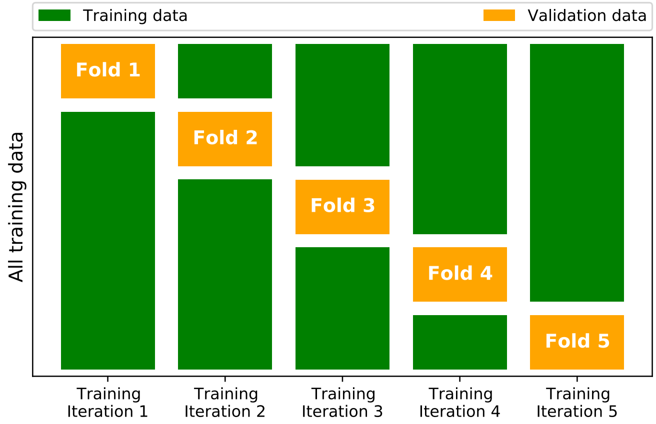


**Figure 3.1:** Depiction of the general process used in a 5-fold cross-validation.

### 3.1.5 Pre-processing

Before data can be used to train a machine learning model, often some pre-processing of the data is required. This is especially the case for experimental or real-world data, which can include measurement errors or other sources of outliers or data in different formats, units or from different sources. For our application only the aspect of scaling is relevant. Most ML algorithms perform worse if the numerical data has vastly different scales. Depending on the algorithm this can affect both the input as well as the output quantities. For example, in regression tasks many algorithms try to minimize a distance metric between the model predictions and actual results. If the output values of the samples in the training data stretch across vastly different scales, single extreme samples could outweigh the effect of the entire remaining training data.

The most common types of scaling are min-max scaling (also called normalization) or standardization. In min-max scaling the minimum value of the data is substracted from each datapoint, which is then divided by the difference between maximum and minimum value $x_{i,\text{normalized}} = \frac{x_i - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})}$. This scales the data to values between 0 and 1. Here $x_i$ is one scalar value from all data samples $\mathbf{x}$. In standardization, the mean of the data is first subtracted and then the datapoints are divided by the standard deviation of the data $x_{i,\text{standardized}} = \frac{x_i - \mu(\mathbf{x})}{\sigma(\mathbf{x})}$. The scaled data has mean 0 and standard deviation 1. For vastly differing scales it can be necessary to first apply a non linear function, such as the logarithm, to the data. In this thesis instead a quantile transformation is used. This scaling procedure first estimates the cumulative distribution function of the data $F(\mathbf{x})$ and then applies this to each datapoint to transform them to a scale between 0 and 1. Then to transform these values to a normal distribution, the quantile function $G^{-1}$ (inverse of the cumulative distribution function $G$) of a standard normal distribution is applied. Thus the scaled data points $x_{i,\text{scaled}} = G^{-1}(F(x_i))$ follow almost exactly a normal distribution. Because it is a rank based method, it is very robust against outliers and works across large scales, but in the process correlations and distances between datapoints are non-linearly distorted (see Section 8.1).

In a dataset, each entry might contain several properties. For example in an experiment or simulation, for each trial multiple properties can be measured and recorded. In machine learning, each trial can be viewed as an independent datapoint, that consists of several data categories, so called features. If different features are different physical properties with different units (e.g. densities and temperatures), they are usually scaled independently. But for similar features with the same units (e.g. two temperatures measured at different locations), it needs to be decided by the modeler whether the statistics for scaling should be common across these features or calculated independently (see Section 8.1).

Because the scaling of the data already incorporates some information about the data statistics, it is important to compute the statistics used in the scaling solely across the training set and not the test set (see Section 3.1.3).

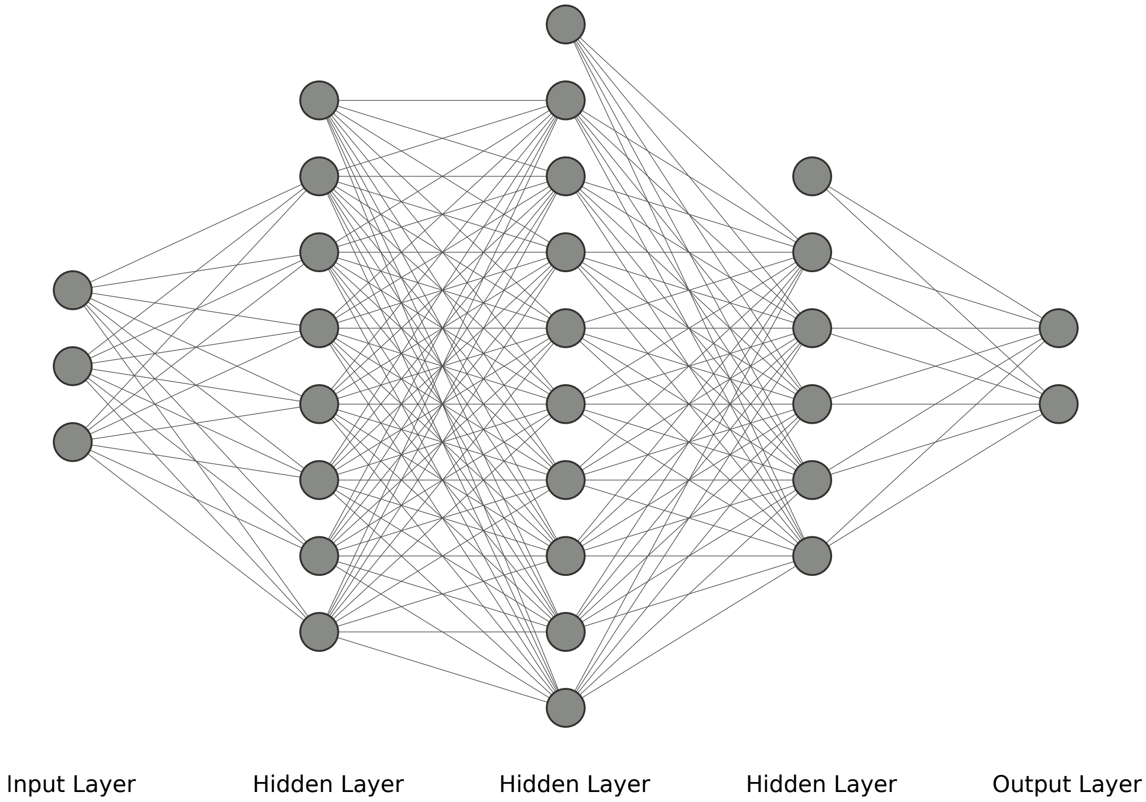


**Figure 3.2:** Visualization of an example fully connected neural network. Each circle represents a neuron, while the connecting lines represent the weighted sums. The neurons without inputs in the hidden layers are the bias terms $b_{ij}$. For a set of values inserted in the input layer $\mathbf{x}_0$, the network computes the output $\hat{\mathbf{y}}$ according to Equation 3.2. The result is seen as output values of the neurons in the output layer.

## 3.2 Neural networks

Neural networks are a fundamental paradigm in machine learning and currently responsible for breakthroughs in many tasks. At their core, these networks consist of interconnected nodes called neurons, which are usually organized into layers. The original inspiration was mimicking the working principles of biological neuronal networks, but todays artificial neural network architectures and learning algorithms differ greatly from those present in nature. The earliest architectures are more than 60 years old, but the breakthroughs only came when efficient algorithms for the training of neural networks with many layers of neurons became available. Such networks are called deep neural networks and hence the field of their design, training and application is called deep learning. The working principles of neural networks can be explained at the simple example depicted in Figure 3.2. This network has three inputs in the input layer and two outputs in the output layer. It therefore requires three numbers as input and provides two distinct outputs. Mathematically the neural network is a vector function $f : \mathbb{R}^3 \to \mathbb{R}^2$. Connecting the input to the output layer are the hidden layers, each with a varying number of neurons. The actual computation underlying this graph representation is given in Equation 3.1.

$$x_{ij} = \sigma_{ij} \left( \sum_{k=1}^{N} w_{ijk} x_{i-1,k} + b_{ij} \right) \tag{3.1}$$

The output $x_{ij}$ of the neuron number $j$ in the $i$th layer is computed as the result of its activation function $\sigma_{ij}$. This activation function receives as input the weighted sum of the outputs of all neurons in the previous layer and the bias term $b_{ij}$. In Figure 3.2 this is depicted by the connections between the neurons, and the bias terms are the extra nodes in the hidden layers that do not receive inputs. If we make the assumption that inside each layer the neurons have the same activation function $\sigma_{ij} = \sigma_i$ (which usually is the case), then for the specific network from Figure 3.2 a single output vector $\hat{\mathbf{y}}$ is computed from the input $\mathbf{x}_0$ according to Equation 3.2.

$$\hat{\mathbf{y}} = \sigma_4 \left(\mathbf{b_4} + \mathbf{W}_4\sigma_3 \left(\mathbf{b_3} + \mathbf{W}_3\sigma_2 \left(\mathbf{b_2} + \mathbf{W}_2\sigma_1 \left(\mathbf{b_1} + \mathbf{W}_1\mathbf{x}_0\right)\right)\right)\right) \tag{3.2}$$

Training the network to represent a desired target function now resolves to finding the right values for the weight matrices $\mathbf{W}_i$ and bias vectors $\mathbf{b}_i$.

In this thesis, only fully connected layers (also called dense layers) are used. Each neuron in such a layer receives input from all neurons in the previous layer. Such networks are called fully connected networks, dense networks or Multi-Layer Perceptron. Other network architectures can be achieved by putting restrictions on the weight matrices $\mathbf{W}_i$. Examples are setting some weights permanently to zero (sparsity) or using the same value at different positions in the matrix. The second can be specifically used to implement desired symmetries in the network. Most notable examples of this are convolutional networks, which have shown huge successes in image related tasks due to their inherent translational invariance. Training a neural network comes down to an iterative optimization of the network weights and biases to minimize a predefined cost function. In practice, first the weights are randomly initialized and then the network is used to make predictions on training examples. These predictions are then evaluated using the cost function and the result is called the loss. Given that the loss is calculated from the network predictions $\hat{y}$, the weights of the network are updated using gradient descent.

$$w_{ijk} \rightarrow w_{ijk} - \alpha \cdot \frac{\partial \text{Loss}}{\partial w_{ijk}} = w_{ijk} - \alpha \cdot \left(\frac{\partial \text{Loss}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_{ij}}\right) \tag{3.3}$$

Equation 3.3 depicts this weight update with a learning rate $\alpha$. This is only possible if the loss is a differentiable function of the network prediction. Using the chain rule the derivative of the loss with respect to any network weight can be calculated (see Equation 3.2). In common neural network software like TensorFlow or PyTorch, this is done efficiently by calculating the gradients using automatic differentiation, that performs the chain rule on low level operations so no analytical calculations of the gradients are required. Nevertheless, for large networks the calculation of the gradients can become enormous. Therefore, these software libraries use the backpropagation algorithm, which progresses backwards through the neural network and reuses the previously calculated derivatives for efficiency. Stochastic gradient descent (SGD) is used to avoid getting stuck in local minima of the loss function. Modern algorithms improve upon SGD, in this study we use the optimizer Adam [164].

Given these equations one can already see a bunch of hyperparameters that are not learned by the model itself, but need to be prescribed. The number of neurons in the input and the output layer are usually determined by the problem to be solved (although different formulations of the problem can also allow for variations in these, as is shown in Section 8.2). The **number of neurons** in the hidden layers can be freely chosen. Similarly, also the **number of hidden layers** is a parameter to choose.

Also the **learning rate** introduced above and the **loss function** used in the optimization need to be prescribed. Examples of common loss functions are the Mean Squared Error (MSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE):

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{3.4}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{3.5}$$

$$\mathrm{MAPE} = \frac{100}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{3.6}$$

Here $y_i$ are the actual outputs in the training examples and $\hat{y}_i$ are the corresponding network predictions. The MSE is very sensitive to large errors and penalizes large outliers significantly more than the MAE. The MAPE mitigates scale dependency but can be problematic with small values.

The training algorithm usually iterates several times through the whole dataset. Each full pass through the dataset is called an epoch. While the losses and the consecutive weight updates can be computed across a whole epoch, it is in practice far more efficient to compute the loss on smaller batches and thus update the weight multiple times during an epoch. Following this approach, the number of samples in each **Mini-Batch** is another selectable hyperparameter.

The **activation function** of the neurons in the hidden layers needs to fulfill two key requirements: differentiable and non-linear. As can be seen in Equation 3.2, besides the activation functions the neural networks consists only out of linear operations. If the activation functions were linear, the whole neural network could be reduced down to a single linear matrix multiplication and loses its advanced non-linear modeling capabilities. In this thesis three commonly used activation functions are tested: Rectified Linear Unit (ReLU), Exponential Linear Unit (ELU) and Scaled Exponential Linear Unit (SELU).

$$\mathrm{ReLU}(x) = \max(0, x) \tag{3.7}$$

$$\mathrm{ELU}(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha \cdot (\exp(x) - 1), & \text{if } x \leq 0 \end{cases} \tag{3.8}$$

$$\mathrm{SELU}(x, \lambda, \alpha) = \lambda \cdot \begin{cases} x, & \text{if } x > 0 \\ \alpha \cdot (\exp(x) - 1), & \text{if } x \leq 0 \end{cases} \tag{3.9}$$

$$\tag{3.10}$$

ReLU replaces negative values with zero, allowing only positive values to pass through. It is computationally efficient and helps mitigate the vanishing gradient problem. However, it may suffer from the "dying ReLU" problem, where neurons become inactive during training. Although it is not differentiable at zero, this poses not a problem in practice and ReLU has become the standard activation function. ELU introduces a non-zero slope for negative inputs, which helps alleviate the dying ReLU problem. It allows negative values to pass through, preventing neurons from becoming inactive during training. The hyperparameter $\alpha$ controls the functions behavior for negative inputs. SELU is an extension of ELU with a self-normalizing property, promoting stable activations and mitigating the vanishing/exploding gradient problem. The hyperparameters $\lambda$ and $\alpha$ control the mean and standard deviation of the activations. SELU has shown to be particularly useful in deep neural networks. The reasoning behind the SELU loss function is that while in the first hidden layer the neurons receive normalized input, deeper in the network the neurons receive input that might follow a totally different distribution. Using the SELU loss the mean and standard deviation of the input to each neuron in deeper layers remains standardized, when the weights are initialized correctly [165]. Non-linear activation functions can in general be used in both the hidden and the output layer, but in this thesis the activation functions are only applied in the hidden layers. Therefore, the neurons in the output layer perform only a weighted linear combination of the results from the last hidden layer.

**Batch normalization** [166] is another method frequently used in training deep neural networks. Like the SELU activation function it has the goal to maintain stable statistics for the inputs to the deeper layers in neural networks. It does this by first rescaling the inputs to each neuron to mean 0 and standard deviation 1 and then applying a shift and rescaling to a finite mean and variance. This shift and rescaling factor are learnable parameters that are optimized like the network weights during training. The standardization happens across each Mini-Batch and is therefore only an approximate to the true mean of the activations. Like in the original paper [166], in this thesis Batch normalization is always applied before the activation functions.

Deep neural networks consist of thousands or millions of trainable parameters and can represent almost any function. They are therefore very prone to overfitting. To counteract this, different strategies for regularization are commonly employed.

One of the most common strategies to avoid overfitting is using early stopping. In this method the error of the networks predictions on the validation set is continuously tracked during training, and the training is stopped once the network starts to improve solely on the training set and not anymore on the validation set. This check is usually done once per epoch, so once for every pass through the entire training data. A hyperparameter that controls early stopping is the **patience**. It determines the number of consecutive epochs in which the validation error does not improve until the training is stopped. The model is then reset to the state of best validation error. Since training with variants of stochastic gradient descent is noisy, using too small patience values will lead to bad model accuracy and setting too large values will lead to overfitting to the validation set. While early stopping prevents the generation of strongly overfitted models, it does not steer the network training itself to avoid overfitting.

One class of techniques aiming to do exactly this modifies the loss function with additional terms to restrict the adaptive freedom of the neural networks weights. Most common are **L1 and L2 regularization**. In both approaches a norm of the weight vector (containing all weights of the network) is added to the loss function: $\mathrm{Loss} + \lambda \left\|\mathbf{w}\right\|_{1/2}$. The methods only differ in the type of norm used $\left\|\mathbf{w}\right\|_2 = \sum_{i=1}^{n} w_i^2$ (L2) or $\left\|\mathbf{w}\right\|_1 = \sum_{i=1}^{n} |w_i|$ (L1), and both require a prescribed scaling factor $\lambda$ that determines the strength of this additional loss term in comparison to the traditional cost function.

Another popular strategy to avoid overfitting in neural networks is called **dropout** [167]. During training a random group of neurons in the network is disabled (output set to zero), which effectively alters the network architecture at each iteration. This forces the network to be less reliant on specific neurons and produce more robust predictions. The technique is typically only applied to hidden layers. When the network is finally used for predictions after training, all available neurons are used for maximum precision but with rescaled weights to maintain the activation strengths. As hyperparameter the dropout rate needs to be provided, which determines the fraction of neurons that is disabled in each pass.

All these methods are already implemented in common deep learning software libraries such as PyTorch [168] or TensorFlow [169].

## 3.3 Gradient Boosted Regression Trees

Besides neural networks, this thesis explores Gradient Boosted Regression Trees as surrogate model architectures. Boosting belongs to the class of ensemble methods. The fundamental idea of ensemble methods is to combine the predictions of many models, which might not be particularly good on their own, into a combined far better prediction. In Boosting predictors are trained sequentially, such that each new predictor tries to correct its predecessors [157]. The two most common methods are AdaBoost [170] and Gradient Boosting [171]. In gradient boosting a base model is trained on the training data and predictions on this data are made. The following model is than trained to predict the residual errors of these predictions. The final prediction is then computed by summing the predictions of all consecutive models, but the impact of each following model generation is reduced by a learning rate factor compared to the previous stage. Because the following models that are added to the ensemble have smaller and smaller impact, gradient boosting is less prone to overfitting, which makes it particularly suited for small datasets. But the method only works if the base models used at each stage are rather weak. If the early models already overfit the data, the consecutive models have no residuals to improve upon. The most common base models in use are decision trees. This yields the name Gradient Boosted Trees or Gradient Boosted Regression Trees (GBRT) for regression tasks.

A decision tree model splits the data into different groups based on criteria evaluated on the input values. The predicted values are formed by calculating the mean values between all training samples that where categorized into one group. Figure 3.3 shows an example of such a decision tree, which was trained to predict the electron temperature at the separatrix at the outer divertor target. In this model the training simulations are separated into several groups based on strengths of the deuterium and nitrogen sources. New predictions for a set of input parameters are made by categorizing it according to the splitting rules and assigning it the value of its final group. As an example, the model in Figure 3.3 predicts for any simulation with an deuterium ion influx $\Gamma_{D^+}$ of more than $10^{22}$ atoms/s an outer target temperature of 381 eV. Training a decision tree corresponds to finding the optimal criteria for splitting the training data. Finding the exact optimal tree is a computationally impossible task even for small datasets. Instead most algorithms which determine the splitting criteria operate in a greedy fashion, by choosing each split independently after another. Even then numerous different algorithms exist to approximate the best splits. Although decision trees in itself can output multiple values, the GBRT implementations used in this thesis (scikit-learn and XGBoost) can only deal with scalar outputs. To predict multiple values, multiple GBRT models are constructed in parallel. Other frameworks e.g. CatBoost offer also multioutput GBRT [172].

The hyperparameters of GBRT are either hyperparameters determining the shape of the decision trees itself or controlling the boosting procedure. Similar as for neural network, a **loss function** needs to be chosen to quantify the differences between the model predictions and the training data. For regression the same loss functions as introduced above (MSE or MAE) can be used. The **learning rate** determines the relative impact of each decision tree on the final ensemble result. The optimal value

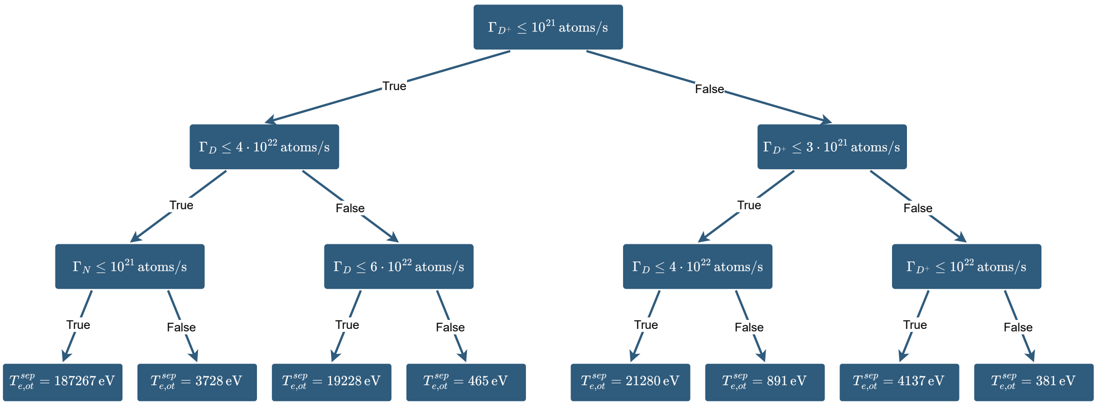


**Figure 3.3:** Visualization of a decision tree model. New cases are sorted based on their $\Gamma_{D+}$, $\Gamma_D$, $\Gamma_N$ values following the splitting rules from the top to the bottom of the decision tree. The model predicts a temperature $T_{e,\mathrm{ot}}^{\mathrm{sep}}$ for each example based on the value of the group it was sorted into. Training such a model corresponds to finding optimal rules to sort the training data.

is closely linked with the total **number of trees** in the ensemble. Similar as the number of training epochs for neural networks, this number could be determined by early stopping. However, in this thesis it is treated as regular part of the hyperparameter optimization. Another key parameter is the **subsampling fraction**. Using subsampling the decision trees at each level of the GBRT model are not trained on the full training data but a random subset. This procedure is sometimes also called "stochastic gradient boosting". Regarding the individual tree structure, the most important parameter is the **tree depth**. It determines how many levels of splitting are allowed in the tree. The example in Figure 3.3 has a depth of three.

# Chapter 4

# Setup of a simulation case

To develop a surrogate model for the scrape-off layer in tokamaks, first a suitable simulation database is necessary. While there are some existing SOLPS-ITER simulation databases, most are unsuitable for surrogate development as they contain only simulations for very specific scenarios and no broad consistent parameter variations. Therefore, it is paramount for this study to create its own dataset.

This requires first a choice of the parameters which are to be varied in the dataset and second a consistent simulation setup that allows for these variations. The former defines the scope of the surrogate model and its applications. The goal of this thesis is to provide a proof-of-concept that is not particularly designed for only a single application but which can be used in a variation of contexts. Because fusion power plants need to be larger in size than current tokamak experiments [37], a key requirement for the surrogate model is the possibility to vary the tokamak size. This way the model can be validated against existing experiments, and be used in design studies for future reactors. Another requirement is that all major physical regimes are contained in the model. Because it is a priori entirely unclear how many simulations are required to train such a SOL surrogate successfully, we focus only on the most relevant parameters, and reduce the computational complexity to allow for faster simulation runtimes. The parameters controlling a SOLPS-ITER simulation can be broadly categorized into three groups (Table 4.1). The parameters determine either the specifics of the machine or discharge that is modeled, the assumptions and models used in the physics description or the specifics of the numerical implementation of these models. It is not intended to vary the numerical parameters of the

| **Machine or discharge** | **Physics model** | **Numerical model** |
|---|---|---|
| • Reactor size and shape | • Neutral transport models | • Grid discretization |
| • Magnetic field strength | • Anomalous cross-field transport | • Initialization |
| • Plasma shaping | • Sheath transmission models | • Timestep |
| • Divertor geometry | • Boundary conditions | • Convergence metric |
| • Heating power | ... | ... |
| • Fueling rates | | |
| • Impurity seeding rates | | |
| • Pumping speed | | |
| ... | | |

**Table 4.1:** Exemplary selection of the parameters defining a SOLPS simulation.

simulations as part of the surrogate model but rather a single setup should be found that is suitable for all simulations. The difficulties associated with finding such a setup are depicted in Chapter 5.

The following sections describe exactly which parameters were implemented in a simulation setup to allow for consistent variations. Section 4.1 explains the approach taken here to incorporate a cross machine size scaling in the simulation database. Then section 4.2 gives a brief overview of the reference simulation that is altered to create the simulation setup used here. In Section 4.3 the changes to the neutral model and the respective choices for the fueling, seeding and pumping model are explained. Finally, Section 4.4 describes the assumptions which are made regarding the anomalous cross-field transport.

## 4.1 Size scaling

A surrogate model is only usable for reactor design if it allows variations in the reactor size and shape. To allow for an extrapolation from present to future experiments, both should be contained in the applicable range of the surrogate model, which is determined by the extent of the underlying simulations.

The overall tokamak size is determined by the major radius of the torus. The machine size is crucial for the dynamics in the scrape-off layer because it determines the connection length along a magnetic field line $L$ between the hot upstream conditions at the confined core plasma and the downstream conditions at the divertor targets [16]. This length scales linearly $L \propto R$ with the major radius of the tokamak torus $R$. Just varying this length alone has a strong impact on the conditions at the divertor targets (see Equations 2.15-2.17 in Section 2.3). The machine size is highly correlated with the design and geometry of the underlying magnetic field. So differently sized experiments also have differing magnetic topologies. To include machine size variations in a simulation database, in principle two approaches are possible. One is to use the geometries of existing or planned reactors with varying sizes, such as Alcator C-MOD, ASDEX Upgrade (AUG), JET, JT60SA, ITER and DEMO. The other approach is use one tokamak geometry and scale it up and down in size. Both approaches have their specific advantages and drawbacks. In the first approach, already existing simulation cases could be used, and the results of the final surrogate will match the existing experiments. This means simulations of AUG size would also have AUG geometry and simulations with JET size the JET geometry. In the latter approach, the training simulations would resemble conceptual but not existing tokamaks. This means simulations of AUG size would not have the exact AUG geometry, which will make them less applicable and less comparable to the existing experiment. However, the former approach has the large disadvantage that parameters cannot be varied individually. As can be seen in Figure 4.1, changing from one of these experimental tokamak designs to another, changes not only the machine size in terms of major and minor radius, but also many properties of the overall geometry (e.g. plasma triangularity, elongation, ...). In this approach it is impossible to vary solely the size alone, which will make it impossible to determine which of the effects stem from the changes in size and which originate due to the changes in the other properties. This makes the predictions of the surrogate less

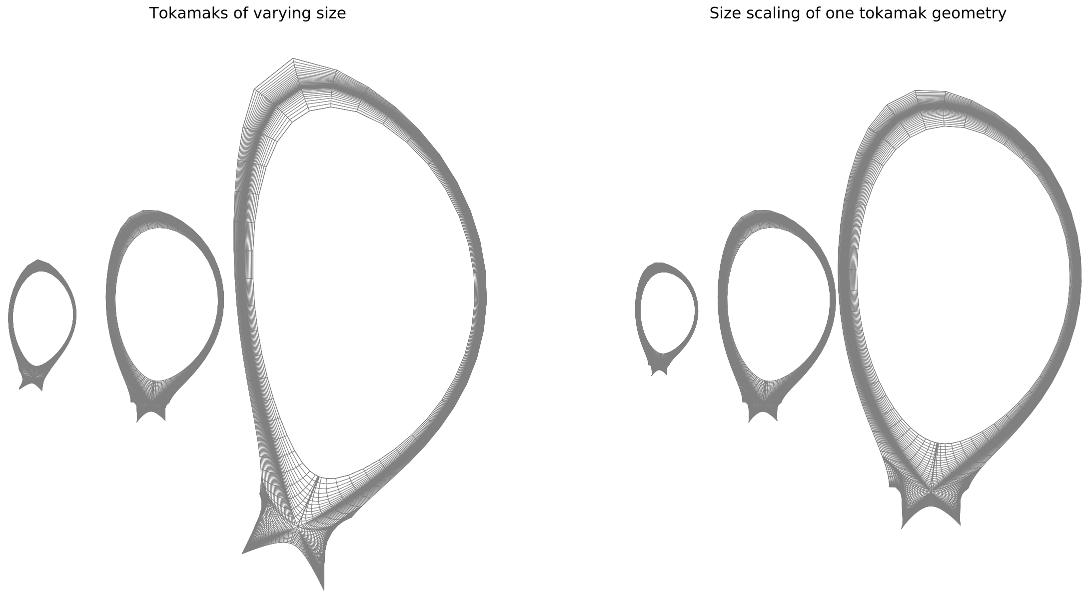


**Figure 4.1:** Sketches of the B2.5 computational grids in the two general approaches to tokamak size scaling. On the left are the grids for different tokamaks of varying sizes (ASDEX Upgrade, JET, ITER). On the right is the grid of one tokamak (JET) scaled up and down in size. The latter approach is chosen for the simulation database as it allows continuous size variations and separates influences of the tokamak size from other geometric effects.

generalizable due to the implicit influences of the exact geometries. Similarly, when developing and training the surrogate model, it will be unclear which underlying parameters need to be chosen for the surrogate input, as many are highly correlated. Exactly this is one of the difficulties when using experimental data for model development, which we would like to avoid in the surrogate. The decision here is to go forward using the latter approach using a conformal scaling, which allows for a clearer understanding of the individual influences of the parameters.

The transport in the SOL is mainly parallel to the magnetic field lines. Numerical stability of SOLPS-ITER simulations requires that the computational grid underlying the finite volume method is perfectly aligned to the magnetic field lines [84]. In this case one direction along the grid is always parallel to the magnetic field lines while the other direction is perpendicular. The equilibrium magnetic field is not self-consistently calculated in SOLPS-ITER but needs to be provided as input, which requires running another set of complex external codes. Also the process of generating a computational grid for SOLPS-ITER based on this equilibrium is in itself not straightforward. The grid generation algorithms from the SOLPS-ITER code suite usually do not provide a usable result straight out of the box but requires hand tuning to ensure proper alignment. Manual inspection and adaptation is not feasible for thousands of simulation grids. Until more reliable automated grid generation algorithms are available, an approach using conformal size scaling needs to be restricted to scalings that leave the computational grid unchanged. Changes to the tokamak size can be implemented by scaling an existing computational grid up and down in size. But such grid transformations are only sensible when the resulting magnetic fields are valid and could appear in a tokamak.

The magnetic field in tokamaks is determined by the vacuum magnetic field generated by the coils surrounding the tokamak and the electric current inside the plasma. From the balance between the force upon the plasma by the magnetic field $\overrightarrow{j} \times \overrightarrow{B}$ and the pressure gradient inside the plasma $\nabla p$, the Grad-Shafranov (GS) equation can be derived [13]. While solving the equation is complicated, simple analysis shows that the shape of the magnetic field is invariant to some transformations. The following summarizes the derivation of these properties made in [173].

The Grad-Shrafranov equation can be written as:

$$\frac{\partial^2 \psi}{\partial R^2} - \frac{1}{R}\frac{\partial \psi}{\partial R} + \frac{\partial^2 \psi}{\partial Z^2} = -\mu_0 R^2 \frac{dp}{d\psi} - \frac{1}{2}\frac{dF^2}{d\psi} \tag{4.1}$$

Here $\psi$ is the flux function for the magnetic field, $p$ the plasma pressure and $F$ the flux function for the current. $R$ and $Z$ are the radius and height coordinates in a cylyndrical coordinate system and $\mu_0$ is the magnetic constant. We are interested in solutions of the GS-Equation $\psi$ that do not change in shape but only by a constant scaling factor. This can be achieved by bringing the equation in a non-dimensional form, using the major radius $R_0$, the vacuum (toroidal) magnetic field at the magnetic axis $B_0$ and the following definitions: $R = r \cdot R_0 \quad Z = z \cdot R_0 \quad \psi = \overline{\psi} \cdot \psi_0 \quad p = p' \cdot B_0^2/\mu_0 \quad F^2 = R_0^2 B_0^2 + R_0^2 B_0^2 f'^2$
$p'$ and $f'$ are the non dimensional pressure and current flux function. Inserting these definitions into the GS equation yields:

$$\frac{\partial^2 \overline{\psi}}{\partial r^2} - \frac{1}{r}\frac{\partial \overline{\psi}}{\partial r} + \frac{\partial^2 \overline{\psi}}{\partial z^2} = -r^2 \frac{R_0^4 B_0^2}{\psi_0^2}\frac{dp'}{d\overline{\psi}} - \frac{1}{2}\frac{R_0^4 B_0^2}{\psi_0^2}\frac{df'^2}{d\overline{\psi}} \tag{4.2}$$

Since $\overline{\psi}$ should not change its form, the right-hand side of Equation 4.2 should remain unchanged under any transformation. Introduction of $\alpha^2 = \frac{R_0^4}{\psi_0^2} B_0^2$ yields the condition:

$$const = -r^2 \alpha^2 \frac{dp'}{d\overline{\psi}} - \frac{1}{2}\alpha^2 \frac{df'^2}{d\overline{\psi}} \tag{4.3}$$

The shape of the solution $\overline{\psi}$ is unchanged when $\alpha^2 \cdot p'$ and $\alpha^2 \cdot f'^2$ remain constant. This can be achieved by keeping $\alpha, p', f'$ constant independently, while changing the real physical pressure $p$ and current $F$. The toroidal $B_t$ and poloidal $B_p$ components of the magnetic field are given by:

$$B_t = B_0 \frac{\sqrt{1 + f'^2}}{r}$$

$$B_p = \frac{\psi_0}{R_0^2}\sqrt{(\frac{1}{r}\frac{\partial \overline{\psi}}{\partial r})^2 + (\frac{1}{r}\frac{\partial \overline{\psi}}{\partial z})^2}$$

Using these it shows that the condition in Equation 4.3 allows for independent variation of the major radius and the toroidal magnetic field strength as long as the aspect ratio $\frac{R_0}{a} = const$ and the ratio of the poloidal to the toroidal magnetic field (e.g. safety factor) $\frac{B_t}{B_p} = const$ remain constant during the transformation.

Implementation of the size scaling requires multiplying all spatial coordinates of the computational grid cells by a fixed scaling factor, and changing the magnetic field strength requires multiplying the magnetic field vectors in each grid cell by a fixed scaling factor. Both is already implemented as an option in the "b2ag" preprocessing routine of SOLPS-ITER.

This conformal size scaling allows us to run SOLPS-ITER simulations with varying tokamak sizes and magnetic field strengths and will be used in the following to create a simulation database. The span of major radii in the simulation database ranges from 1 m to 10 m (see Section 6). This way the database covers similar sizes as in small tokamak experiments like ASDEX Upgrade up to the expected sizes for future experiments like DEMO [37]. But this size scaling prohibits independent changes to the divertor size and geometry or the overall plasma shape. Similarly the safety factor remains unchangeable. Therefore, it is important to choose a suitable reference simulation setup to use in this scaling.

## 4.2 Reference simulation setup

The size scaling procedure described in the previous section requires a single reference simulation setup, whose geometry is scaled up and down in size. Here a SOLPS-ITER simulation of JET is chosen. The simulation is modeled after JET shot number 85423 and has been validated against the experiment. The simulation stems from a similarity study between AUG and JET that was presented in [174]. It models an H-Mode discharge with nitrogen seeding gas and tungsten divertor and beryllium vessel walls. The magnetic equilibrium underlying this simulation was created with EFIT [175]. This simulation setup was chosen for several reasons:

- It uses a single null divertor with vertical targets, which is the relevant configuration for ITER and possibly DEMO.
- The aspect ratio and overall geometry is similar to AUG, ITER and possibly DEMO.
- JET is an existing experiments so using this reference case will allow tests to integrate experimental measurements into the surrogate.
- Compared to other tokamak experiments, it has the largest size so the extrapolation gap to ITER and DEMO is the smallest.
- A tungsten divertor and full metallic walls are used as foreseen for future tokamaks.
- Nitrogen gas seeding was already studied extensively across several tokamak experiments and SOLPS simulations [58], [74], [176]–[179], which allows for validation of the final surrogate model.

Besides these physical reasons, the simulation setup also offers an above average grid resolution with 102x48 grid cells. Since the grid remains unchanged throughout the scaling, it is paramount that the number of grid cells is sufficient for all sizes. It is assumed that this grid resolution is adequate for all later used tokamak sizes since simulations with the used fluid neutrals (Section 4.3) are more resilient against low grid resolutions [180].

At the inner boundary of the simulation grid in the region of closed field lines (see e.g. Figure 4.1), a constant temperature and a constant flux of deuterium ions are prescribed as boundary conditions. The temperature is prescribed identically for the ions and electrons. The values of these boundary conditions will be varied in the simulation dataset to model scenarios with varying heating power and core fueling.

However, there are still some changes to be made to the simulation setup before it can be used for generation of a dataset. First, the present simulation uses a kinetic description of the neutral gas particles, which is more exact but requires longer compute times compared to a simplified fluid description. To ensure that a sufficient number of simulations can be generated, a change to the latter is needed. The changes made are described in Section 4.3. Also this case uses an edge transport barrier to create plasma profiles inside the separatrix which match an H-mode discharge. While this is the foreseen operating scenario for future tokamaks, the configuration of an edge transport barrier requires many parameters and for simplicity it is therefore changed to a uniform transport assumption in Section 4.4.

## 4.3 Fluid neutral model compared to kinetic neutral solution

Using the 2D standard fluid neutral model in the B2.5 transport code, instead of coupling to an external kinetic neutral gas solver like EIRENE [140], [181], drastically lowers the compute time of a simulation by a factor of $10-100$ [141]. But using this model requires more tweaking of the boundary conditions to obtain physically sound solutions [142]. There is no standard procedure for converting a simulation case from kinetic to fluid neutral settings but some guidelines and recommendations are provided in [142]. When changing the reference simulation described in the previous section from kinetic neutrals to fluid neutrals, it makes no sense to observe the neutral properties directly, as there differences are bound to appear, but instead settings should be found in which the plasma properties change as little as possible.

Although the standard neutral model is less precise than the kinetic neutral model it offers the superior speed and should in general also show the same trends as the kinetic neutral models [127], [142], [182], such that we can use this as a first step in the development of surrogate models.

Linked to the neutral model is also the implementation of the gas puffs. In the kinetic neutral model the gas puffs introduce molecular deuterium $D_2$ and atomic nitrogen $N$. In the fluid neutral model only atomic fluids are possible. The computational domain of the kinetic neutral model extends beyond the grid of the fluid plasma solver (which is the same for the fluid neutrals), and includes a large subdivertor area in the simulation chosen here (Figure 4.2). Both gas puffs are located inside the private-flux region, and the pump removing neutral gas particles from the system is a dome like structure below the outer divertor target (Figure 4.2). These structures are not present in the corresponding computational grid of the fluid plasma solver. Because the grid cells need to be aligned with the magnetic field lines and orthogonal to any wall boundaries, which is not the case in these sub-divertor structures, these cannot be included in the standard fluid approach. Several attempts were made to implement the gas puffs in the fluid neutral simulations as boundary condition on the B2.5 grid boundary in the private-flux region (Figure 4.2). But in all these attempts large amounts of particle moved from this boundary across the PFR into the region of confined field lines. There this leads to diverging plasma densities and finally crashing simulations. Instead the approaches used in [127], [141], [180], [182], [183] were followed by implementing both the deuterium as well as the impurity gas puff as boundary condition at the far-SOL boundary of the B2.5 grid (Figure 4.2). This gas puff boundary condition is a constant inflow of neutral gas particles at the far-SOL boundary given as an absolute value, which is equally distributed across the whole boundary. In addition, a leakage of particles is used as boundary condition in the PFR, which acts as particle pump. The results of a simulation with this setup can be seen in Figures 4.3 and 4.4. Overall, both the deuterium and nitrogen ion densities match fairly well in the whole domain but there are still some deviations at the high field side, at the inner target, where both densities are significantly higher in the fluid neutral simulation than in the kinetic neutral setup (Figure 4.3). A similar pattern shows for the electron density and reversed for the electron temperatures (Figure 4.4). Also in the PFR, the kinetic neutral case has higher nitrogen ion densities and lower electron temperatures. These drastic changes at the inner target but also the reduced densities in the PFR likely originate from the changes in gas puff location and neutral transport, which ultimately changes the whole fueling efficiency. The changes in neutral transport might be strongly influenced by the use of a 5-point stencil to calculate particle fluxes in the fluid neutral model [184].

To get a better picture of the differences of the plasma between the neutral models, Figure 4.5 shows the profiles of electron density and temperature at the outer midplane, outer divertor target and inner divertor target. At all three positions the fluid neutral case has generally lower electron density than the kinetic neutral case, although the general shapes of the density distributions are maintained. Similarly, the shape of the electron temperature profile at the outer midplane is maintained, and here the temperatures almost exactly match. While at the inner divertor the temperatures do not match, they are on similarly low scales below 2 eV, showing that in both cases this divertor is fully detached. At the outer target both simulations obtain drastically differing temperatures with a drastically higher peak temperature in the fluid neutral simulation. This might hint at a change in the power distribution such that more heat flows to the outer divertor target and less to the inner.

Here the fluid neutral simulations and the kinetic neutral simulations use the same value for the gas puffs. Due to the different neutral models but also due to the changed position of the gas puffs, the efficiency of how many of these neutral gas particles become ionized and drive the general plasma density or stay as neutral gas particles is likely different. To get a better comparison between the simulations, one

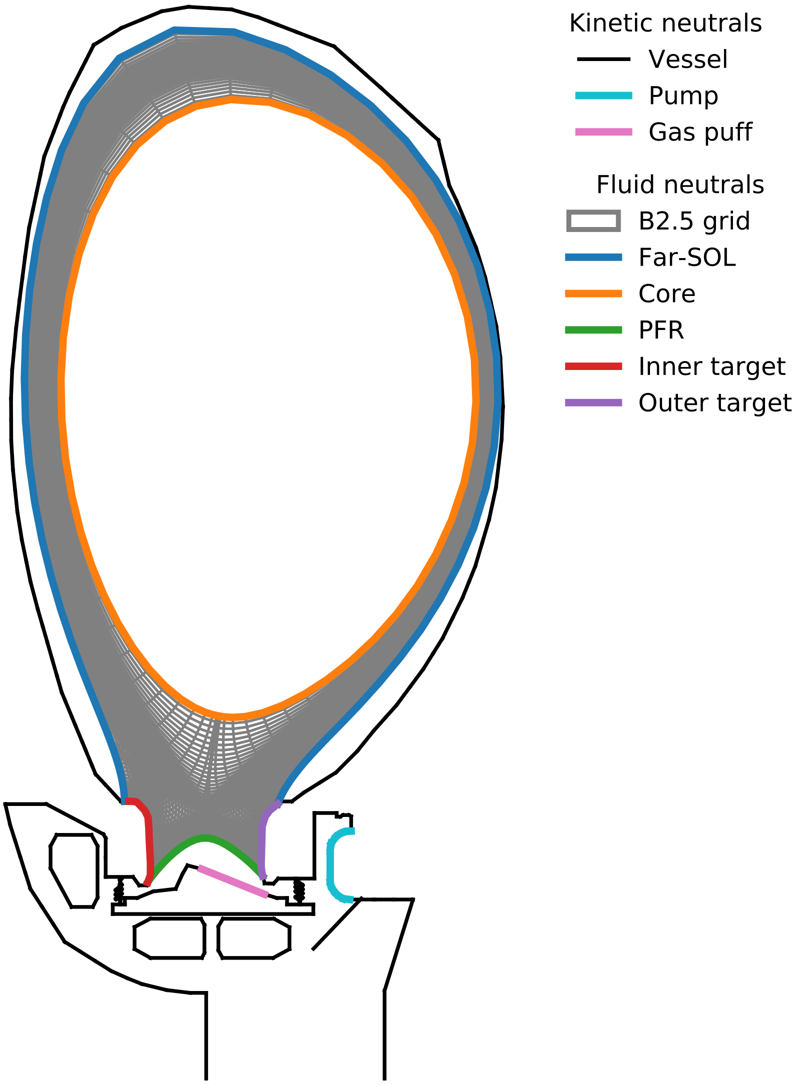


**Figure 4.2:** The 2D cross-section of the JET tokamak used in the SOLPS-ITER simulations. The actual vessel wall and pump and gas puff locations are only used in the original kinetic neutral simulation. The fluid neutral simulations use only the B2.5 grid, with an inflow of neutral atoms at the far-SOL boundary to model a gas puff and an absorption of all particles at the PFR boundary to substitute the pump.

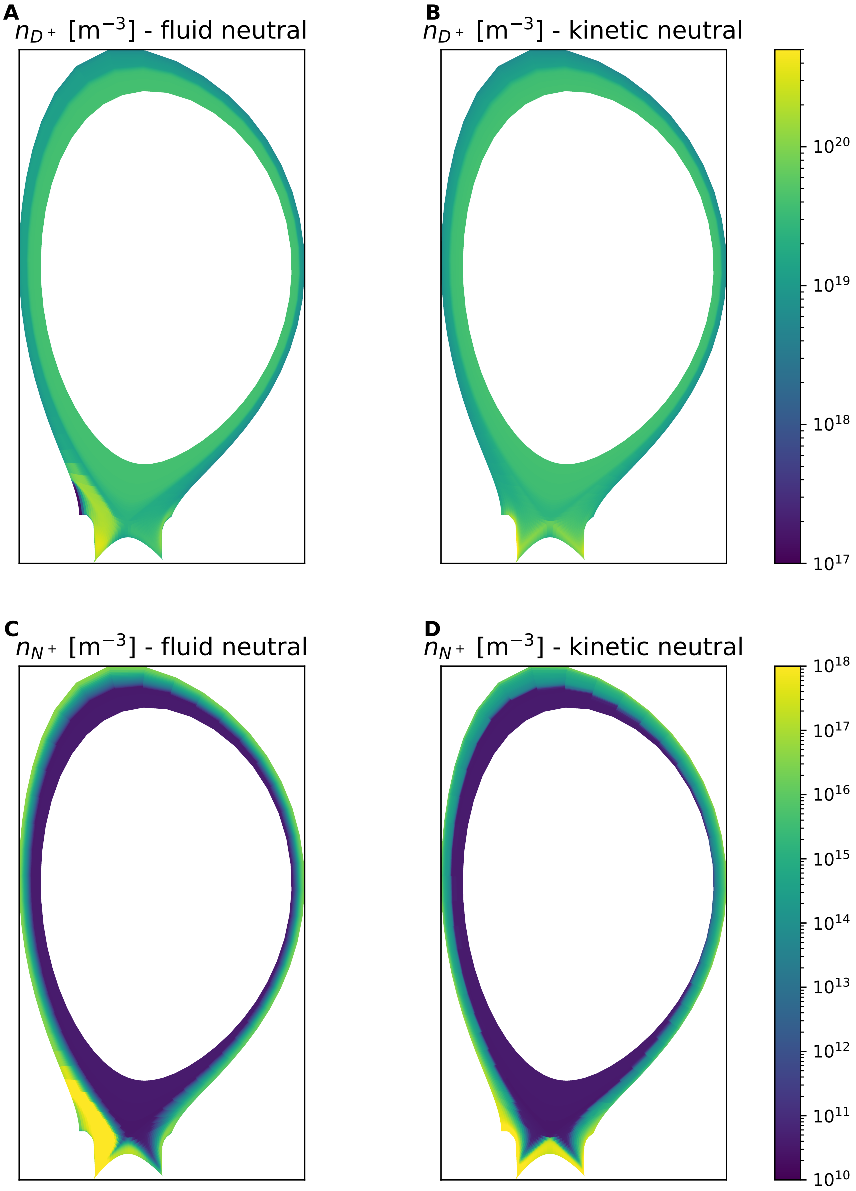


**Figure 4.3:** Deuterium and nitrogen ($N^+$) ion densities in a fluid neutral simulation with deuterium and nitrogen atom gas puff at the far-SOL domain boundary (A, C) and in the reference kinetic neutral simulation with gas puffs in the PFR (B, D).

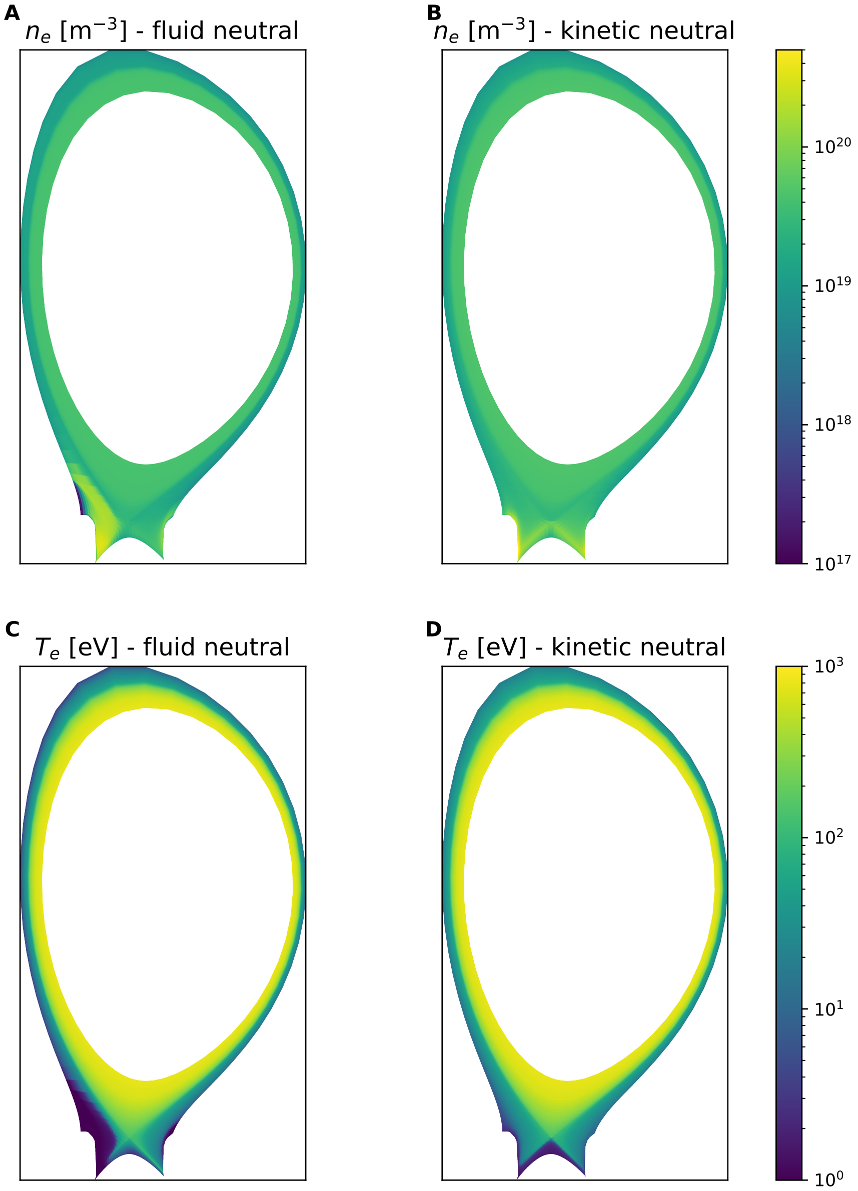


**Figure 4.4:** Electron densities (A, B) and temperatures (C, D) in a fluid neutral simulation with deuterium and nitrogen atom gas puff at the far-SOL domain boundary (A, C) and in the reference kinetic neutral simulation with gas puffs in the PFR (B, D).

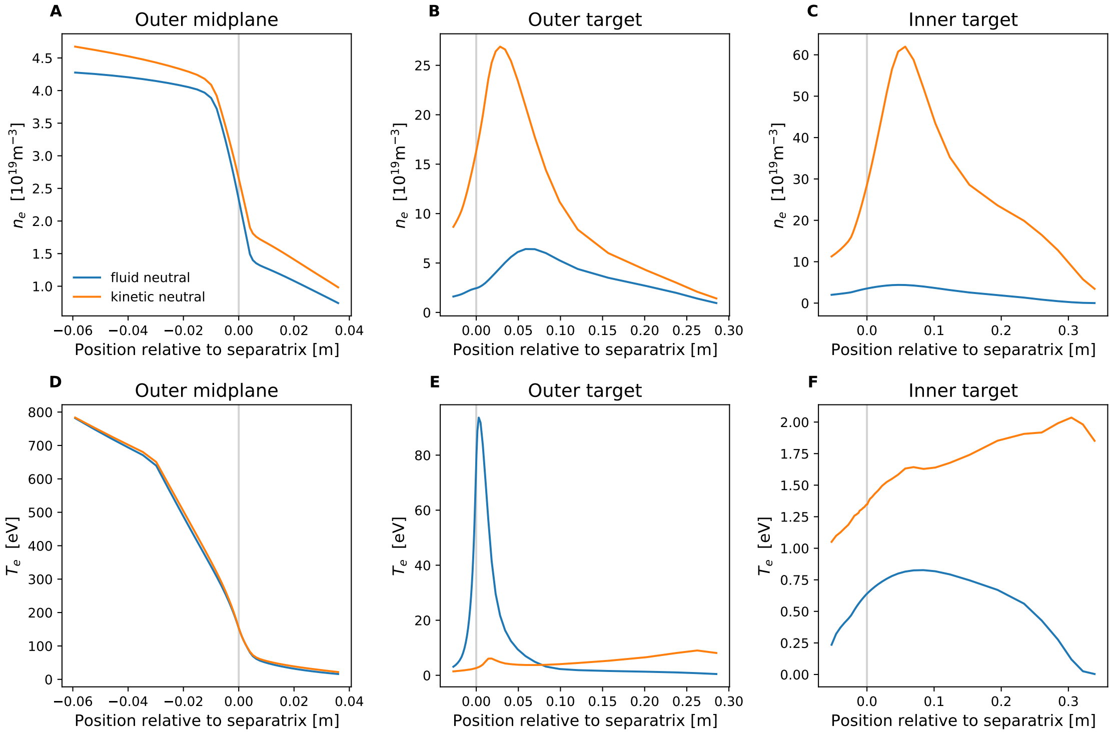


**Figure 4.5:** Electron densities (A, B, C) and temperatures (D, E, F) at the outer midplane (A, D), the outer divertor target (B, E) and the inner divertor target (C, F) in a fluid neutral simulation with deuterium and nitrogen atom gas puff at the far-SOL domain boundary (blue) and in the reference kinetic neutral simulation with gas puffs in the PFR (orange).

would need to rescale the gas puffs to match some specific key physical quantities such as the upstream density or the neutral gas pressure in the divertor. But even then it is questionable how much better the results can be aligned because ultimately it is impossible to bring both neutral implementations to yield the same results in all observables and optimizing one might increase the discrepancy at another place. For example [185] showed that fluid neutral simulations lead to a particle flux rollover at lower plasma densities. In our case, it is not necessary to find a gas puff scaling to improve the one-to-one resemblance because the simulation will be anyway used in a simulation farming procedure with varying gas puffs. As long as the gas puff range in this farming procedure is large enough, such that all scenarios of interest are included, the rescaling can be performed much easier later using the developed surrogate models (see Chapter 10).

## 4.4 Cross-field transport

The cross-field transport is not computed self-consistently inside SOLPS-ITER (Section 2.5) and needs to be prescribed through effective coefficients. These are usually chosen by matching the resulting plasma profiles at the outer midplane to experimental measurements. The strength of the cross-field transport can have a strong impact on the heat flux and particle distribution seen at the divertor targets. While there are some scaling relations for the SOL width [57], which defines the cross-field transport, these come with large uncertainties (e.g. in the dependence on the major radius). It is therefore much more suitable to treat these parameters as unknowns and changeable input of the surrogate, which requires their variation in the SOLPS-ITER dataset. In the reference simulation setup plasma drift effects are neglected but an edge transport barrier is implemented to model H-Mode conditions. This edge transport barrier is a reduction in the cross-field transport coefficients close to the separatrix, that achieves higher plasma gradients at the separatrix as seen in the plasma pedestal in H-Mode discharges [176], [178]. Although operating in H-Mode is the anticipated scenario for future tokamaks, the implementation of the edge transport barrier (coefficient inside, width, depth, coefficient outside) increases the number of free parameters by a factor of 4 (see e.g. [114]) compared to the simpler ansatz of a fixed coefficient for the whole domain. It is therefore chosen to take this simpler approach and keep the cross-field transport coefficients constant in the simulated domain and only vary one parameter for the particle transport of all ions $D_\perp \in [0.1, 2.0]$ m$^2$/s and one parameter for both the electron and ion heat equations $\chi_\perp \in [0.1, 2.0]$ m$^2$/s. While this yields plasma profiles inside the closed-field lines representing an L-Mode discharge, a similar approach was taken in almost all SOLPS simulations designing the ITER divertor [58], [76].

# Chapter 5

# Improvements to the simulation stability

The simulation case developed in the previous section now is completely setup to be used in simulation farming with varying parameters. In total there are eight parameters setup for variation: The size and magnetic field strength, the temperature and the deuterium ion flux at the core boundary, the deuterium and nitrogen gas puff and the two cross-field transport coefficients. For each of these parameters minimum and maximum values were fixed and then 2560 random samples were drawn from a uniform distribution. This procedure distributes the simulations uniformly in the eight dimensional parameter space. However, of the 2560 started simulations a majority crashed within less than 10,000 iterations. Figure 5.1 depicts the dependence of the success rate of simulations against the input parameters. A simulation crash occurs when the values of atleast one of the state variables diverge to extreme values, causing SOLPS-ITER to stop the iteration prematurely. Hence a successful simulation reaches the full 10,000 iterations. If no simulation would diverge Figure 5.1 would depict uniform distributions at 100%. Some level of diverging simulations is anticipated, as seen in similar studies [141], and due to the random combination of parameters, some scenarios might correspond to systems of coupled differential equations that have no solution. But here, most predominantly the number of successful simulations decreases with increasing major radius. While for the smallest size over 80 percent of simulations do not diverge, above a major radius of 8 meters there is almost no simulation that does work (Figure 5.1A). This is surprising as given the scanned parameter ranges there should be reasonable scenarios also for tokamaks of this size such that solutions to the system of equations should be findable. The magnetic field strength, the core temperature or the ion flux at the core boundary seem to have no strong influence on the success rate of simulations (Figure 5.1B, C, D). A strong dependence is instead found with the deuterium gas puff (Figure 5.1E). For low values the success rate is around 30 percent, which drops between $10^{23}$ atoms/s and $10^{24}$ atoms/s dramatically and then rises again for higher gas puff. Similarly for the nitrogen gas puff (Figure 5.1F), an abrupt decline in successful simulations above $10^{21}$ atoms/s is seen. Also the cross-field heat transport coefficient has a small impact on the failure rate of simulations (Figure 5.1H). Increasing this coefficient increases the number of successful simulations.

Especially the almost total lack of successful simulations with large major radius is an intolerable behavior. With a lack of simulations for large tokamak sizes, a surrogate model would be unable to provide reliable predictions in this range of the parameter space, drastically lowering its applicability for future tokamaks like ITER and DEMO, which lie in this size range. Since not all simulations fail but only the ones with larger sizes, there seems to be some problem with the numerical setup that is correlated with the machine size. To investigate the nature behind this behaviour and more importantly to find a fix that allows simulations with increased size, a series of parameter scans was performed with varying setups and objectives. In total more than 40 parameter scans were conducted. The following sections only describe those conducted parameter scans and tests that might be informative for the scientific community.

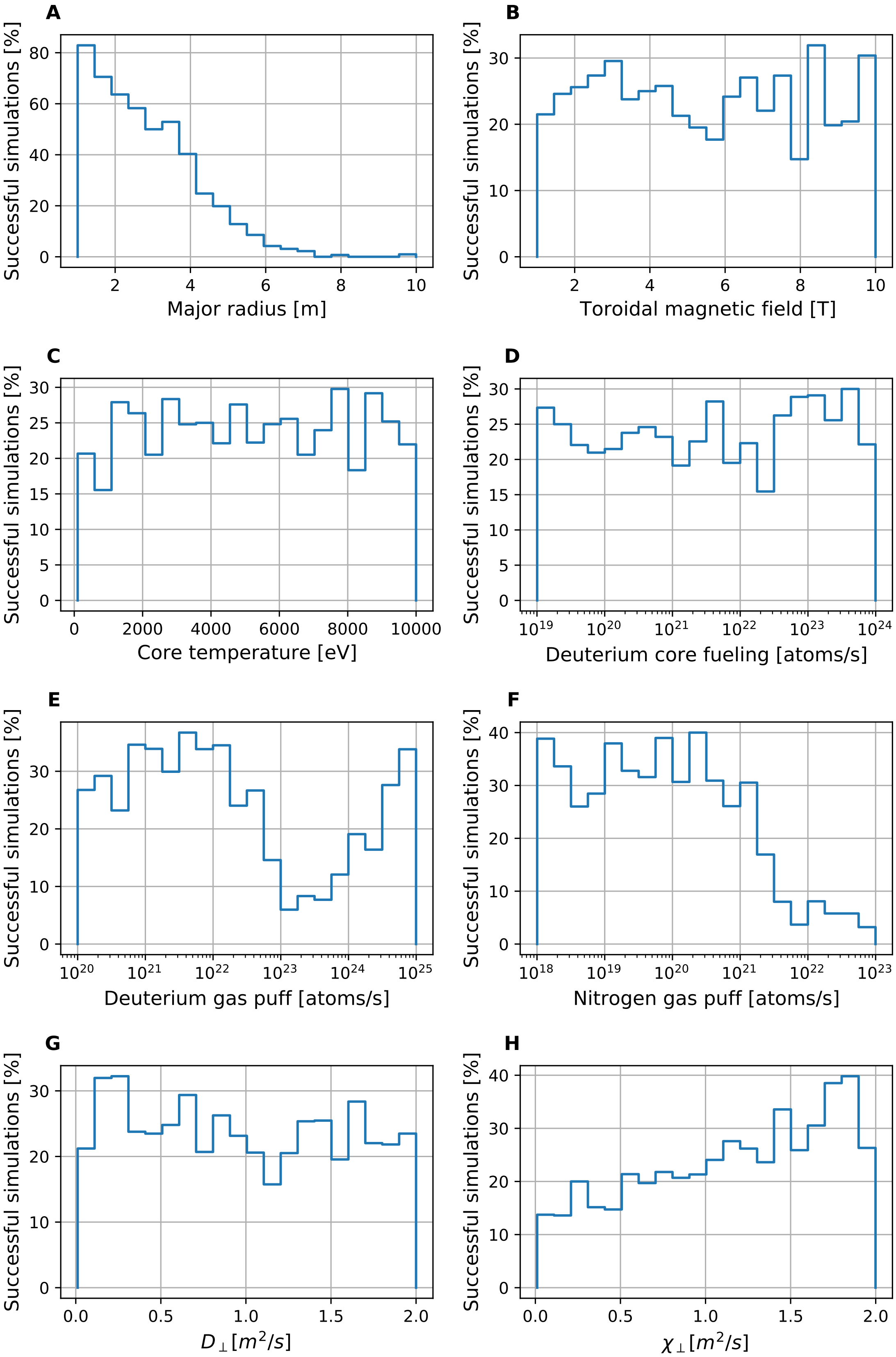


**Figure 5.1:** Histograms of the success rate of simulations with varying parameter values in the first attempted parameter scan. The success rate is calculated by dividing the number of non diverging simulations by the total number of started simulations inside each histogram bin.

## 5.1 Timesteps

One of the central numerical parameters is the timestep that is employed in the pseudo time-stepping routine to run the simulations [186]. While the parameter scan in Figure 5.1 uses the same timestep as in the original kinetic neutral simulation $10^{-5}s$, this parameter scan was repeated with varying timesteps from $5 \cdot 10^{-7}$ s to $1 \cdot 10^{-4}$ s (Figure 5.2). Intuitively one would assume that lowering the timestep would lead to a higher success rate but the results show the opposite. The overall success rate is increased when going to larger timesteps. But the overall trends of simulation failures with the major radius, the deuterium and nitrogen gas puff, and also the anomalous cross-field heat transport still hold for any of the timesteps used. Since the overall increase in the success rate with larger timesteps is small, the desired improvement of achieving more simulations with larger sizes is not reached.

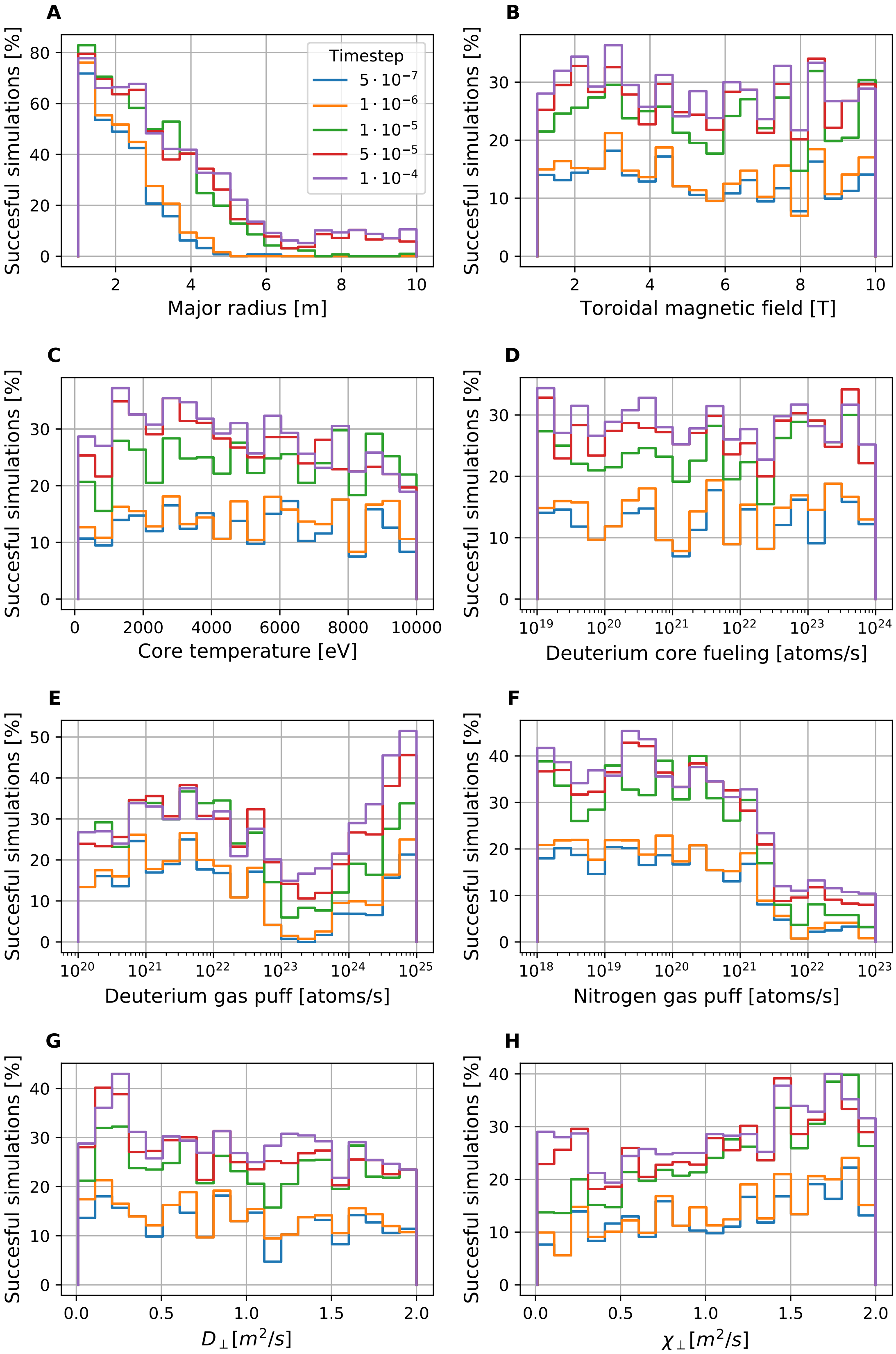


**Figure 5.2:** Histograms of the success rate of simulations with varying parameter values. The parameter scans use identical sets of random parameter values and differ only in the simulation timestep used as denoted by the colors. The tested timestep sizes are $5 \cdot 10^{-7}$ s (blue), $10^{-6}$ s (orange), $10^{-5}$ s (green), $5 \cdot 10^{-5}$ s (red) and $10^{-4}$ s (purple). The success rate is calculated as in Figure 5.1.

## 5.2 Initializer

In the previous tests the simulations were restarted from the final state of the reference kinetic neutral simulation. It is a common procedure to start from such a pre-converged state such that the plasma state is already close to the final converged plasma state and will then converge faster. Here parameter scans were conducted to test the influence of different initial plasma states. The most basic option is to start from the default initial state, in which the plasma density and temperature is uniform in the whole 2D fluid simulation domain. The same parameter scan is thus started from constant densities ($[n_D, n_{D^+}, n_N, n_{N^{x+}}] = [10^{16}, 10^{19}, 10^{12}, 10^{12}]\,\mathrm{m}^{-3}$) and temperatures ($T_e = T_i = 100\,\mathrm{eV}$) in the whole simulation domain. The parallel velocities and all dependent plasma quantities (fluxes, etc.) are set to zero. As before the simulations are started for 0.1 s and the results are depicted in Figure 5.3. Starting from this uniform state has drastically improved the overall number of successful simulations. Although the number of successful simulations with very low major radius around 1 meter has slightly decreased in comparison to before, in the rest of the parameter range it is improved. While there is still the dependency that the number of successful simulations decreases with major radius, the trend is softened such that for any tokamak size the ratio of running simulations to attempted simulations never falls below 30% (Figure 5.3A). The improvement in stability with this initial state could be due to either the uniform state truly being better or the initial densities and temperatures being reduced compared to the previous start state. To test whether just decreasing the densities and temperatures has a similar effect, we started from the same pre-run initial state that was used before, but reduced the densities and temperatures everywhere by a factor of 10. Running the same parameter scan starting from this condition for 0.1 seconds, we find that this has an extremely strong effect on the dependence of the simulation failure rate with the major radius. Doing so there is no more tendency of increasing failure for larger sizes and for any size the success rate is above 40% (Figure 5.3A). To exclude the possibility that the lower failure rates are just achieved because it takes longer too reach the point of divergence due to the lower starting densities, we restarted both tests with uniform and reduced starting conditions and ran them for 1.0 s of simulated time. Only this time the densities and temperatures are reduced by a factor 100 in the reduced case. This longer run time has a dramatic effect on the success rate of the pre-run state, which is now lowered to less than half of the value observed in the test before in all parameter ranges (Figure 5.3A). For the uniform initial state, the longer runtime also decreases the success rate but not to the same extent. Also it shows a clear pattern that for small tokamak sizes the success rate remains almost unchanged while for larger sizes it decreases. The dependencies of the success rate on the simulation parameters are, however, different between the pre-converged state with reduced temparatures and densities and the uniform state. The success rate with the pre-run state has no strong dependence on the tokamak size and the success rate even rises a bit for larger sizes. Instead starting from uniform conditions yields a drastic decrease in success rate for larger sizes. Because the success rate for small sizes is a lot higher, the success rate for the largest sizes still remains above 10% (Figure 5.3A). Concerning the nitrogen gas puff and core temperature, both schemes lead to decreasing success rates for higher parameter values (Figure 5.3C, F). Starting from a uniform distribution the success rate has no correlation with the deuterium ion influx (Figure 5.3D), while the success rate decreases in the pre-run case for core ion influxes above $10^{23}$ (Figure 5.3D). Over the whole parameter space starting from a uniform initial state leads to a success rate of 31% while the pre-run states with reduced properties yield only 21% successful runs.

While the histograms in Figure 5.3 depict the single parameter correlations with the success rate, it is apparent that not only a single parameter is responsible for the simulation failure but higher order correlations are relevant. The most predominant effect is seen for combinations of certain tokamak sizes and deuterium gas puffs. Figure 5.4A depicts how the combination of both parameters affects the success rate of the simulations started from the uniform state. It is apparent that the reduced success rate for larger tokamaks sizes appears only for increased values of deuterium gas puffs while for lower gas puffs a large major radius leads to running simulations. When starting with the reduced pre-run setting (Figure 5.4B), the picture becomes more complicated. Only the simulations of the larger radii and with low deuterium gas puff seem to run consistently, and smaller tokamak sizes will fail unless the deuterium gas puff value is increased. Since the simulations were done in both cases with exactly the same parameters and only the starting conditions were changed, we can compare one-to-one which of the simulations failed in one setup and worked in another. This is shown in Figure 5.4C. Overall the majority of simulations diverged with either of the starting conditions. The largest group of running simulations runs exclusively with the uniform starting conditions while also a large fraction works with both starting conditions. The simulations that only run when started from the pre-run state are almost exclusively located in the range of major radius above 5 meters and deuterium gas puffs below $10^{22}$ atoms/s.

Because the uniform initial state yields the overall highest success rate yet, we continue with this starting condition, but from the findings here it is apparent that any choice of the initial state results in diverging simulations that might be successful with other starting conditions. To have the highest turnout of non diverging simulations it is advisable to use a selection of initial states such that simulations failing with the uniform initial state are restarted using different starting conditions. Also the analysis has so far only considered whether a simulation diverges or not. It was not investigated whether the initial conditions have any influence on the final state that is obtained once the simulations converge to a steady state.

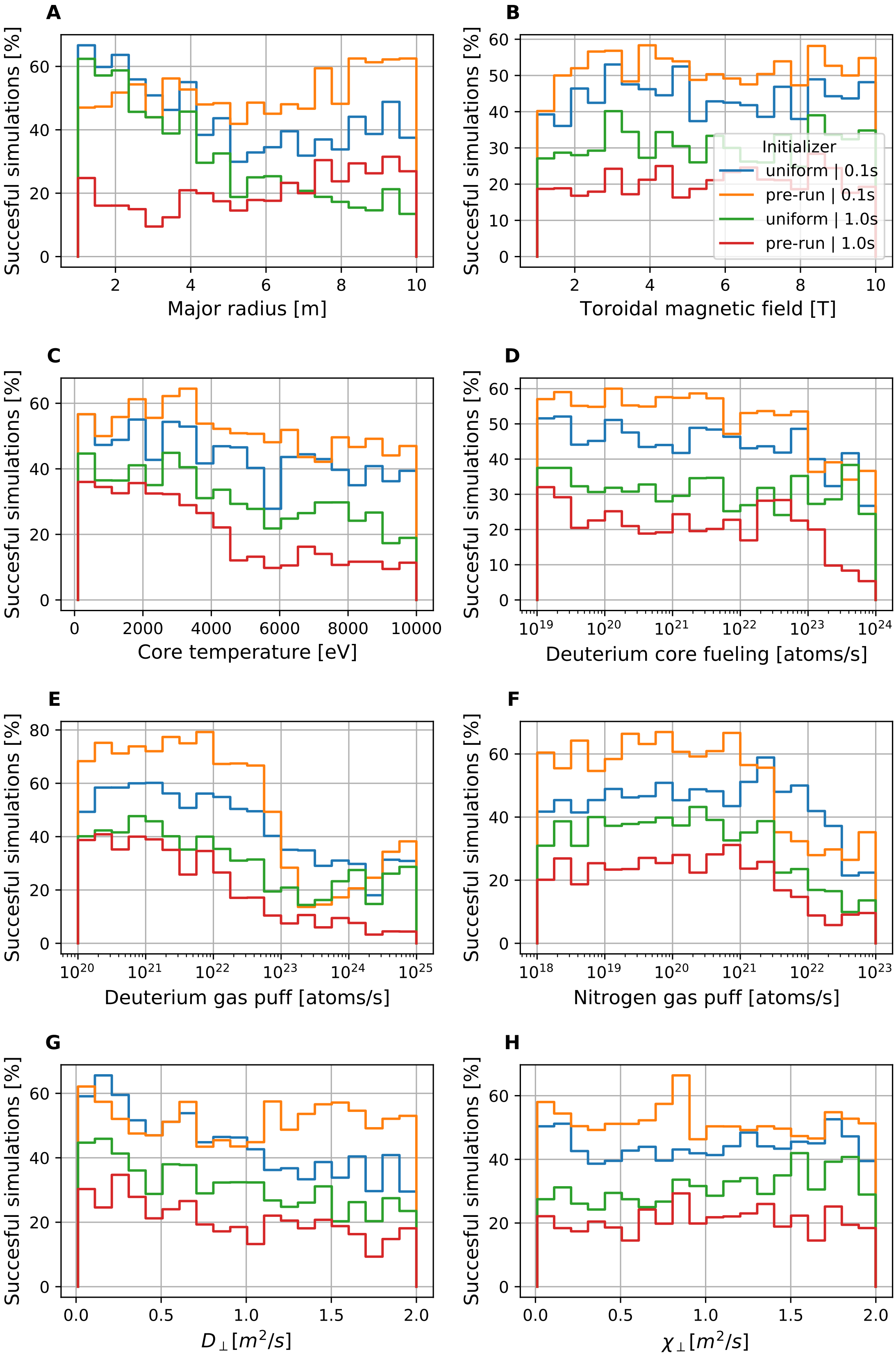


**Figure 5.3:** Histograms of the success rate of simulations with varying parameter values. The parameter scans use identical sets of random parameter values and differ only in terms of the initial plasma state used and the simulated time as denoted by the colors. The simulations are started either from a uniform plasma state for 0.1 s (blue) or 1.0 s (green) of simulated time or from a pre-converged plasma state for 0.1 s (orange) or 1.0 s (red) of simulated time. All simulations use the same timestep of $10^{-5}$ s. The success rate is calculated as in Figure 5.1.

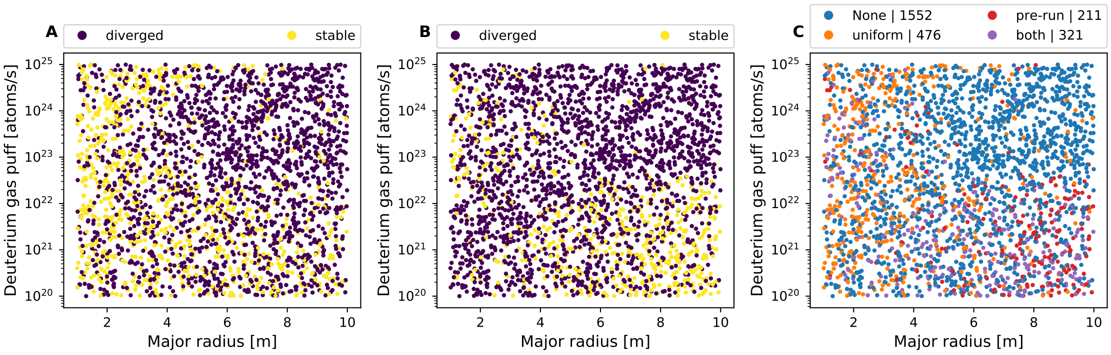


**Figure 5.4:** Each dot represents a started simulations from the parameter scans with 1 s runtime with a given deuterium gas puff and major radius. The simulations were either started from the uniform initial state (A) or the pre-run state with reduced densities and temperatures (B). Figure C depicts which of the started simulations run solely when started from the uniform or the pre-run initial state or from either or diverge in both cases (None).

## 5.3 Numerics

In Section 5.1 the effects of different timesteps on the success rate of the simulation farming procedure were investigated, and it was found that increasing the timesteps has a small positive effect on the success rate. This should be further investigated here more deeply. While in Section 5.1 only the overall timestep was modified, independent scaling factors for this timestep can be applied to the equations for the fluid neutrals, namely the neutral continuity (Equation 2.25) and the neutral momentum equation (Equation 2.26). The goal is to find parameters that decrease the failure rate specifically for cases with larger major radius. Instead of doing a whole parameter scan through the whole tokamak parameter domain, a single parameter configuration with high major radius and large deuterium gas puff is selected, such that it lies in the region of most failures (see Figure 5.4). This case was run using the previous settings of timestep $10^{-5}s$ and neutral timestep multipliers of 1.

Varying both the timestep and the neutral multipliers for this example case leads to the results shown in Figures 5.5 and 5.6. The results depicted Figure 5.5 stem from simulations that were started from the previous initial state (as done in Figure 5.1) while in Figure 5.6 the simulations were started from the initial state with uniform temperatures and densities. Starting from the pre-converged state leads to the same conclusion as in Section 5.1 that increasing the timestep leads to fewer diverging simulations while with decreasing timesteps more simulations fail (Figure 5.5). The influence of the neutral timestep multiplier is small as with smaller timesteps all simulations fail regardless of the multiplier value and for larger timesteps the majority of simulations run with any multiplier value. In all these cases, except for the timestep $5 \cdot 10^{-6}s$, the simulations run with the default multipliers of 1 or not at all. Starting from the uniform initial state shows absolutely different relations (Figure 5.6). For smaller timesteps the simulation runs, while for larger timesteps it diverges, until with timestep $5 \cdot 10^{-4}s$ the simulation runs again. For this large timestep, the influence of the neutral timestep multiplier on the divergence of the simulation is rather obscure and no clear pattern is recognizable (Figure 5.6H).

The increased stability with smaller timesteps in simulations starting from uniform distributions might be intuitively understandable. Starting with uniform density and temperature, no spatial gradients exist. A large timestep might lead to a too rapid development of gradients, that become too large and lead to divergence. While with smaller timesteps, the gradients can develop more slowly and strong gradients do not lead to diverging transport so fast. But this is just a hypothesis as no in-depth analysis into the mechanisms of the divergences is conducted here. Regardless of the timestep value all tests were run for the same total number of timesteps. Therefore, the simulations with smaller timesteps were run for shorter physical time. It could be that starting from the state with uniform spatial distributions the smaller timesteps only appear more stable because they have not been run long enough to reach the point of divergence.

Since the increase in success rate with using the uniform initial state (Figure 5.3) is higher than the effect seen when using larger timesteps (Figure 5.2) and both cannot be reliably combined together, it is better to stick solely to the uniform initial state. It appears that starting from this initial state a modification of the timestep for the fluid neutral equations in comparison to the plasma equations has no added benefit (Figure 5.6) and can be disregarded.

Even if larger timesteps were compatible with the uniform initializer, there is an additional deficit observed when going to larger timesteps. All previous tests only considered the stability of a simulation but the ultimate requirement is that some converged state is reached. When the timestep is increased between $10^{-5}s$ and $10^{-4}\,\mathrm{s}$, the simulation converges to the same final plasma properties, but oscillations are introdcued in some observed quantities. (Figure 5.7A, C, E). Similarly, increasing the neutral timestep multipliers can increase the amplitude of oscillations (Figure 5.7B, D, F). So while increasing the timesteps might benefit the stability of simulations, the added noise should be avoided and might even prolong the runtimes necessary to achieve convergence.

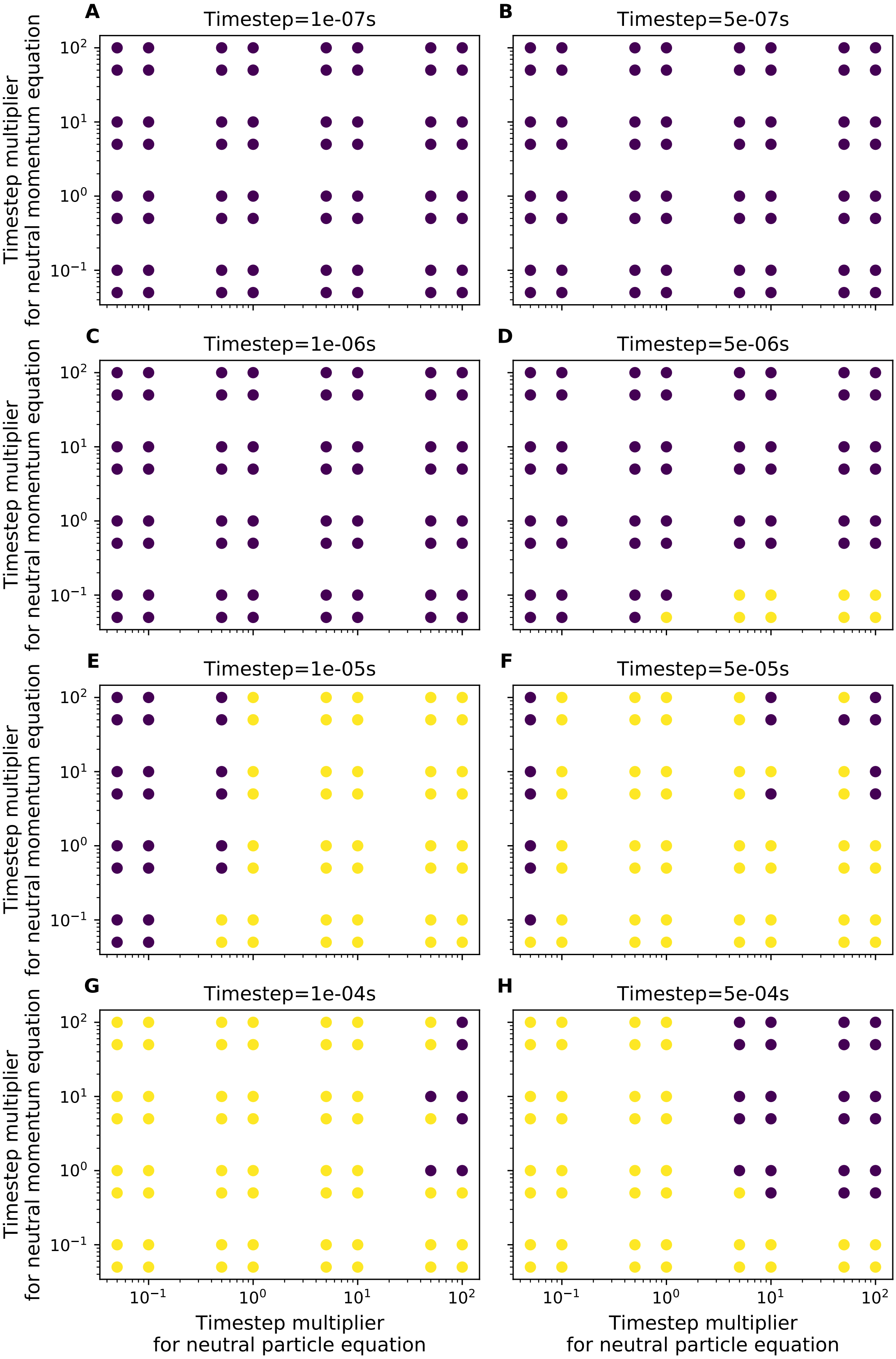


**Figure 5.5:** Overview of the parameter scan with varying neutral timestep multipliers. Each dot represents a simulations with varying timestep and varying multipliers to the timestep for the neutral momentum and particle balance equations. The simulations are the same in all other parameters and are started from the previous initial state (as used Figure 5.1). Dark dots represent diverging simulations, while bright dots show simulations that arrived at the maximum of $10^5$ timesteps.

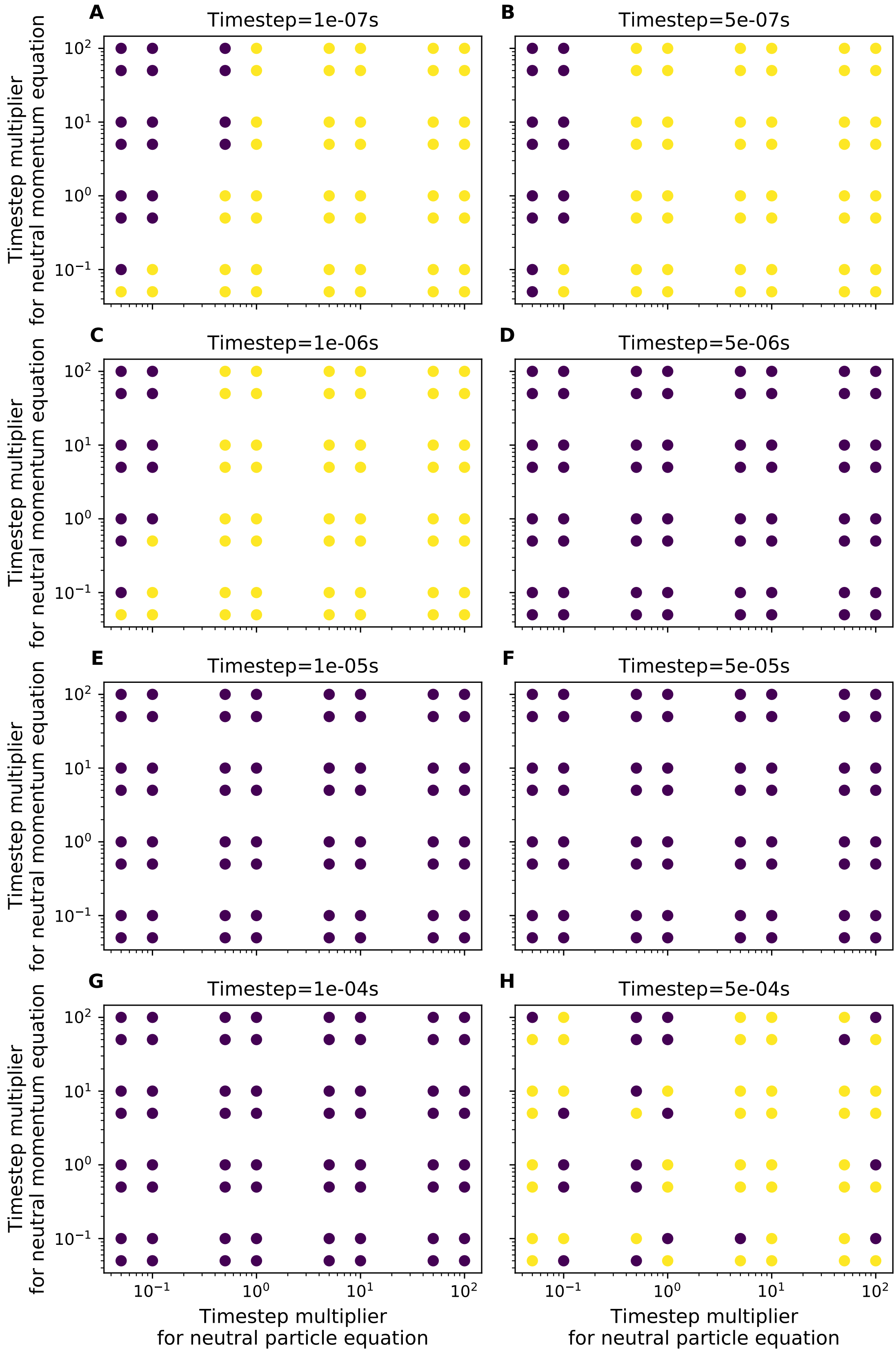


**Figure 5.6:** Overview of the parameter scan with varying neutral timestep multipliers. Each dot represents a simulations with varying timestep and varying multipliers to the timestep for the neutral momentum and particle balance equations. The simulations are the same in all other parameters and are started from the uniform initial state. Dark dots represent diverging simulations, while bright dots show simulations that arrived at the maximum of $10^5$ timesteps.

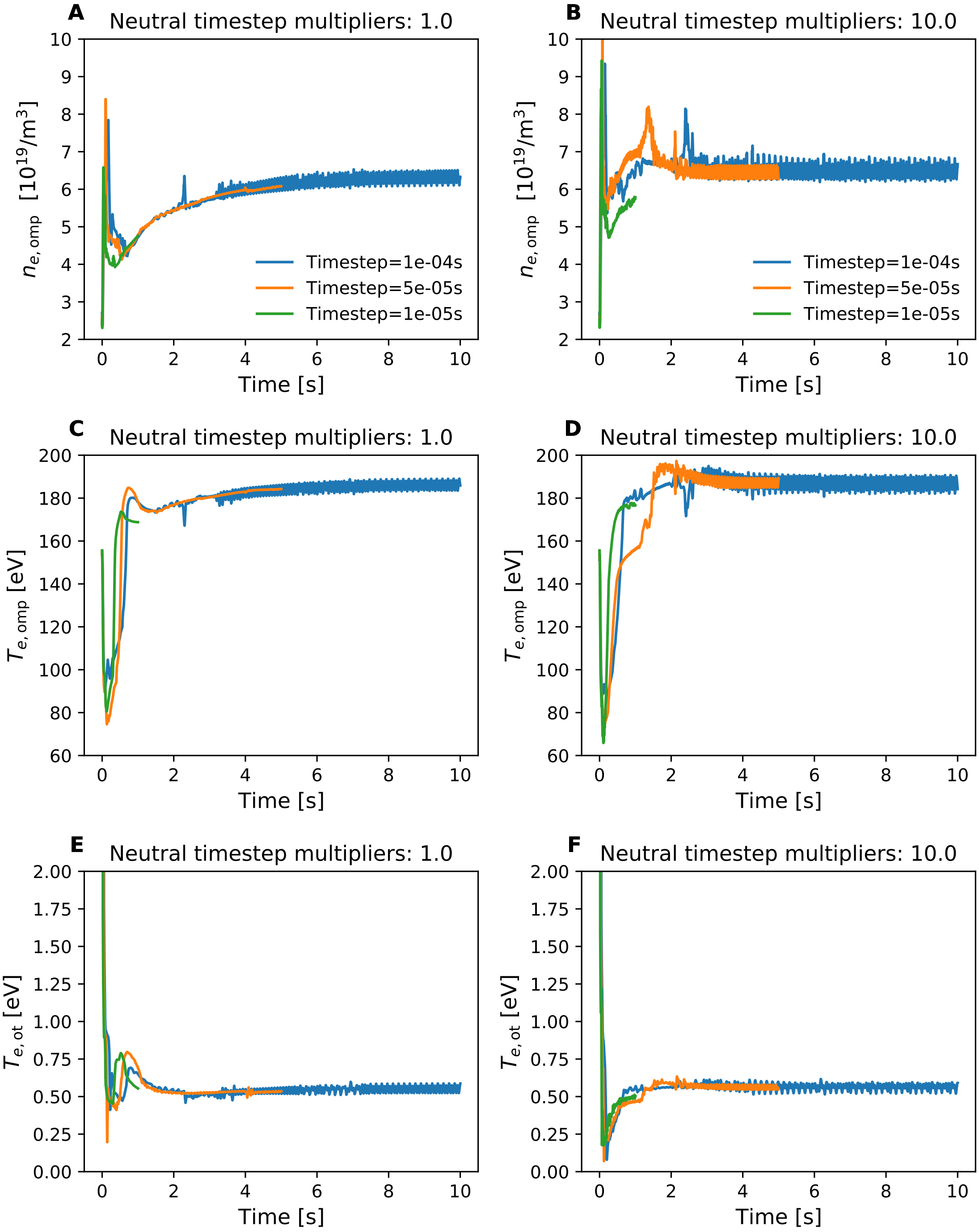


**Figure 5.7:** Timetraces of selected simulations from the parameter scan in Figure 5.5. The simulations have either a multiplier of 1 (A, C, E) or 10 (B, D, F) for the timestep in the neutral momentum and particle equation. All simulations are run for $10^5$ timesteps. Due to the different timestep sizes the simulations are run for different simulated time periods. The timesteps are $10^{-4}$ s (blue), $5 \cdot 10^{-5}$ s (orange) and $10^{-5}$ s (green). The electron density $n_{e,\mathrm{omp}}$ (A, B) and temperature $T_{e,\mathrm{omp}}$ (C, D) at the outer midplane and the electron temperature at the outer target $T_{e,\mathrm{ot}}$ (E, F) are all measured directly outside the separatrix.

## 5.4 ITER testcase

While starting from an initial state with spatially uniform temperatures and densities leads to some improvements, still almost all simulations for larger sizes and high deuterium gas puff diverge (Figure 5.4A). Since none of the tests conducted so far showed any significant improvement, the question remains whether this behaviour can be tolerated or not. While the overall success rate of simulations is important for the efficiency of the data generation process such that the most amount of results is obtained for the number of started simulations, this is not the ultimate criterion. More important is that all relevant physical regimes (Section 2.3) can be simulated, so all are present in the simulation database. If this were not the case, the predictive capabilities of the developed surrogate model would be strongly restricted. To determine whether the present simulation setup is suitable, it remains to be checked if all major SOL regimes are obtainable. It can be difficult to distinguish different regimes in a distributed data set, when all tokamak parameters are randomly varied, especially since it needs to be verified that the regimes are obtainable for physically sensible scenarios and not only in rare extreme cases. Therefore to perform this test, we fix most parameters to a specific scenario and only vary the density controlling parameters because the change in plasma density is one of the main drivers of regime transitions (Section 2.3).

Because the problems of excessive simulation failure occurs only for large sizes above major radius 5 m, an ITER-like scenario is chosen here because it falls in this domain. The chosen major radius is $R = 6.2\,\mathrm{m}$, the toroidal magnetic field $B = 5.3\,\mathrm{T}$ (defined on the magnetic axis), the transport coefficients take the values of the usual assumptions for ITER SOL simulations $D_\perp = 0.3\,\mathrm{m}^2/\mathrm{s}$, $\chi_\perp = 1.0\,\mathrm{m}^2/\mathrm{s}$ [58]. The boundary condition at the core boundary is changed from a fixed temperature condition to a fixed input power flowing as heat flux into the domain (see Section 2.7). Here two scenarios for the input power into the domain are chosen: $P_\mathrm{in} = 100\,\mathrm{MW}$ and $P_\mathrm{in} = 200\,\mathrm{MW}$. In both cases $P_\mathrm{in}$ is equally distributed among the electron and ion heat balance equation. Here $P_\mathrm{in}$ refers to the power entering the domain modeled by SOLPS-ITER. It includes also the expected power generated by fusion in the core plasma. Only the deuterium ion inflow at the core and the deuterium and nitrogen gas puff are varied as before. Although the upper limit for the deuterium gas puff was reduced to $10^{24}$ atoms/s to yield a higher density of points in the tested section of the parameter space. All simulations are started for 1 s of simulated time.

Figure 5.8 depicts the success rates of this ITER testcase depending on the value of the three varied parameters. As expected the overall success rate of the simulations is fairly low and almost vanishes for high deuterium ion influx and high deuterium gas puff. This behaviour is similar in both the 100 MW and 200 MW scenario. The 100 MW case has a generally higher success rate than the 200 MW case for almost all parameter values. The 200 MW case fails almost entirely for nitrogen gas puffs below $5 \cdot 10^{20}$ atoms/s, while the 100 MW case has no strong dependence on the nitrogen gas puff.

To analyse which regimes are obtained, only the successful simulations can be used because crashing simulations diverge to unreasonable values before being stopped

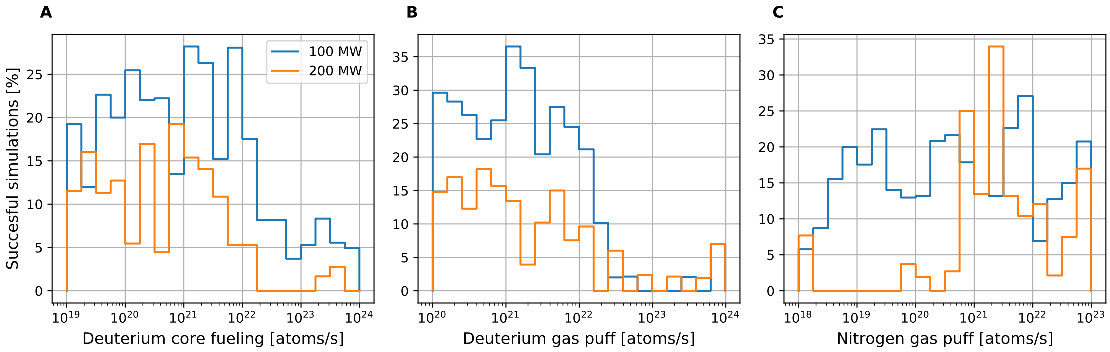


**Figure 5.8:** Histograms of the success rate of simulations for ITER-like scenarios with varying parameter values. The two parameter scans $P_{\text{in}} = 100\,\text{MW}$ (blue) and $P_{\text{in}} = 200\,\text{MW}$ (orange) use identical sets of random parameter values for $D_{\text{core}}$, $D_{\text{puff}}$, $N_{\text{puff}}$ and are identical in all other parameters. The success rate is calculated as in Figure 5.1.

and therefore yield no interpretable results at all. Therefore, it is necessary to anticipate which regimes the diverging simulations would obtain if they were successful. Given that the simulations fail for large values of deuterium ion inflow and gas puff (Figure 5.8A, B) it is clear that the diverging simulations should achieve generally higher plasma densities that the successful ones. For reliable results also the successful simulations need to be performed until some form of convergence to a stationary state is reached. This is not necessarily achieved here because the simulations are only run for one second of simulated time. So the results of the successful simulations might still change or the simulations might still diverge at a later stage. But inferring from the timetraces in Figure 5.7, one can assume that the values are already representative of the overall density levels that are achieved. Figure 5.9A, B, C show the relations between the upstream electron density and the three varied input parameters. As expected all three of them show a positive correlation with the upstream density. The most determining factor, however, seems to be the nitrogen gas puff (Figure 5.9C). This in itself is concerning as the plasma density should be primarily driven by the deuterium fueling and nitrogen is only a comparatively small impurity (see e.g. [187], [188]). Especially since simulations with large variations in the impurity concentration should be possible. Another concerning factor is the almost total lack of simulations with upstream densities between $10^{19}\,\text{m}^{-3}$ and $10^{20}\,\text{m}^{-3}$, which is the usual operating range of modern tokamaks and where regimes transitions should occur. While there are a few simulations in that range, the total number is low compared to the overall number of simulations. All of these higher density simulations are achieved only for high deuterium input (bottom right in Figure 5.9E), which is the area of the parameter space that has the highest failure rates (Figure 5.8A, B).

Figure 5.9D compares the temperature at the outer target against the temperature at the outer midplane (in the first fluxtube outside the separatrix). In the majority of successful simulations the temperatures at the outer target are almost identical to the outer midplane, which characterizes a sheath-limited regime (Section 2.3). The simulations with high nitrogen gas puff (green points in Figure 5.9D) show a strong cooling of the whole SOL with temperatures at the outer midplane of less than 10 eV. The most interesting regimes with high temperature at the outer

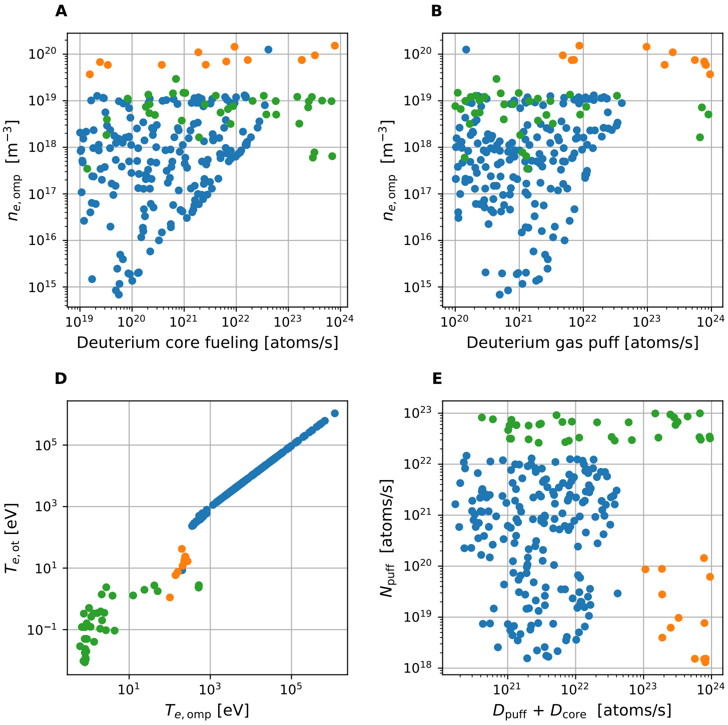


**Figure 5.9:** Results (after $10^5$ timesteps) of all non diverging simulations from the parameter scans in Figure 5.8. The color highlights simulations based on their combination of $D_{\mathrm{core}}$, $D_{\mathrm{puff}}$, and $N_{\mathrm{puff}}$ as seen in E. The electron density $n_{e,\mathrm{omp}}$ and temperature $T_{e,\mathrm{omp}}$ at the outer midplane and the electron temperature at the outer target $T_{e,\mathrm{ot}}$ are all measured directly outside the separatrix.

midplane and a low temperature at the outer target are seen predominantly only in the very few cases that achieved high plasma densities (orange points in Figure 5.9).

Overall, one can conclude that the most relevant physical regimes with high upstream and low target temperatures occur precisely in the density range that is expected. However, with the current simulation setup parameter configurations that achieve this range diverge predominantly. The fraction of successful simulations in this part of the parameter space is so small (Figure 5.8A, B) that even a higher sampling rate will yield only very few of these scenarios. The current simulation setup is therefore unsuitable for the dataset generation and changes need to be made.

## 5.5 New boundary conditions

The approach followed in the previous sections was to introduce as few changes in the simulation settings as possible and incorporate those changes necessary to use the fluid neutral model or to increase the simulation stability. The aim of this approach was to facilitate a one-to-one correspondence between kinetic and fluid neutral simulations by consciously selecting and limiting the changes in the simulation setup such that at a later stage simulations could be repeated with kinetic neutrals turned on. While the original fluid neutral case does run, the setup apparently becomes unstable in the more complex SOL regimes for larger tokamak sizes, which makes the setup unsuitable. Because the tests in the previous sections provided no or only little improvement of the situation, it is apparent that major changes to the simulation setup are required. Instead of testing individual numerical parameters one after another, we therefore change to a totally different set of numerical settings. The only existing study that performed a remotely similar parameter scan using fluid neutrals in SOLPS-ITER was conducted in [141]. Although this study was only conducted with ITER specific parameters, included no size scaling and was focused on comparing the effects of different impurity seeding gases, it is the most promising choice for a numerical setup for fluid neutrals. The publication [141] in itself does not provide a full report on the numerical settings used, but thankfully we were granted access to the archive of the simulations to copy the numerical setup. Table 5.1 provides an overview of the settings that were copied and implemented in the ITER-like testcase from Section 5.4.

Using these settings, the same parameterscan as in Section 5.4 was conducted. Figure 5.10 shows the rate of successful simulations depending on the three varied simulation parameters. It is obvious that the overall success rate both in the 100 MW as in the 200 MW case have improved drastically with an average success rate of 62% compared to the previous 24% in Section 5.4. Also the correlations between the success rate and the input parameters have changed compared to Figure 5.8. For both the 100 MW as well as the 200 MW case, the success rate increases with increasing deuterium influx (Figure 5.10A, B). The deuterium gas puff has the strongest influence on the success rate. Using the old settings all simulations failed above a certain deuterium gas puff value (Figure 5.8B), but now the trend is opposite and the number of successful simulations even increases with increasing deuterium gas puff (Figure 5.10B).

Figure 5.11 shows the physical results after one second of simulated time (so not necessarily converged) similar to Figure 5.9. While in the previous setup there were hardly any simulation with an upstream electron density above $10^{19}\,\mathrm{m}^{-3}$ (Figure 5.9A), now the majority of simulations has upstream densities in the physically expected range between $10^{19}\,\mathrm{m}^{-3}$ and $10^{20}\,\mathrm{m}^{-3}$ (Figure 5.10A, B, C). Also there are no more large gaps but densities are realized in a continuous range from very low densities to more than $10^{20}\,\mathrm{m}^{-3}$ (Figure 5.10A, B, C). Figure 5.11D again compares outer target to outer midplane temperatures. Similar as before regimes are obtained in which no heat mitigation takes place and target temperatures are identical to the upstream temperatures and other regimes where the whole SOL is cooled down But the major difference is the existence of a continuous spectrum of simulations with

| | | Old | New |
|---|---|---|---|
| Core | $n_{ions}$ | Constant<br>$\mathit{\Gamma}_{D^+} = D_{\text{core}}/A_{\text{core}}$<br>$\mathit{\Gamma}_{N^{x+}} = 0$ | Feedback<br>$\mathit{\Gamma}_{D^+} = D_{\text{core}}/A_{\text{core}}$<br>$\mathit{\Gamma}_{N^{x+}} = 0$<br>$\alpha = 0.001$ |
| | $n_{neutrals}$ | Constant<br>$\mathit{\Gamma}_D = \mathit{\Gamma}_N = 0$ | Leak<br>$\alpha = 0.1$ |
| | $u_{\|\|ions}$ | $\nabla u_{\|\|} = 0$ | Weak<br>$u_{\|\|} = 0$<br>$h = 10^6$ |
| | $u_{\|\|neutrals}$ | $\nabla u_{\|\|} = 0$ | Weak<br>$u_{\|\|} = 0$<br>$h = 10^6$ |
| Target | $T_e$ | Sheath<br>$\gamma_e = 1.0$ | Sheath<br>$\gamma_e = 0.9$ |
| | $T_i$ | Sheath<br>$\gamma_i = 1.5$ | Sheath<br>$\gamma_i = 2.5$ |
| PFR | $n_{ions}$ | Decay<br>$\lambda = 0.03\,m$ | Leak<br>$\alpha = 0.001$ |
| | $n_{neutrals}$ | Leak<br>$\alpha = 0.1$ | Leak<br>$\alpha = 0.5$ |
| | $u_{\|\|ions}$ | $\nabla u_{\|\|} = 0$ | Weak<br>$u_{\|\|} = 0$<br>$h = 10^6$ |
| | $u_{\|\|neutrals}$ | $\nabla u_{\|\|} = 0$ | Weak<br>$u_{\|\|} = 0$<br>$h = 10^6$ |
| far-SOL | $n_{ions}$ | Decay<br>$\lambda = 0.03\,m$ | Leak<br>$\alpha = 0.001$ |
| | $u_{\|\|ions}$ | $\nabla u_{\|\|} = 0$ | Weak<br>$u_{\|\|} = 0$<br>$h = 10^6$ |
| | $u_{\|\|neutrals}$ | $\nabla u_{\|\|} = 0$ | Weak<br>$u_{\|\|} = 0$<br>$h = 10^6$ |
| | $T_e$ | Decay<br>$\lambda = 0.03\,m$ | Decay<br>$\lambda = 0.01\,m$ |
| | $T_i$ | Decay<br>$\lambda = 0.03\,m$ | Decay<br>$\lambda = 0.01\,m$ |

Additional changes to the numerics are:
Core timestep is multiplied by factor 1000 (previously 1)
Adiabatic coefficient $\gamma$ is set to 5/3 (previously 1)
Viscosity of the ion fluids is set to 0 (previously 0.3)

**Table 5.1:** Summary of all the changes made to the boundary conditions and numerical parameters between the old settings used in Section 5.4 and the new settings introduced now. A full list of all boundary conditions used is given in Table 6.1. The definitions of the relevant conditions can be found in Table 2.1. $A_{\text{core}}$ is the size of the boundary surface at the core boundary. $D_{\text{core}}$ is the deuterium fueling rate varied in the parameter scan. The new boundary conditions are taken from the simulations in [141].

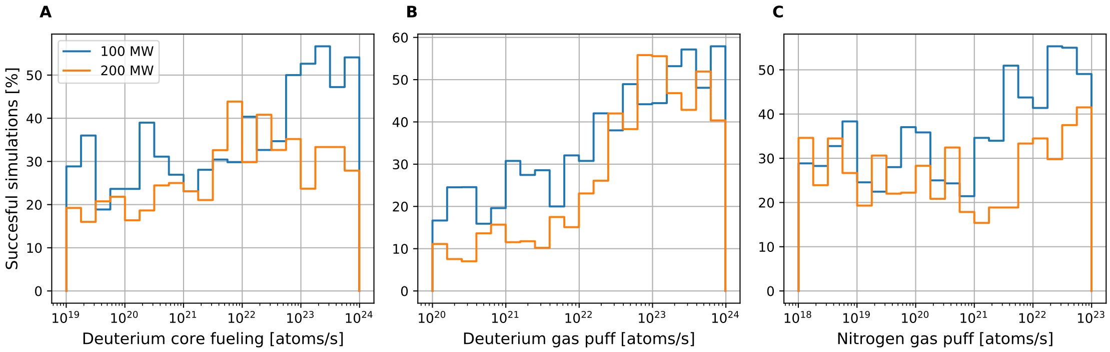


**Figure 5.10:** Same display as in Figure 5.8 but with data from the ITER-like test simulations conducted with the new boundary conditions. The two parameter scans $P_{\text{in}} = 100\,\text{MW}$ (blue) and $P_{\text{in}} = 200\,\text{MW}$ (orange) use identical sets of random parameter values for $D_{\text{core}}$, $D_{\text{puff}}$, $N_{\text{puff}}$ and are identical in all other parameters.

similar upstream conditions at $200 - 300\,\text{eV}$ but drastically lower target temperatures. The target temperatures can get as low as $1\,\text{eV}$, which shows that different heat mitigation channels are relevant and are strong enough to reach detached conditions (Section 2.3). Compared to Figure 5.9E, Figure 5.11E now does not show a gap due to failed simulations when increasing the nitrogen gas puff but instead also for high deuterium inflow the nitrogen gas puff can be set to anywhere between the minimum and maximum values. The only large area in the parameter space where almost no simulation succeeds is where both deuterium and nitrogen inflow are small (Figure 5.11E). From the densities of the simulations surrounding this area (blue dots), it can be estimated that simulations in this part of the parameter space would have extraordinarily low densities. Since the densities in the successful simulations already go down to $10^{15}\,\text{m}^{-3}$, which is orders of magnitude lower than what is relevant for present and future tokamaks, this behaviour is entirely tolerable.

Figure 5.12 compares the number of successful simulations per upstream electron density obtained between the two different numerical setups. Although these simulations are not yet converged, this new setup shows an enormous improvement in the number of successful simulations in the relevant range of upstream densities between $10^{19}\,\text{m}^{-3}$ and $10^{20}\,\text{m}^{-3}$. Across the whole distribution also for low densities the new setup is not drastically worse, but similarly low densities can be oberserved as before.

Going forward it is clearly beneficial to use this new set of numerical parameters. The following chapter summarizes the final simulation setup with these settings and how it is used to generate a dataset for surrogate model training.

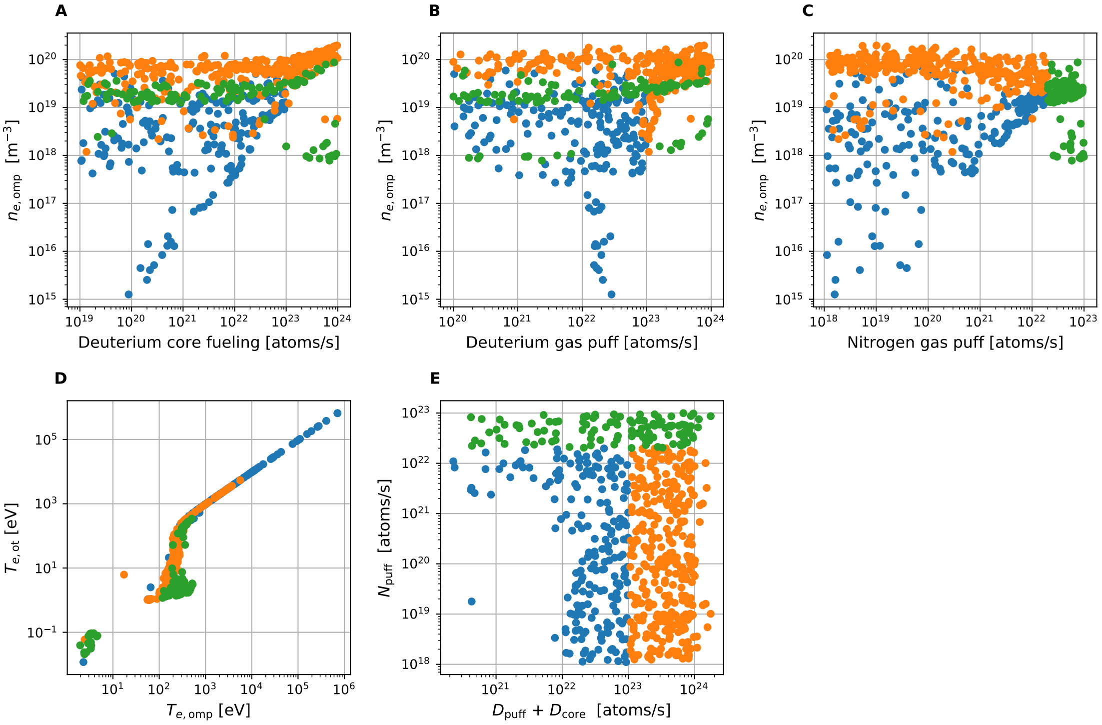


**Figure 5.11:** Results (after $10^5$ timesteps) of all non diverging simulations from the parameter scans in Figure 5.10. The color highlights simulations based on their combination of $D_{\mathrm{core}}$, $D_{\mathrm{puff}}$, and $N_{\mathrm{puff}}$ as seen in E. The electron density $n_{e,\mathrm{omp}}$ and temperature $T_{e,\mathrm{omp}}$ at the outer midplane and the electron temperature at the outer target $T_{e,\mathrm{ot}}$ are all measured directly outside the separatrix.

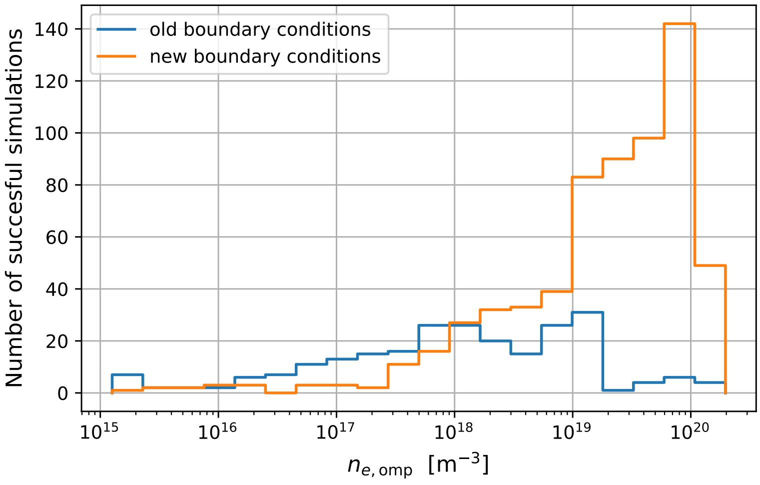


**Figure 5.12:** Histograms of the absolute number of non diverging simulations with given outer midplane separatrix electron density $n_{e,\mathrm{omp}}$ (after $10^5$ timesteps) from the ITER-like test parameters scans with old (blue) and new boundary conditions (orange).

# Chapter 6

# Final simulation for dataset generation

The prelimary tests conducted in the preceeding chapters made various changes to the SOLPS-ITER simulation case. For clarification and as a reference, this chapter shortly summarizes the final simulation setup and the procedures used to create the final dataset of SOLPS-ITER simulations. Every analysis and model development conducted in the following chapters will be based purely on this dataset generated here (except for the additional ITER simulations used in Chapter 10).

All simulations are conducted on the same numerical grid from the reference JET simulation characterized in Section 4.2 and with the fluid neutral model introduced in Section 4.3. The boundary conditions are thus defined on the domain boundaries as given in Figure 4.2. The settings used for all simulations in the database are entirely identical except for eight varied parameters. The varied parameters are the tokamak major radius $R$, toroidal magnetic field strength $B$, deuterium gas puff rate $D_{\mathrm{puff}}$, nitrogen gas puff rate $N_{\mathrm{puff}}$, deuterium core fueling rate $D_{\mathrm{core}}$, input power $P_{\mathrm{in}}$, and the cross-field transport coefficients for the ion densities $D_{\perp}$ and the anomalous thermal diffusivity $\chi_{\perp}$. As only these parameters are varied, the analysis in the following chapters will refer to these eight parameters as *the simulation parameters.* The tokamak major radius $R$ and magnetic field strength are defined at the magnetic axis. Changes to the tokamak major radius also change the size of the tokamak cross-section as described in Section 4.1. In practice this mean that proportional to the changes in the major radius also the horizontal and vertical coordinates of each grid cell in the computational domain are changed. If the major radius is increased by a factor $x$ compared to the size in the reference JET simulation, the spatial coordinates of each grid cell will be multiplied by the same factor $x$. So changing $R$ not only affects the connection length between the upstream plasma and the divertor but also the width of the scrape-off layer and private-flux region that are covered by the simulation. Therefore in simulations with small $R$, the far-SOL, PFR and core boundaries of the simulation domain (see Figure 4.2) will be at a shorter distance to the separatrix than in simulations with larger $R$. Similarly, changes to the toroidal magnetic field strength $B$ invoke proportional changes to the poloidal magnetic field in every grid cell. So if $B$ is increased by a factor $x$ compared to value in the reference JET simulation, all components of the magnetic field in each grid cell are multiplied by the same factor $x$. The parameters $D_{\mathrm{puff}}$, $N_{\mathrm{puff}}$, $D_{\mathrm{core}}$, $P_{\mathrm{in}}$ change

| | Core | Target | PFR | far-SOL |
|---|---|---|---|---|
| $n_{ions}$ | Feedback<br>$\mathit{\Gamma}_{D^+} = \frac{D_{\text{core}}}{A_{\text{core}}}$<br>$\mathit{\Gamma}_{N^{x+}} = 0$<br>$\alpha = 0.001$ | Sheath<br>$\mathit{\Gamma}_{D^+} = c_{s,\text{avg}} n_{D^+}$<br>$\mathit{\Gamma}_{N^{x+}} = c_{s,\text{avg}} n_{N^{x+}}$ | Leak<br>$\alpha = 0.001$ | Leak<br>$\alpha = 0.001$ |
| $n_{neutrals}$ | Leak<br>$\alpha = 0.1$ | Recycling<br>$\mathit{\Gamma}_D = -\mathit{\Gamma}_{D^+}$<br>$\mathit{\Gamma}_N = -\sum_{x=1}^{7} \mathit{\Gamma}_{N^{x+}}$ | Leak<br>$\alpha = 0.5$ | Constant<br>$\mathit{\Gamma}_D = \frac{D_{\text{puff}}}{A_{\text{far-SOL}}}$<br>$\mathit{\Gamma}_N = \frac{N_{\text{puff}}}{A_{\text{far-SOL}}}$ |
| $u_{\|\|ions}$ | Weak<br>$u_{\|\|} = 0$<br>$h = 10^6$ | Sheath<br>$u_{\|\|} = c_{s,\text{avg}}$ | Weak<br>$u_{\|\|} = 0$<br>$h = 10^6$ | Weak<br>$u_{\|\|} = 0$<br>$h = 10^6$ |
| $u_{\|\|neutrals}$ | Weak<br>$u_{\|\|} = 0$<br>$h = 10^6$ | Constant<br>$u_{\|\|} = 0$ | Weak<br>$u_{\|\|} = 0$<br>$h = 10^6$ | Weak<br>$u_{\|\|} = 0$<br>$h = 10^6$ |
| $T_e$ | Constant<br>$q_e = \frac{P_{\text{in}}}{2A_{\text{core}}}$ | Sheath<br>$\gamma_e = 0.9$ | Decay<br>$\lambda = 0.01\,m$ | Decay<br>$\lambda = 0.01\,m$ |
| $T_i$ | Constant<br>$q_i = \frac{P_{\text{in}}}{2A_{\text{core}}}$ | Sheath<br>$\gamma_i = 2.5$ | Decay<br>$\lambda = 0.01\,m$ | Decay<br>$\lambda = 0.01\,m$ |

**Table 6.1:** The boundary conditions used to generate the final dataset. The definitions and formula of the different boundary conditions can be found in Section 2.7. The location of the different boundaries are shown in Figure 4.2. $A_{\text{core}}$, $A_{\text{far-SOL}}$ are the sizes of the boundary surfaces at the core and far-SOL simulation boundary. $P_{\text{in}}$, $D_{\text{core}}$, $D_{\text{puff}}$ and $N_{\text{puff}}$ are simulation parameters varied in the parameter scan.

the boundary conditions for the plasma simulations. $D_{\text{puff}}$ and $N_{\text{puff}}$ prescribe an influx of neutral deuterium and nitrogen atoms at the far-SOL boundary into the computational domain. $D_{\text{core}}$ determines the influx of deuterium ions at the core boundary into the computational domain. $P_{\text{in}}$ determines the energy flow into the computational domain at the core boundary. This heat flux is evenly distributed to the electron and ion heat balance equations. For these simulations it is equivalent whether in an experiment this power is generated solely by external heating or as fusion power. All these boundary fluxes are uniformly distributed across the surfaces of the respective boundaries. $D_\perp$and $\chi_\perp$ are scalar parameters used inside the solved equations. The parameters are constant in the whole simulation domain (Section 4.4). The domain boundary in the private-flux region acts as pump for all particles, with fixed pumping rates. The exact boundary conditions used at each of the domain boundaries are given in Table 6.1. The definitions and formula associated with the names of these boundary conditions can be found in Table 2.1 in Section 2.7.

The simulations contain eight plasma fluids, the deuterium ions ($D^+$) and all ionization stages of nitrogen $N^+ ... N^{7+}$ and two neutral fluids, atomic deuterium $D$ and nitrogen $N$. All simulations are started from an initial state with constant densities ($[n_D, n_{D^+}, n_N, n_{N^{x+}}] = [10^{16}, 10^{19}, 10^{12}, 10^{12}]$ m$^{-3}$) and temperatures ($T_e = T_i = 100\,\text{eV}$) uniform in the whole simulation domain.

Using these settings two large sets of simulations are generated. The first is meant to acts as training data for the development of surrogate models and the second as test set to evaluate the accuracy of the final models. In both sets the eight parameters for each simulation are varied inside the limits given in Table 6.2. These parameter ranges are selected because they encompass many presently existing tokamaks (AUG, JET, ITER) and possible future developments (DEMO) [11], [37]. Also compact high-field tokamaks like SPARC [38] might be represented in this dataset, due to the combination of high powers $P_{\text{in}}$ with small sizes. Although the included magnetic field strengths are slightly less than anticipated for these reactors (12.2 T [38]). But the magnetic topology of such high-field tokamaks might be to different to the geometry here to allow for meaningful comparisons. The parameter values for $D_{\text{puff}}$, $N_{\text{puff}}$ and $D_{\text{core}}$ are chosen from a logarithmic scale (base 10) between the minimum and maximum values while all other parameters are uniformly distributed. The ranges of the gas puff values are deliberately chosen extra large to ensure that all regimes can be obtained also with the less known fueling behaviour in fluid neutral simulations (see Section 4.3). For simulations in the training set the parameter combinations are not chosen randomly inside the defined limits but according to an eight dimensional Sobol sequence [189]. This low-discrepancy sequence covers the parameter space more efficiently compared to random numbers or other such sequences [190]. The sequence has the additional benefit that specific smaller subsets of simulations can be taken from the training set which still cover the whole parameter space evenly. This feature is used in Section 8.4. For the training set 8192 simulations are started. For the test set 2048 simulations are started but with parameter values distributed pseudo-randomly in the parameter space according to a uniform distribution.

The simulations are run for intervals of 1 s simulated time with a timestep of $10^{-5}$ s. At the end of each 1 s period it is checked whether the simulation has either converged to a steady state, shows stable oscillations or has been run for a total of 10 s. If either of the three conditions is met, the simulation is stopped. Simulations are defined as steady state when the ratio between the maximum and the minimum outer midplane electron separatrix density $n_{e,\text{omp}}$ in the last 0.1 s of the 1 s period is less than 0.1%. A simulation counts as oscillating if the maximum and minimum values of the outer midplane separatrix density $n_{e,\text{omp}}$ in the first 0.1 s of a 1 s period are close to the values in the last 0.1 s of the same period. This closeness is reached if the values (maximum and minimum) from the start and end of the period differ by less than 0.5%. These convergence metrics are adapted from [141] and are less strict than most commonly used. These metrics only determine whether some form of stationarity of the plasma state is reached and do not consider the residuals from the implicit equation solver. Strictly speaking it is not guaranteed whether the simulation results generated by this procedure are actual true solutions to the underlying system of equations. This relaxed approach is, however, necessary because minimizing the residuals in all simulations would require enormous compute times, probably necessity additional fine tuning of numerical parameters and might not even be impossible in some cases due to the simple fluid neutral model that is used. But for creating a surrogate model it is also not necessary to find a true solution in every case as long as in most of the simulations the found solutions are close to the actual result. Therefore, one focus of the analysis in the following chapter

| | $R$ | $B$ | $D_{\text{puff}}$ | $N_{\text{puff}}$ | $D_{\text{core}}$ | $P_{\text{in}}$ | $D_\perp$ | $\chi_\perp$ |
|---|---|---|---|---|---|---|---|---|
| min | 1 | 1 | $10^{20}$ | $10^{18}$ | $10^{19}$ | 10 | 0.1 | 0.1 |
| max | 10 | 10 | $10^{24}$ | $10^{23}$ | $10^{24}$ | 200 | 2 | 2 |
| units | m | T | atoms/s | atoms/s | atoms/s | MW | $m^2/s$ | $m^2/s$ |
| scale | lin | lin | log | log | log | lin | lin | lin |

**Table 6.2:** Overview of the parameters varied in the final dataset. The datapoints are distributed either uniformly over a linear scale (lin) or uniformly over a logarithmic scale (log) between minimum and maximum values.

is to determine whether the results seen in the database are reasonable, follow the expected trends and are free from overly strong influences invoked by the simulation procedure.

# Chapter 7

# Analysis of the simulation dataset

Before the dataset created in Section 6 is used to train surrogate models, it is paramount to first analyse the data. Just like in any machine learning task, a better understanding of the dataset itself can help to train better models. However, here another aspect is more important. Because the surrogate model is trained only on the simulation data and has no experimental data, the validity of the simulation data restricts the validity of the trained surrogate model. Therefore, it is vital to check whether the procedure described in Section 6 provides a useful dataset, which restrictions apply, which physical phenomena are seen and what kind of correlations exist. The reduced fluid neutral model and the simplified convergence metrics which are used, might reduce the validity of the results. Only by analysis of the data, it can be identified whether some quirks are introduced by the surrogate model itself or already present in the underlying simulations. As a third motivation for the analysis of the dataset, the dataset in itself might provide valueable physical insights.

Section 7.1 analyses the correlation between the varied simulation input parameters and the statistics of the stopping conditions by which the simulations are halted. Section 7.2 looks into the oscillations in the simulations and how they need to be treated in the surrogate model. Section 7.3 describes investigates the influence the selection of the maximum simulation runtime has on the results and what choices are suitable for efficient data generation. Section 7.4 checks which physical regimes are present in the dataset and finally Section 7.5 looks into the statistics and origin of a temperature rise in the PFR seen in some simulations.

## 7.1 Statistics of the stopping conditions

The efforts in the previous sections are centered around finding a simulation setup that yields successful simulations across the whole parameter space. The first step is, therefore, to confirm whether this is the case and whether there are some systematics behind the occurances of the different convergence metrics. The analysis in this section is conducted solely on the simulations performed as training data, but the statistics in the test data should be identical.

Of the 8192 started training simulations, 2436 diverged, while the other 70.2% reached one of the three stopping conditions. Of these 5756 successful simulations, 2794 achieved the steady state condition, 2514 were stopped because of stable oscillations and 448 reached the maximum runtime of 10 s (in the later called max. time).

Figure 7.1 depicts how the occurrence of these different stopping conditions depends on the simulation parameters. It is clear that the new boundary conditions (Section 5.5) drastically improved the success rate of the simulations, as compared to Figure 5.1. Especially for none of the eight varied simulation parameters, a range exists where all simulations diverge. For almost all parameters and all ranges, the number of successful simulations (counting all three stopping conditions together) is higher than the number of diverging simulations. The number of started simulations is roughly uniform across all ranges of all parameters. Changes in the absolute number of diverging simulations directly correspond to the failure rate. If a parameter has no influence on which stopping condition is reached, their rate should be uniform across the whole parameter range. This is the case for the toroidal magnetic field strength (Figure 7.1B). The strongest influences on the failure rate stem from the major radius (Figure 7.1A), the input power (Figure 7.1C), the deuterium ion influx (Figure 7.1D) and the heat transport coefficient (Figure 7.1H). The higher failure rates for small tokamaks and large input powers seem plausible especially when these occur in combination. In these situations the input powers might be too large to be dissipated in the smaller tokamak volumes (volume scales with $R^3$), which results in diverging temperatures. Also the higher failure rate at low $\chi_\perp$ can be an issue in simulations because then any cross-field transport can only occur when exceedingly large gradients are present. That an increase in the simulation failure rate is only seen for small $\chi_\perp$ and not $D_\perp$, could partly stem from the tested ranges. The heat transport coefficient usually has higher values than the density transport coefficient. E.g. for ITER the default assumption is $D_\perp = 0.3\,\mathrm{m^2/s}$ and $\chi_\perp = 1.0\,\mathrm{m^2/s}$. The minimum values tested here $0.1\,\mathrm{m^2/s}$ constitute a factor 10 reduction of heat transport and only a factor 3 reduction of particle transport Testing even smaller values might also show a similar increase in the failure rate when similarly strong relative decreases of $D_\perp$ are reached. However, the increase in failure rate for $\chi_\perp$ starts already at high values around $1\,\mathrm{m^2/s}$ (Figure 7.1H), while the failure rate shows almost no dependence on $D_\perp$ (Figure 7.1G). Therefore, it is more likely that the heat cross-field transport has a generally higher influence on simulation divergence. The simulation failure rate also decreases with increasing deuterium ion influx (Figure 7.1D) and deuterium gas puff (Figure 7.1E). For the ion input this effect persists across the whole parameter range, while for deuterium gas puffs below $10^{22}$ atoms/s the failure rate is almost constant. The rate of simulations reaching the maximum simulation time is almost constant across all parameters except for the deuterium and nitrogen gas puffs. For deuterium gas puffs above $10^{23}$ atoms/s, the number of these simulations is slightly increased (Figure 7.1E). For nitrogen gas puffs above $10^{21}$ atoms/s, the number of max. time simulations increases continuously with rising gas puff and shows a spike at the highest gas puff values (Figure 7.1F). Concerning the number of simulations converging to a steady state or oscillations, two types of patterns are seen. If the parameter dependence of both is similar and inverse to the failure rate, then the total number of converging simulations is simply increased because less diverge. This is the case for the input power (Figure 7.1C) and the heat transport coefficient (Figure 7.1H). In both histograms the simulations also split almost evenly into steady state and oscillatory cases, with a slightly higher number of steady state cases. In other cases (Figure 7.1A, D, E, F) the occurance of steady state and oscillatory simulations are inverse to one another. $D_{\text{core}}$, $D_{\text{puff}}$ and $N_{\text{puff}}$ all have in common that for high values of these parameters the number of simulations reaching steady state decreases significantly while the number of simulations showing oscillations increases. For $N_{\text{puff}}$ this pattern is almost across the whole parameter range (Figure 7.1F) while for $D_{\text{core}}$ and $D_{\text{puff}}$ this pattern only starts around $10^{23}$ atoms/s (Figure 7.1D, E). Interestingly, the number of steady state simulations also decreases for ion influx below $10^{20}$ atoms/s, which is correlated with the increasing failure rate. But in that range the increasing failure rate reduces only the steady state simulations, and the number of simulations in both other groups remains stable (Figure 7.1D).

Figure 7.2 depicts the dependence of the obtained stopping conditions based on pairwise combinations of simulation parameters. This confirms the expectation that a higher diverging rate of simulations occurs for a combination of small tokamak size and large input power (Figure 7.2B) although the trend is only small. More pronounced is the increase of the failure rate for high input power and low $D_{\text{puff}}$ (Figure 7.2F) and for a combination of low $D_{\text{puff}}$ and $D_{\text{core}}$ (Figure 7.2N). Otherwise Figure 7.2 shows predominantly the single parameter influences already seen in Figure 7.1 such as the increase of simulations ending in stable oscillations for high deuterium gas puff values (Figure 7.2D, E, F).

In conclusion: The success rate of the simulations in the database is much higher than in the previous parameter scans in Sections 5.1-5.4. Additionally there is no part in the input parameter space that leads exclusively to divergence or only a singular convergence metric appearing. Nevertheless, the probability for a simulation to diverge or to meet a specific convergence metric depends on the input parameters of the simulations. Some of the correlations, which are found, seem physically plausible. Also no hints were found that one of the convergence metrics is solely associated with unrealistic scenarios.

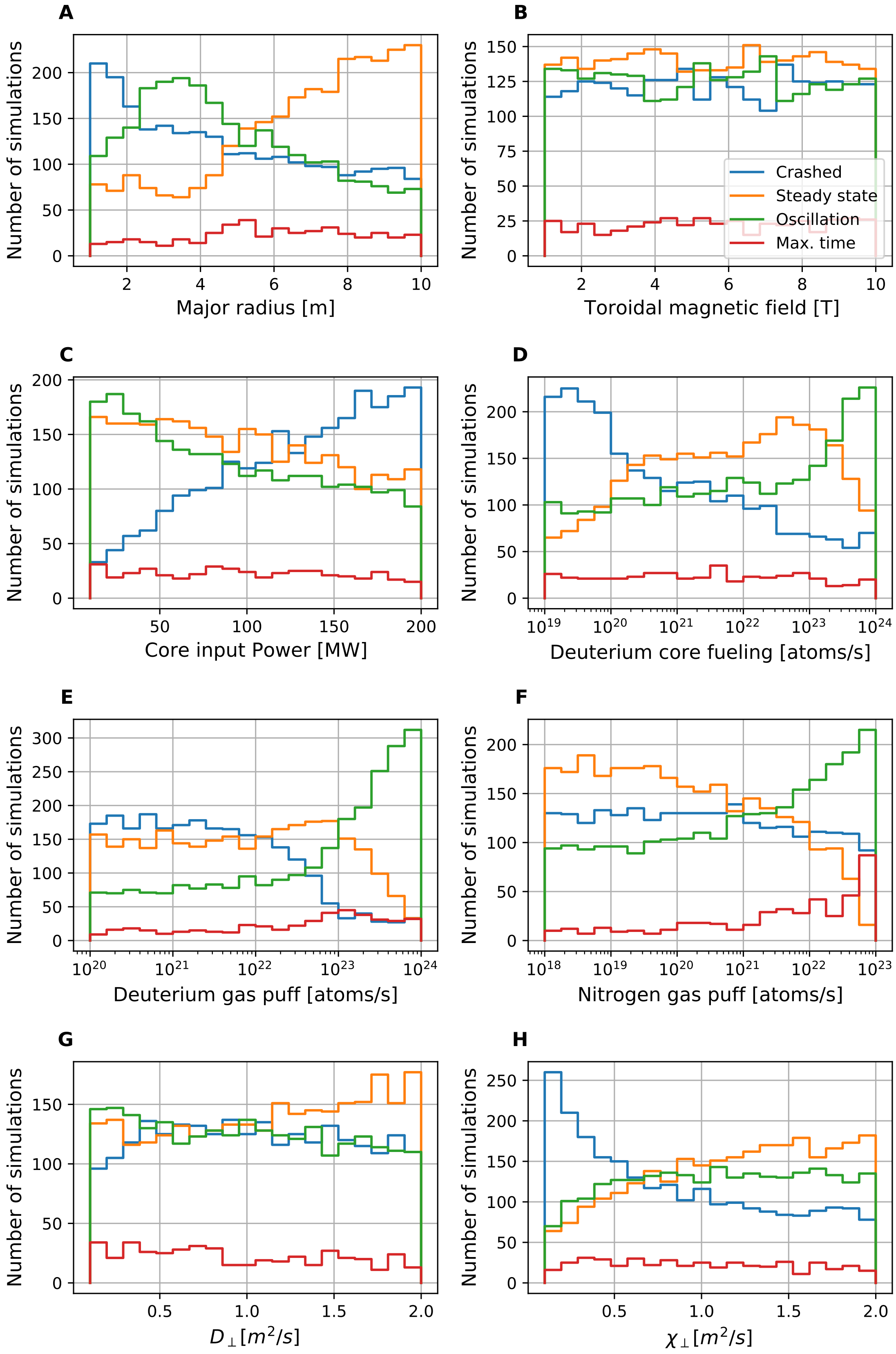


**Figure 7.1:** Histograms of the number of simulations over varying parameter values in the training set, separated into the four possible stopping conditions. These are the simulations that crashed due to divergence (blue), simulations reaching a steady state (orange), showing stable oscillations (green) or that reached the maximum of 10 s simulated time (red).

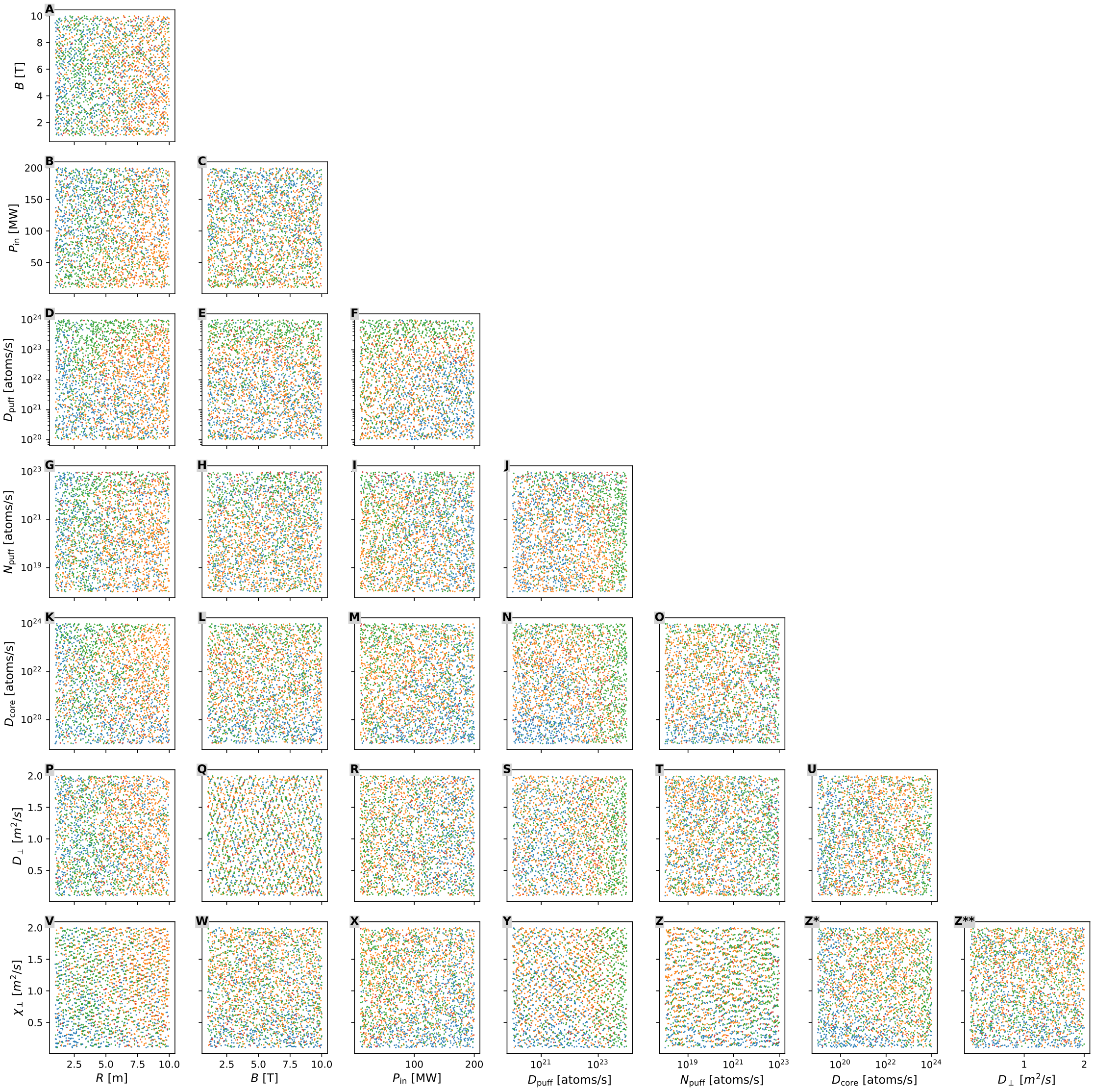


**Figure 7.2:** Each dot represents a started simulation from the training data set, which is color coded by the condition that stopped the simulation. These are the simulations that crashed due to divergence (blue), simulations reaching a steady state (orange), showing stable oscillations (green) or that reached the maximum of 10 s simulated time (red). For visibility only 30% randomly selected training simulations are shown.

## 7.2 Oscillations

The previous section has shown that a significant part of the simulations is stopped because they show oscillations or run for 10s without reaching convergence. This section should provide a basic overview of how these runs look and how they should be used in the process of developing a surrogate model.

Figure 7.3 shows the timetraces of the upstream electron density in the last 1 s before the simulations were stopped by one of the three stopping conditions. The examples are chosen to provide an overview of the different situations that are commonly appearing. It is apparent that the used convergence metrics provide not a 100% clear separation of the different situations. Some simulations reaching steady state still show oscillations, but with a small enough amplitude that these still fall inside the steady state metric (as in Figure 7.3D). Or because the metric to detect oscillations only compares the start and end of 1 s intervals, misclassifications can appear, where oscillations are not detected when the frequency of oscillations is low ($\approx 1Hz$) as is the case in Figure 7.3L and Figure 7.3C. For data storage efficiency only the 1 s time traces are stored. For cases like Figure 7.3H it is therefore not possible to analyse whether the classification was correct in finding a very slow oscillation or whether actually some transient behaviour is ongoing. The examples in Figure 7.3B, E, H, K demonstrate drastically differing oscillation frequencies and some cases whow multiple intertwined oscillations. Some oscillations, as in Figure 7.3E, K, contain small transients. The oscillations in some simulations are clear periodic patterns (Figure 7.3B), but other simulations show only noisy indistinguishable patterns (Figure 7.3E). The simulations stopped after 10 s runtime are even more diverse (Figure 7.3C, F, I, L). The cases in Figure 7.3F, I might still reach a steady state or stable oscillation with longer runtimes. Figure 7.3L shows a very slow oscillation and in Figure 7.3C it is unclear whether it is a slow oscillation or transient behaviour. The example in Figure 7.3C hints at another type of possible misclassifications, which is not seen in these examples. For very slow oscillations, it is possible that the phase of constant upstream density coincides with the last 0.1 s of the simulation run, which would be classified as steady state convergence.

Overall it is clear that the choice in the classification metrics is always a trade-off between catching all cases and misclassifications. Here the detection of oscillations was performed purely based on the electron density at the outer midplane separatrix as in [141], [191]. This is chosen because the upstream density is a significant driver for regime transitions (see Section 2.3). Other metrics might consider different quantities and locations or macroscopic measurements such as the total particle content in the simulation domain.

With the settings used in SOLPS-ITER in this thesis, the time traces are not necessarily some physically valid time-dependent behaviour but the goal of the simulations is to obtain steady-state plasma properties corresponding to the given input variables. Regarding the simulations, which do not reach a steady state, the most critical question is: Which values should be taken as the final results of these simulations and what kind of errors or uncertainties are introduced because of these dynamics. The most critical error would be if simulations are stuck in local minima

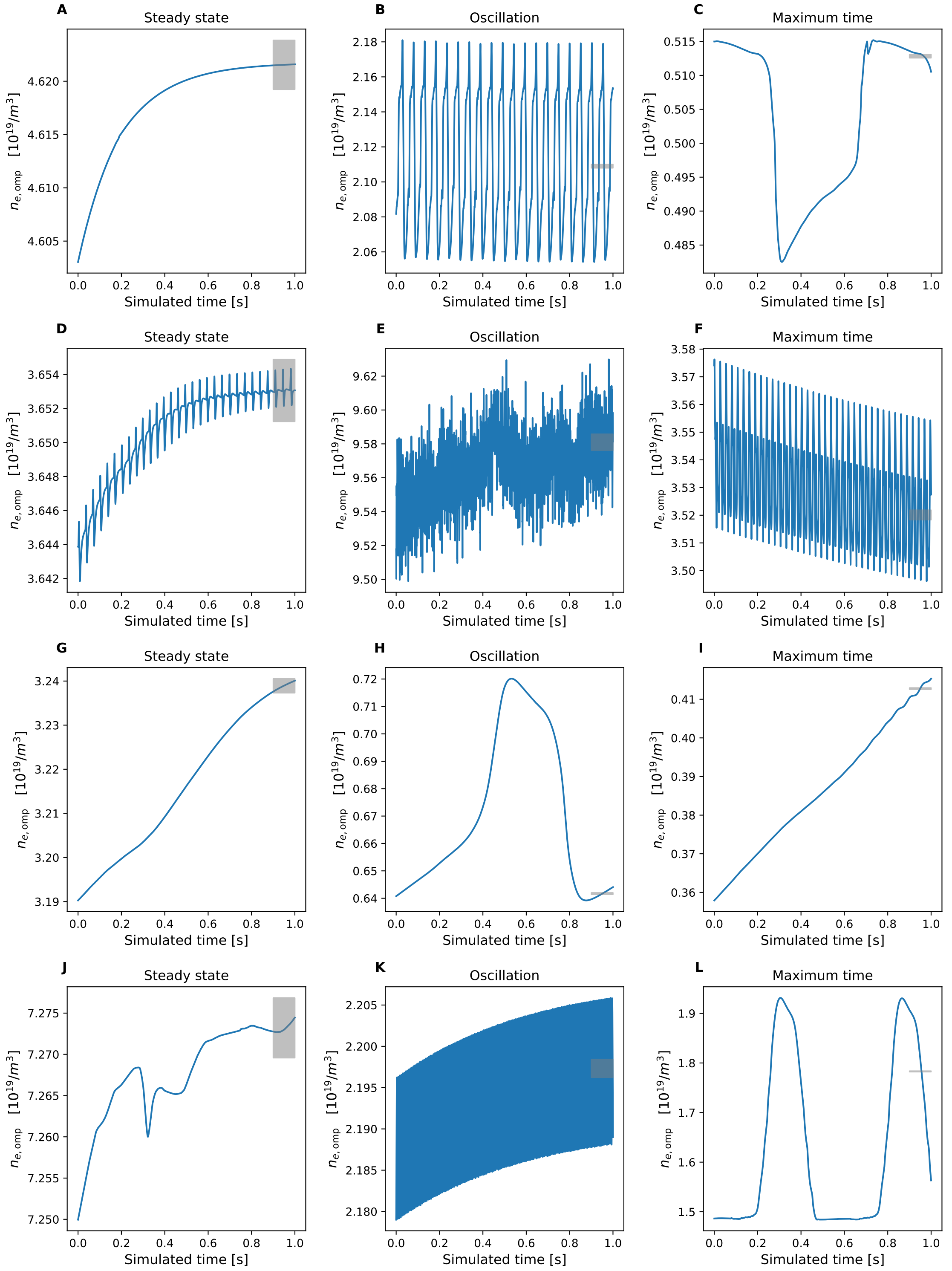


**Figure 7.3:** Time traces of the outer midplane separatrix electron density $n_{e,\mathrm{omp}}$ in examplary simulations over the last second before they are stopped because they are either classified to have reached a steady state, a stable oscillation or reached the maximum runtime of 10 s. The shown simulations are hand selected to display the varying characteristic behaviours that are seen. If the time trace remains fully in the grey box at the end it is classified as steady state.

for numerical reasons and do not achieve results that are close to the actual steady state output. But it is unknown if this is the case, and it can not be easily tested. Also the same can happen in principle also for the steady-state cases due to the simple convergence metric. Figure 5.7 in Section 5.3 has shown that some oscillations occur when the timestep is too large. Because the simulation database covers different physical regimes, it is likely that the fixed timestep of $10^{-5}$ is too large in some situations while it is sufficient in others. Nevertheless, oscillations in the SOL with stationary boundary conditions have been seen both in experiments and simulations, and physical mechanisms behind these have been derived [192]–[195]. Especially [191] analyses the oscillations in similar fluid neutral simulations as used here and observes a cycle between pumping and radiation. Without an extensive designated study into the origin of the oscillations, it can not be determined which fraction of the oscillations in the database stems from numerical artifacts or valid physical mechanisms. But Figure 5.7 shows that even in the case where the oscillations are purely a numerical phenomena, the solutions oscillate closely around the true solution, which would have been found with different settings. Therefore, we estimate that also the results of the oscillating simulations are representative results for the simulation input parameters. To reduce oscillating cases to a representative steady-state value, a straight forward approach would be to take the mean value averaged over some time interval. But due to the drastically differing oscillation frequencies, it is questionable what time interval is adequate. Also calculating a time-average comes with practical difficulties. Not for all state variables a time-dependent output is implemented in SOLPS-ITER and doing so would require large hard drive space. For example, saving all 22 state variables on the 104x50 grid every 10 timesteps for a time interval of 0.1s, results to 8.5 GiB for each single simulation. For all training and test simulations this would result to more than 85 TiB of storage just for these minimum quantities. It is much simpler if the final values at the last timestep of each simulation are taken as proxy for the simulation result.

Since large parts of the later analysis are concerned with the electron temperature at the outer divertor target, this quantity is used to assess the impact of the variability in the oscillatory cases. To do so, the electron temperature at the outer target is recorded every 10 timesteps for the last 0.1 s of simulation time. Then the mean $\mu_{T_{e,\mathrm{ot}}}$ and standard deviation $\sigma_{T_{e,\mathrm{ot}}}$ over time in each grid cell at the outer target are calculated. The "error" when taking the final values $T_{e,\mathrm{ot}}$ instead of the time average $\mu_{T_{e,\mathrm{ot}}}$ can be quantified by calculating the median absolute error between the two. This results to 0.003 eV for the oscillatory cases, 0.003 eV for the simulations reaching the maximum simulation time and 0.006 eV for the cases converged to a steady state. Compared to the variations in temperatures across several scales seen in the simulation database (see Section 7.4), this difference between the final and the time averaged value is minimal. Figure A.1 shows a graphical comparison between the final and time averaged temperature profiles at the outer target. It is apparent that in the majority of cases the variability due to the oscillations is negligible and both the time averaged and the final plasma values provide similar plasma profiles.

Calculating the coefficient of variation $CV_{T_{e,\mathrm{ot}}} = \frac{\sigma_{T_{e,\mathrm{ot}}}}{\mu_{T_{e,\mathrm{ot}}}}$ (across the last 0.1 s of each simulation) allows for a comparison of the variability in the given stopping criteria (Figure 7.4). It is apparent that the steady state simulations have the lowest

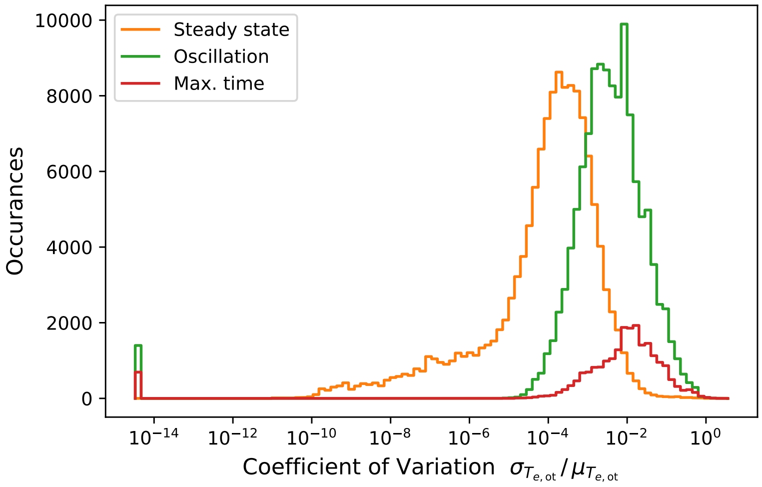


**Figure 7.4:** Histograms of the coefficient of variation $CV_{T_{e,\mathrm{ot}}}$ of the temperatures at the outer divertor target calculated across the last 0.1 s of each training simulation. The colors denote the different condition by which the simulations were stopped. These are simulations reaching a steady state (orange), showing stable oscillations (green) or that reached the maximum of 10 s simulated time (red).

variability with a median $CV$ of 0.02%. The oscillatory cases have higher temporal variability with a median $CV$ of 0.3% and the cases at maximum simulation time the highest with 1%. Taking the final values of all training simulations and calculating the coefficient of variation not across time but across the different simulations yields a median $CV$ of 450%. So also this analysis shows: The temporal variability experienced at the end of the simulations is orders of magnitude lower than the variability between different simulations. But it should be noted that the temporal variability experienced here serves as the lower limit for the accuracy of the later developed surrogate models. Due to the different oscillation frequencies, a surrogate model can not anticipate at which phase of the oscillation a simulation is halted. Because all simulations are halted at a random phase, a surrogate model might even mitigate the effect of the noise in the data and perform the regression to the temporal mean value. But the test data set contains the same noise so if the surrogate accuracy is evaluated on such a dataset this level of noise causes a minimum for the test error that is obtainable. Based on the $CV$ (Figure 7.4), it seems unlikely that the surrogate models will achieve better than 1% relative errors for the temperatures at the outer target. But since the models presented in Chapter 8 obtain only larger test errors, this limitation has no practical relevance.

In conclusion, the dataset contains many simulations that exhibit temporal oscillations or did not reach convergence in 10 s simulated time. It not clear to which extend these are physical phenomena or simply numerical artifacts. Because the temporal variability in these simulations is small compared to the variability across simulations, the analysis in all following sections and chapters relies only on the final values experienced in the last timestep of each simulation. The level of noise this introduces in the training data should impact the developed surrogate models only slightly, so neither the oscillatory nor the max. time simulations are removed from the dataset.

## 7.3 Simulation runtime

Similar as in [141], we use a maximum time for which the simulations are run until they are forcefully stopped. If the simulations are not stopped earlier because they conform to either the convergence metric detecting steady states or stable oscillations, they are stopped at this maximum time regardless of any metric. Here 10 seconds are chosen as the maximum time span while [141] chose a much larger time frame of 100 seconds. This maximum simulation time span is a free parameter which needs to be selected by the model developer and has the potential to impose strong influences on the dataset. Generally two questions arise around this issue: What is the optimal value for the maximum simulation timespan and how should the simulations be treated that exceed it? While it is beyond the possible scope here to find general answers to these questions, this section discusses some of the thoughts around this choice and analyses the present dataset to determine whether the choices made here are reasonable.

Regarding the optimal choice of the maximum simulation time two aspects are in conflict. If the simulations are stopped too early, many simulations will be stopped while they still show some transient time evolution. The observed results in those simulations would differ from the steady state behaviour and will be strongly influenced by the exact numerical settings and initial state of the simulations. This would introduce systematic influences in the surrogate models trained with these simulations and reduce their physical accuracy. But if simulations are simulated for longer maximum time spans, this make the data generation process less efficient as the efficiency is not only determined by the number of simulations needed to train a model but also the compute time required to get each simulation result.

Increasing the maximum simulation time has only an effect on simulations that are still in a transient phase. Figure 7.3 shows that several of the max. time cases exhibit oscillations that are not captured in the oscillation metric. The higher the maximum simulation time, the more likely it is that significant parts of the non converged simulations exhibit uncaptured oscillations rather than very slow transients. Relaxing the metric which detects stable oscillations would likely reduce the number of max. time simulations in favor of detecting oscillations but this would also increase the amount of wrongfully classified and therefore prematurely stopped simulations. Determining the best trade off between capturing all oscillations and as few other cases as possible will require a designated study into oscillation metrics.

Figure 7.5A depicts the time it took until the simulations in the database were stopped depending on the different condition that caused the simulation to stop. The oscillations occur only in simulations stopped after 2 seconds because the design of the oscillation metric does not detect oscillations in the first 1 second segment. The number of simulations which are stopped decreases continuously with increasing simulation time, and only at 10 seconds it is increased drastically due to the simulations that are forcefully stopped. Also after 10 seconds of simulated time, some simulations still reach either of the two convergence metrics but the fraction is a lot smaller compared to the simulations that were forcefully stopped. So increasing the maximum allowed time beyond 10 s will increase the number of simulations

**Figure 7.5:** Histograms of the total runtime of non diverging (A, C) and diverging/crashing (B) simulations. A: The colors denote the different conditions that stopped a simulation. Such are reaching a steady state (grey), showing stable oscillations (olive) or reaching the maximum runtime (cyan). C: The colors denote the different physical regimes seen in the simulations. The classifications into the four regimes sheath-limited (blue), attached (orange), detached (green) and cold core (red) is performed based on the criterion introduced in Section 7.4. This analysis is conducted on half of the simulations in the test set, as only for these simulations the exact timings were recorded.

that reach a steady state or stable oscillations but the return for each added second is diminishing. Exceedingly long times would be required to drastically reduce the number of simulations which are forcefully stopped.

Besides the lack of oscillatory simulations after one second, there are no differences between steady state and oscillatory cases (Figure 7.5A). Neither of them becomes more likely with increasing simulation times. In contrast, most simulations crash either during the first second of runtime or not at all. Therefore, the number of diverging simulations after the first second is very low (Figure 7.5B). A more in-depth analysis could provide finer graded insight into the exact stopping time of these simulations inside the first second. Nevertheless it can be concluded that simulating beyond one second should be encouraged because the probability of failing simulations is low and every simulation is likely to provide some result. When failing simulations crash so early, they require only a small fraction of the overall compute time budget. Therefore to make the data generation process more efficient, it is more beneficial to speed up the convergence of running simulations instead of trying to mitigate the occurance of diverging simulations. Here no answer can be given concerning what the optimal maximum simulation runtime is. To do so, it would require a longer designated study, which tracks the physical values throughout the whole time to determine the amount of changes that are expected for times longer than ten seconds.

Regarding the question whether the results of the simulations forcefully stopped after 10 seconds should be included in the dataset, it has to be considered that the physical regimes seen in the simulations might be correlated with the time it takes for these simulations to converge. If there are some correlations then excluding these simulations would mean excluding particular sections of the parameter space or of the physical regimes. Another aspect of the question is that even if they are not converged they still might be beneficial for the surrogate. The surrogate model might still learn something from these simulations since they are already run for

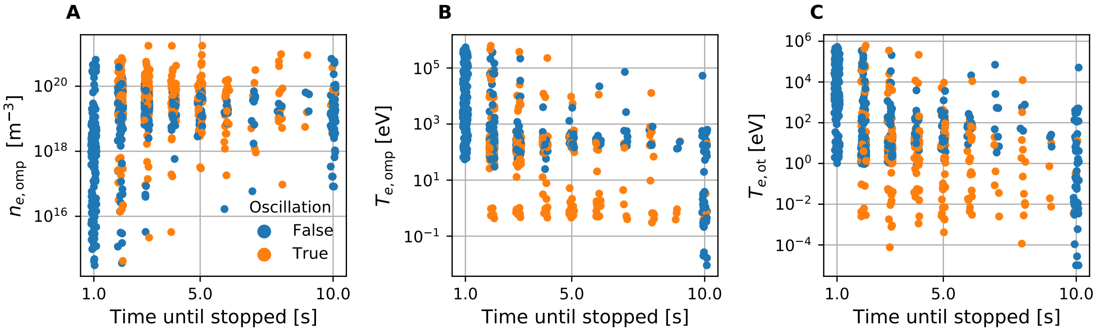


**Figure 7.6:** Electron densities (A) and temperatures (B) at the outer midplane separatrix and electron temperatures at the outer target separatrix (C) in each simulation against the total simulated time of the simulations. Orange dots represent simulations stopped due to stable oscillations being detected, while blue dots show simulation stopped due to another condition. All results are from simulation that did not diverge. This analysis is conducted on half of the simulations in the test set, as only for these simulations the exact timings were recorded.

10 seconds and therefore likely close to the actual solution. This really depends on whether or not the added amount of noise in these added data points outweighs the benefit of having more data points in general. Looking back at the coefficients of variation in Figure 7.4 shows that the forcefully stopped simulations have similar variability as the oscillatory cases, therefore discarding the max. time cases based on an argument of too low signal-to-noise ratio seems unwarranted.

Figure 7.5C investigates how the regimes seen in the simulations correlate with the time at which they are stopped. The definitions of the regimes are given in Section 7.4. The sheath-limited cases seem to converge the fastest while the other cases take longer and almost no simulations from the other regimes are stopped during the first second. In the forcefully stopped simulations after 10 seconds around half of the simulations are in the cold core regime, while the other half is in the other three regimes. So while the cold core regime takes longer to converge, this is not exclusive but simulations in the other regimes can also take long times. Figure 7.6 further investigates how the simulation time affects the physical parameters seen in the simulations. Figure 7.6A shows that cases with low electron density at the outer midplane separatrix appear most prominently with low simulation times and become less and less over time. Interestingly all cold core cases with electron temperatures at the outer midplane separatrix of less than 10 eV are either stopped after 10 seconds or classified as oscillation (Figure 7.6B). While there is some connection between these oscillatory cases and the cold core cases, the oscillations are not exclusive to the cold core cases but also appear in simulations with higher upstream temperatures.

As conclusion it can be summarized that simulating longer than one second is required to also bring non sheath-limited cases to convergence and that the convergence behaviour is linked to the physical regimes seen. In the following analysis and the development of the surrogate models the results from all simulations will be used (that did not diverge) even those that are forcefully stopped after the maximum ten second runtime. Excluding these simulations would exclude the majority of the cold core cases and we expect the added information of these simulations to outweigh the errors they might introduce due too lack of convergence. However, a more thorough investigation, in particular in cases where more strict convergence metrics are used, might conclude that another strategy is superior.

## 7.4 Plasma regimes

The validity of the developed surrogate models is restricted by the validity of the underlying simulations. Due to the fluid neutral model, the results of the simulations will be less accurate than in higher fidelity simulations (Section 4.3). The goal of this section is to determine which of the different physical regimes and effects are covered by the developed database.

Because in the simulation database multiple parameters are varied at once, it is much more difficult to detect some features used for classifying the different divertor regimes and especially detachment (e.g. rollover) than compared to a density scan of a single tokamak scenario. Because the different SOL regimes are fundamentally linked to the temperature gradients present (see Section 2.3), we categorize the simulations solely by comparing the electron temperature in the first grid cell outside the separatrix at the outer midplane against the electron temperature in the first grid cell next to the separatrix at the outer divertor target (Figure 7.7A). This categorization has the added benefit that the regimes align with the absolute temperature scales. In 27% of the training simulations (Table 7.1) these temperatures differ by less than 20%, which suggests that these are in a sheath-limited regime (see Section 2.3). In this case the only meaningful temperature gradient occurs in the sheath, which is not modeled in these simulations (only as a boundary condition). Simulations with larger temperature gradient and a electron temperature directly outside of the outer target separatrix $T_{e,\mathrm{ot}}$ above 5 eV are called "attached" in the following while similar simulations with a $T_{e,\mathrm{ot}} < 5\,\mathrm{eV}$ are called "detached". By using such a broad distinction, not all cases labeled as detached will show all signs associated with detachment but some simulations with $T_{e,\mathrm{ot}} < 1\,\mathrm{eV}$ maintain upstream temperatures $T_{e,\mathrm{omp}}$ of around 100 eV (Figure 7.7A) demonstrating that significant heat mitigation takes place and volume recombination is a relevant effect. With this distinction 38% of the training simulations are attached and 20% are detached (Table 7.1). Although these low detached target temperatures are present, it should be considered that the simulations do not contain any molecular effects, which become highly relevant for the particle and power balance at these temperature scales. Regardless of the temperature gradients and target conditions all simulations with an outer midplane separatrix temperature below 10 eV are categorized into the "cold core" category. This concerns 27% of the training simulations (Table 7.1). Figures 7.7B, C depict the temperature statistics in the different regimes. For the sheath-limited cases the temperature distributions are almost entirely identical at the outer midplane and outer target. The detached and attached cases have very sharply peaked distributions for $T_{e,\mathrm{omp}}$ with maximums at 123 eV and 179 eV, while the distributions at the target are much more diverse. This small variation in upstream temperatures with larger variation at the target is characteristic for the SOL both in experiments and simulations (see e.g. [178], [196]). The developed surrogate models will have to

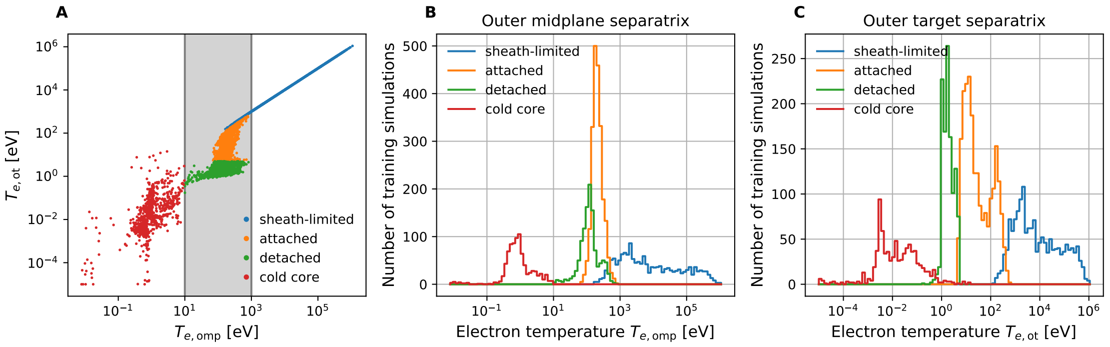


**Figure 7.7:** Electron temperature at the outer target separatrix against the electron temperature at the outer midplane separatrix in all training simulations (A). Histograms of the temperature at the outer midplane (B) and outer target (C) separatrix in all training simulations. The colors denote the different physical regimes of the simulations: Sheath-limited (blue), attached (orange), detached (green) and cold core (red).

| | all | sheath-limited | attached | detached | cold core |
|---|---|---|---|---|---|
| Number of simulations | 5756 | 1549 | 2216 | 1165 | 826 |
| Relative fraction | 1.0 | 0.27 | 0.38 | 0.2 | 0.14 |

**Table 7.1:** Absolute and relative numbers of simulations in each of the distinguished physical regimes in the training set.

deal both with the different statistics at different domain locations as well as with the overall large range of temperatures present, spanning several orders of magnitude. The test set to evaluate the developed models is created in similar fashion as the training simulations and therefore follows similar statistics (not shown here).

Radiation plays an important role for the power balance as it is the main volumetric power loss. The SOL regimes should be correlated with the fraction of the input power $P_{\text{in}}$ that is radiated away. In the sheath-limited regime almost no radiation takes place at all, neither in the SOL nor in the confined plasma. In most of the sheath-limited simulations less than 1% if the power entering the simulation is dissipated by radiation (Figure 7.8). In the attached and detached simulations the radiation occurs predominantly in the SOL, divertor and PFR (Figure 7.8A). In most of these simulations, the fraction of the total input power that is radiated in the confined plasma is way below 10%. The total amount of power that is radiated stretches between 5% to over 90%. Above 80% power dissipation almost no simulations are attached anymore but the simulations with low core radiation are predominantly in the detached regime. In contrast, in the cold core simulations the radiation occurs almost exclusively in the confined core plasma. In most cases less than 5% of the input power is radiated outside of the core. Figure 7.8A highlights the regime transition between the cold core and the attached and detached cases. There are hardly any simulations in which the radiation occurs inside the confined and on the open field lines, but rather exclusively either or the other. While the cold core simulations start to appear already when only 20% of the input power is radiated (Figure 7.8A), the overwhelming majority of these simulations has a total radiation of more than 90% of the input power (Figure 7.8B). The value of the nitrogen gas puff is one of the major determining factors for the amount of radiation in the simulations (Figure 7.8C). Yet below $10^{20}$ atoms/s the occurring radiation is not correlated with the nitrogen gas puff and thus determined by other factors. In these conditions, the radiation in the simulations reaches from 0 to 50%. For higher gas puffs Figure 7.8C shows a clear correlation with the total radiation in the simulations. Above a nitrogen gas puff of $10^{22}$ atoms/s no sheath-limited simulations are appearing. Also noticeable is that for any level of nitrogen gas puff the radiation occurring in the detached simulations is always higher than in the attached cases. This also shows as a shift of the distribution of detached cases against the attached cases in Figure 7.8B. The attached and detached scenarios for low nitrogen gas puff contain high deuterium throughput (Figure 7.8D). Deuterium gas puff values larger than $10^{23}$ atoms/s introduce some radiation, such that no sheath-limited simulations are appearing. Yet even for the maximum gas puff values radiation levels below 20% are observed. This depicts clearly that the simulations require the introduced nitrogen gas for significant enhancement of the radiation. In the cold core regime, the power crossing the separatrix from the core plasma into the SOL is strongly diminished, due to the high amount of radiation in the core. This situation resembles an x-point radiator (XPR) regime, which contains strong radiation inside the confined plasma [197]. But unlike the cold core cases here, in XPR discharges the temperature at the outer midplane usually stays high and decreases on the closed field lines only close to the x-point [198]. Obtaining such a clear spatial distribution using the simple fluid neutral model is likely not possible. Therefore, it is questionable whether the cold core simulations could be interpreted as XPR scenarios.

Figure 7.9 depicts the dependence of various quantities at the outer and inner divertor target against the electron density at the outer midplane separatrix. In the sheath-limited regime, the electron density at both targets is linearly correlated with the upstream density although the density at the targets is lower (Figure 7.9A, B). For low upstream densities only the sheath-limited regime is possible. In these cases the densities at both targets are identical. The sheath-limited cases extend to upstream densities of less than $10^{15}\,\mathrm{m}^{-3}$. The attached and detached regimes exist only for upstream densities higher than $10^{18}\,\mathrm{m}^{-3}$. Here the plasma density at both targets is increased and can be up to two orders of magnitude larger than the upstream density. Most simulations in these regimes have upstream densities between $10^{19}\,\mathrm{m}^{-3}$ and $10^{21}\,\mathrm{m}^{-3}$. At the outer target the detached cases tend to have higher target densities than the attached cases (Figure 7.9A). At the inner target both attached and detached cases have similar target densities (Figure 7.9B). This is likely because the classification into detached and attached is done purely based on the temperature gradients towards the outer target. For standard vertical target configurations, the inner target usually has lower temperatures, so possibly all attached cases show already signs of detachment at the inner target. Another difference between the targets is that at the outer target the densities increase continuously from the sheath-limited cases while at the inner target the target density jumps from one regime to the other with a visible gap, in which only few simulations exist (Figure 7.9B). The outer target density in the detached simulations maintains a positive correlation with the upstream density while for the inner target this correlation is less clear. Going from the attached to the detached cases, no clear rollover of the electron density can be observed at either of the targets although at the inner

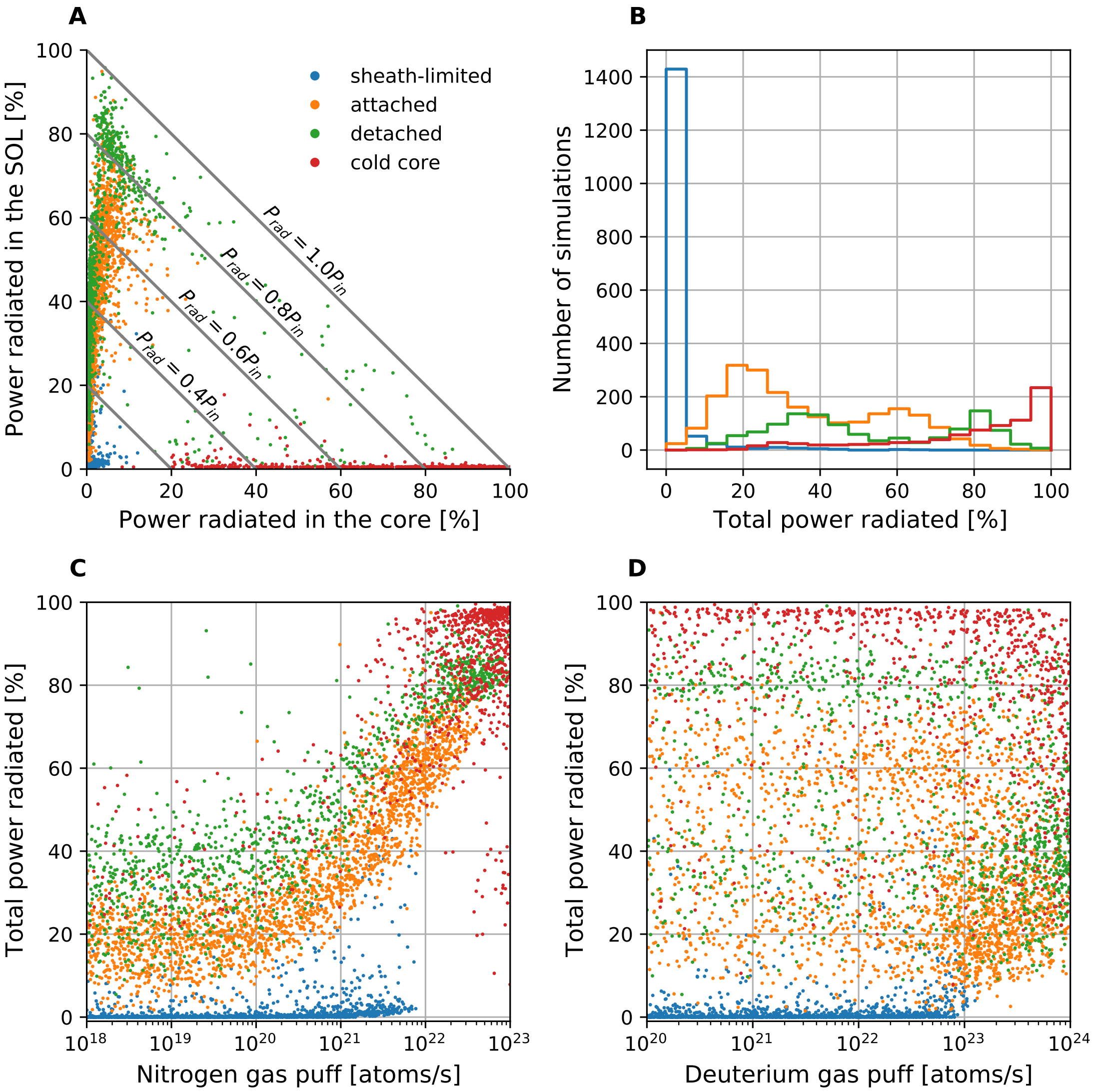


**Figure 7.8:** Overview of the line radiation in the simulations. The line radiation is calculated as fraction of the input power $P_{\text{in}}$ that enters each simulation. A: The power radiated in the SOL (including divertor and PFR) against the power radiated inside the separatrix for each simulation. The diagonal lines highlight the total amount of radiated power in all regions. B: Histograms of the total amount of radiated power. C: The total amount of radiated power against the nitrogen gas puff for each simulation. D: The total amount of radiated power against the deuterium gas puff for each simulation. The colors denote the different physical regimes of the simulations as in Figure 7.7: Sheath-limited (blue), attached (orange), detached (green) and cold core (red).

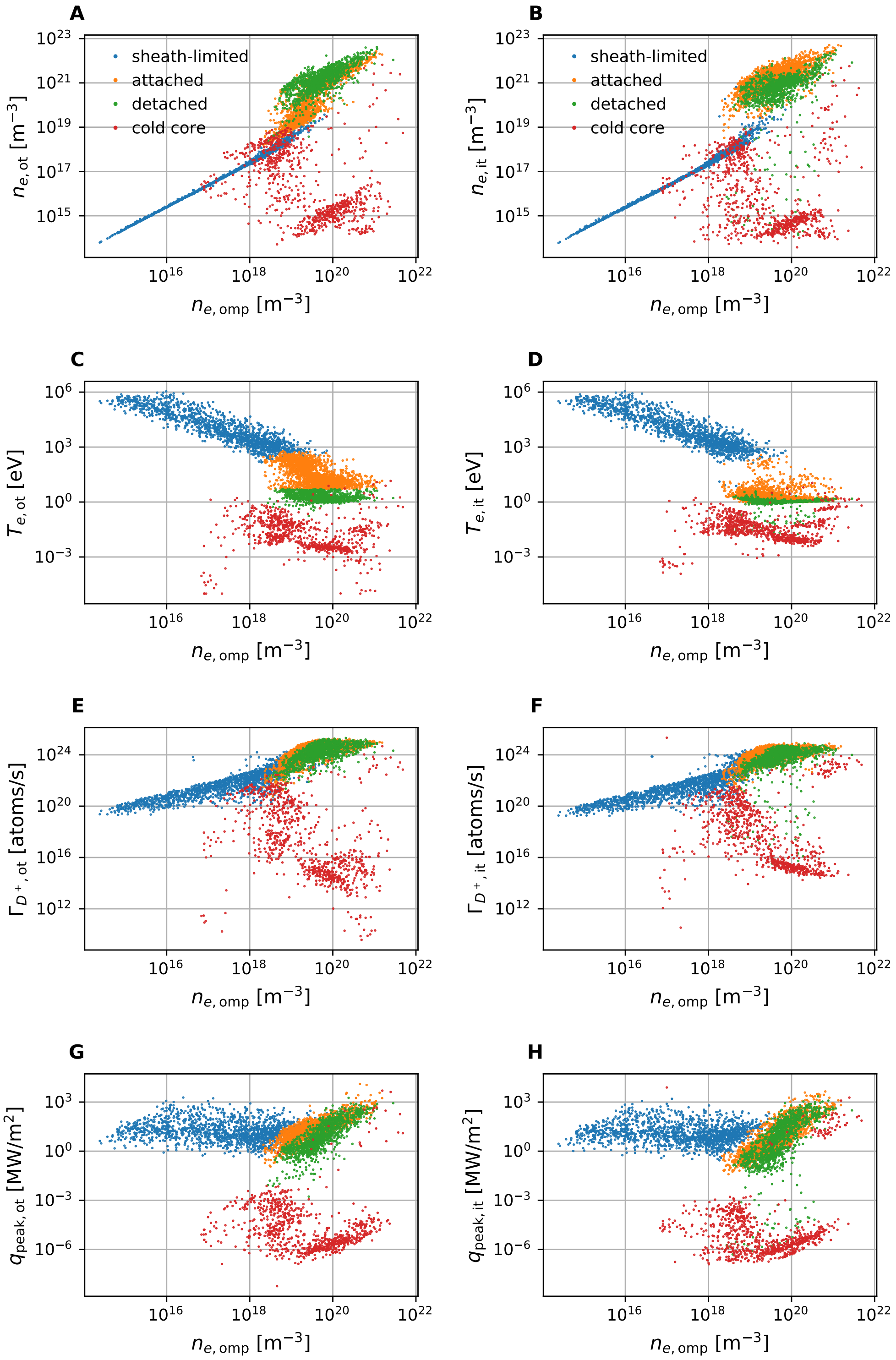


**Figure 7.9:** Plasma properties at the outer (A, C, E, G) and inner (B, D, F, H) divertor target against the electron densities at the outer midplane separatrix $n_{e,\mathrm{omp}}$ in the training simulations. The target densities $n_{e,\mathrm{ot}}$, $n_{e,\mathrm{it}}$ and temperatures $T_{e,\mathrm{ot}}$, $T_{e,\mathrm{it}}$ are measured directly outside the separatrix, while the deuterium ion fluxes $\Gamma_{D^+,\mathrm{ot}}$, $\Gamma_{D^+,\mathrm{it}}$ are integrated across each divertor target and the peak heat flux $q_{\mathrm{peak,ot}}$, $q_{\mathrm{peak,it}}$ are the maximum values at each target. The colors denote the different physical regimes of the simulations: Sheath-limited (blue), attached (orange), detached (green) and cold core (red). The high peak heat fluxes of some attached and detached simulations might be artifacts of the boundary conditions in the PFR (see Section 7.5).

target the electron densities are slightly lower in many detached cases than in the attached scenarios (Figure 7.9A, B). The cold core cases show an entirely different behaviour. While some show similar density relations as in the sheath-limited cases, the majority lies around upstream densities of $10^{20}\,\mathrm{m}^{-3}$ with exceptionally low target densities around $10^{15}\,\mathrm{m}^{-3}$.

As expected, the temperatures at the targets generally decrease with increasing upstream density. This is seen as a linear trend in the sheath-limited cases, which becomes non-linear for the attached and detached conditions (Figure 7.9C, D). For the lowest densities, the target temperatures in the sheath-limited regimes reach values beyond $10^5\,\mathrm{eV}$. This is only possible because the simulations do not consider the erosion of the divertor targets self-consistently. In reality such high target temperatures likely cause extreme levels of sputtering, and consequently the eroded divertor material will enter the plasma and cool it drastically. It is highly unlikely that under such conditions a stable plasma can be maintained. While these simulations constitute valid solutions to the solved plasma equations, they are unphysical in the larger picture. In the sheath-limited regime the temperatures at outer and inner target are fairly similar (Figure 7.9C, D). Upon entering the attached regime, the decrease in target temperatures at the outer target is continuous from the sheath-limited regime as opposed to the inner target that exhibits a discrete jump of the temperatures. This leads to the majority of attached cases having an inner target temperature of less than $10\,\mathrm{eV}$ and only few cases with temperatures between $10\,\mathrm{eV}$ and $100\,\mathrm{eV}$. This jump in temperatures is consistent with the jump in densities seen at the inner target (Figure 7.9B) but while the inner target detaches earlier this does not explain the apparent lack of target temperatures in the $10 - 100\,\mathrm{eV}$ range. At the outer target the separation between attached and detached regimes is by design of the classification metric. The attached cases have slightly higher temperatures at the inner target than the detached cases but both groups still show a large overlap. The upstream densities at which regime transitions occur depend on many of the parameters varied in the simulation database [71], [74], [199]. Therefore, it is clear that in the density ranges where such transitions are expected ($10^{19} - 10^{20}\,\mathrm{m}^{-3}$) a large variety of outer target temperatures appears because with similar upstream density different simulations will be in different regimes (Figure 7.9C). The cold core simulations generally have lower target temperatures than all other regimes (Figure 7.9C, D).

One usual diagnostic to determine detachment is a rollover in the hydrogen ion flux towards the target, which can be measured in experiments using Langmuir probes [105]. For a rollover the particle flux at the target would first increase with increasing density and then start decreasing. This is not clearly visible across the dataset here (Figure 7.9E, F). Here both detached and attached cases lie at similar target fluxes, which are higher than in the sheath-limited simulations. Only the cold core cases show a strong decrease in the deuterium ion flux to both targets with high upstream densities. As mentioned earlier, these simulations might have slight resemblance to an x-point radiator regime, which is distorted by the simple fluid neutral model. Since the overall physical validity (or relevance) of these simulations is questionable, it is also questionable whether the reduction in target ion flux should be considered a valid sign of detachment. Nevertheless, a particle flux

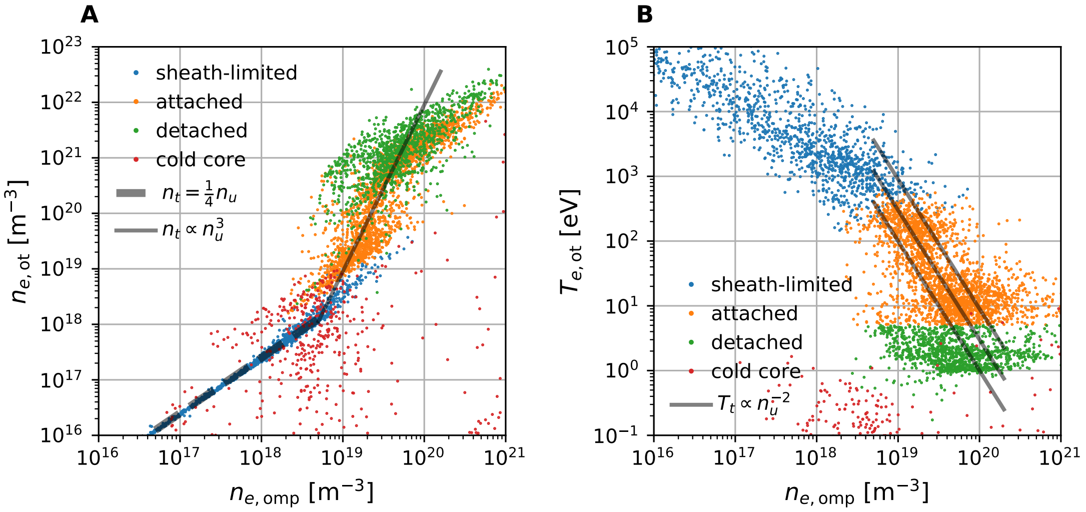


**Figure 7.10:** Comparison between the electron density (A) and temperature (B) at the outer target separatrix with the trends expected by the two point model as function of the electron density at the outer midplane separatrix. The colors denote the different physical regimes of the simulations: Sheath-limited (blue), attached (orange), detached (green) and cold core (red). For a comparison with the two-point model the relations $n_{e,\text{ot}} = \frac{1}{4} n_{e,\text{omp}}$, $n_t \propto n_u^3$ and $T_t \propto n_u^{-2}$ are highlighted by grey lines.

rollover is physically achievable also with high upstream temperatures. The fact that this is not clearly seen in this analysis should not be taken as sign that it is not included in the physics model of these simulations. The particle flux depends on multiple parameters beside the upstream density and the deuterium ion fluxes to the targets in attached and detached simulations stretch across almost three orders of magnitude. Within this group of simulations there is plenty of room for a density depending rollover to hide. Especially since the integrated ion flux depends on the overall variation in tokamak size. In Section 9.3 the developed surrogate models are used to show that a rollover does exist also with high upstream temperatures. This is possible because the surrogate model allows for variation of parameters independently, unlike the data analysis here.

As introduced in Section 1.3, another key requirement of suitable divertor conditions is a peak heat flux towards the targets of less than 10 MW/m$^2$. In the sheath-limited regime the peak heat flux at the outer target lies between 1 MW/m$^2$ and 300 MW/m$^2$, with no visible dependence on the upstream density (Figure 7.9G, H). The attached and detached regimes exhibit similar peak target heat fluxes as the simulations in the sheath-limited regime. But the variability is slightly increased such that values between 1 MW/m$^2$ and 1000 MW/m$^2$ are common (Figure 7.9G, H). There is no strong difference between the inner and outer target. The cold core cases have drastically lower target heatfluxes, because most of the power is radiated before it reaches the targets. The majority of cases exhibit peak target heat fluxes below $10^{-3}$ MW/m$^2$.

For a slightly more in depth analysis the functional relations of the properties at the target can be compared to the scaling laws predicted by the two-point model

(Section 2.3). In the sheath-limited regime the temperature at the target should be identical to the upstream temperature while the electron density at the target should be exactly half of the upstream density. This temperature relation is clearly seen in Figure 7.7A between the temperature at the outer midplane and the outer target. While in the sheath-limited simulations the electron densities at the outer target separatrix are linearly correlated with the densities at the outer midplane, these are not exactly half of the upstream density as predicted but rather a quarter $n_{e,\mathrm{ot}} = \frac{1}{4} n_{e,\mathrm{omp}}$ (Figure 7.10A). In the conduction-limited regime the two-point model predicts a relation $n_t \propto n_u^3$ with additional modulating factors such as the connection length, which are also varied in this dataset. Instead of performing one-to-one comparisons, Figure 7.10A depicts the $n_t \propto n_u^3$ trend scaled such that it matches the target densities of the sheath-limited regime at $n_{e,\mathrm{omp}} = 5 \cdot 10^{18}\,\mathrm{m}^{-3}$. It is clear that at the onset of the attached regime, this trend is roughly followed (Figure 7.10A). The highest density cases from both the attached and detached regimes deviate from this scaling towards lower target densities, which is a sign that stronger volume losses occur. For the target temperature in the conduction-limited regime, the two-point model predicts a dependence $T_t \propto n_u^{-2}$. This relation is shown by the parallel lines in Figure 7.10B. It is apparent that the scaling roughly matches the relation seen for the attached simulations. The detached simulations deviate from this trend and exhibit lower target temperatures, indicating that in these simulations additional SOL losses become significant. From this analysis two thing can be concluded: The simulation database includes effects that go beyond the capabilities of the basic two point model, and the regime classification roughly matches the regime distinction in the two-point model. For a more fine tuned comparison the simulation results could be brought into "2PM Formatting" [103], which allows comparison with the analytical expressions (Equations 2.20,2.21,2.22) to identify the momentum $f_{mom}$ and power $f_{pow}$ loss factors along the magnetic field lines.

As a next step it is investigated how the machine parameters influence the SOL regimes. Again in this analysis one should be aware of the survivorship bias. Only the trends in the non diverging simulations can be observed, therefore the findings might change a bit if the numerical setup causes other sets of simulations to succeed. Also only trends different between the regimes are relevant as trends seen in all regimes appear by the changes in the overall success rate of the simulations (see Section 7.1). The dependence of the regimes on each simulation parameter individually is visualized as histograms in Figure 7.11. The strength of the parameters influence varies drastically across the different parameters. As expected the parameter with the least influence is the toroidal magnetic field strength. The distribution of all four regimes is almost uniform between minimum and maximum values of the magnetic field strength (Figure 7.11B). Another relatively weak influence stems from the cross-field transport coefficients. The density transport coefficient $D_\perp$ has almost no influence besides a reduction of the number of sheath-limited cases for transport coefficients between $0.4 - 1.2\,\mathrm{m}^2/\mathrm{s}$ (Figure 7.11G). The heat transport coefficient has a slightly stronger effect (Figure 7.11H). The amount of attached and sheath-limited cases decreases for coefficients below $0.5\,\mathrm{m}^2/\mathrm{s}$ but remains stable above. The number of detached cases continuously increases with increasing this cross-field transport, which matches the expectation because stronger cross-field transport broadens the heat and temperature distribution in the SOL. The influence of the variations in

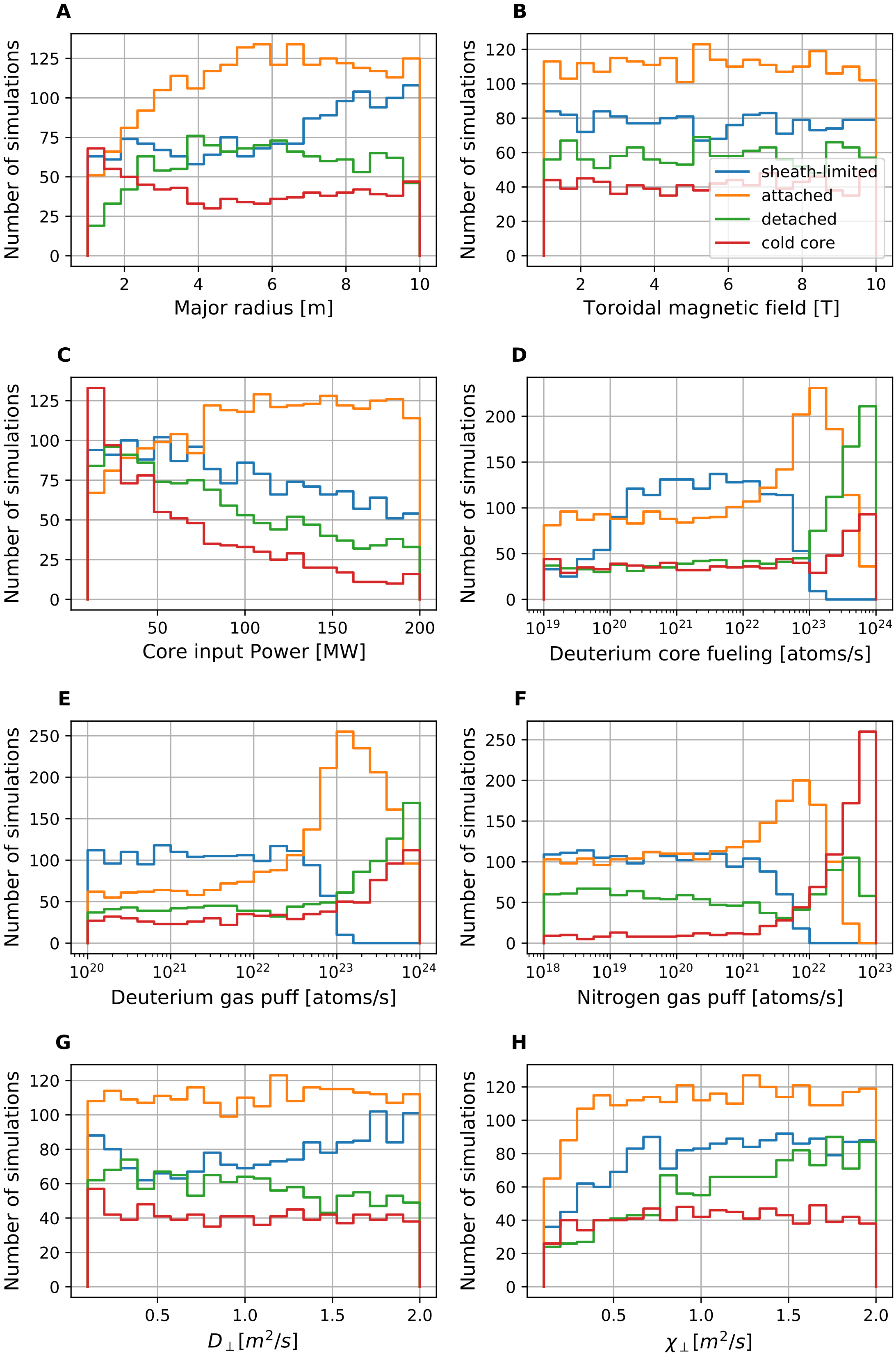


**Figure 7.11:** Histograms of the number of non diverging simulations over varying parameter values in the training set separated into the four distinguished physical regimes: Sheath-limited (blue), attached (orange), detached (green) and cold core (red).

the major radius is slightly stronger (Figure 7.11A). Increasing major radius size, the number of attached and sheath-limited simulations rises. Below four meters the number of cold cores cases seems to rise while the number of detached cases decreases. This does not have to be an effect of the major radius alone but because the range of tested input powers is quite high for such small devices, the possible heat dissipation might be not enough to reach detachment. But this explanation is in contradict to the increasing number of cold core cases for sizes below 4 m as these would also have to decrease for the same reason. A possible explanation for this increase in cold core cases is that the distance between domain boundary at which the gas puff is inserted and the core plasma is smaller in smaller devices (see Section 4.1) and since the neutral transport is not restricted by the magnetic field it might be easier for neutrals to enter and cool the confined plasma. For all regimes except the attached regime, the number of simulations decreases with increasing input power while in contrast the number of attached cases increases with a jump at 80 MW (Figure 7.11C). Interestingly the sheath-limited cases and the cold core cases which are opposite ends of the temperature spectrum show similar dependence on the input power. For the cold core cases its absolutely clear that smaller input powers will lead to more cases as the input power might be too little compared to the size or the cooling effects. For the sheath-limited cases the explanation is most likely linked to the overall success rate of the simulations, which shows a similar trend (Figure 7.1C). If increasing input power makes simulations with low density less stable, this would result in only the attached and detached cases with higher densities surviving. The number of attached cases is then rising, since with otherwise equal heat mitigation an increase in the input power will lead to higher target temperatures. By far the strongest influence on the regimes stems from the three density controlling parameters $D_{\mathrm{core}}$, $D_{\mathrm{puff}}$, and $N_{\mathrm{puff}}$. For all three parameters an increase in particle influx leads to a decrease in the number of sheath-limited cases and an increase in all other three regimes (Figure 7.11D, E, F). Interestingly the breakdown of the sheath-limited cases is rather abrupt at fixed threshold values and not a continuous decrease. This breakdown appears at similar values for both the ion influx at the core as well as the deuterium gas puff while for the nitrogen gas puff it is an order of magnitude lower. However, contrary to intuition the number of sheath-limited cases is higher for ion core influx between $10^{19} - 10^{20}$ atoms/s than for lower values (Figure 7.11D). For all three parameters an increase of the input fluxes first increases the number of attached cases, which then decreases for higher fluxes while the number of detached and cold core cases rise. The nitrogen gas puff values has a much stronger influence on the appearance of cold core cases than either of the deuterium influxes. The most cold core cases appear for nitrogen gas puff values above $10^{22}$ atoms/s (Figure 7.11F). For the two deuterium controlling parameters the number of detached simulations is always higher or similar than the number of cold cases throughout the whole parameter range (Figure 7.11D, E). While for increasing nitrogen gas puff the number of detached simulations also increases this increase is far less pronounced than for the cold core cases (Figure 7.11F). Therefore, the ratio of detached to cold core cases is in favor of the detached simulations for low nitrogen gas puff, while beyond $10^{22}$ atoms/s this relation reverses.

Upon examining the second-order correlations (Figure 7.12), there are almost no exclusive regions where only a single regime is solely present. One exemption is the

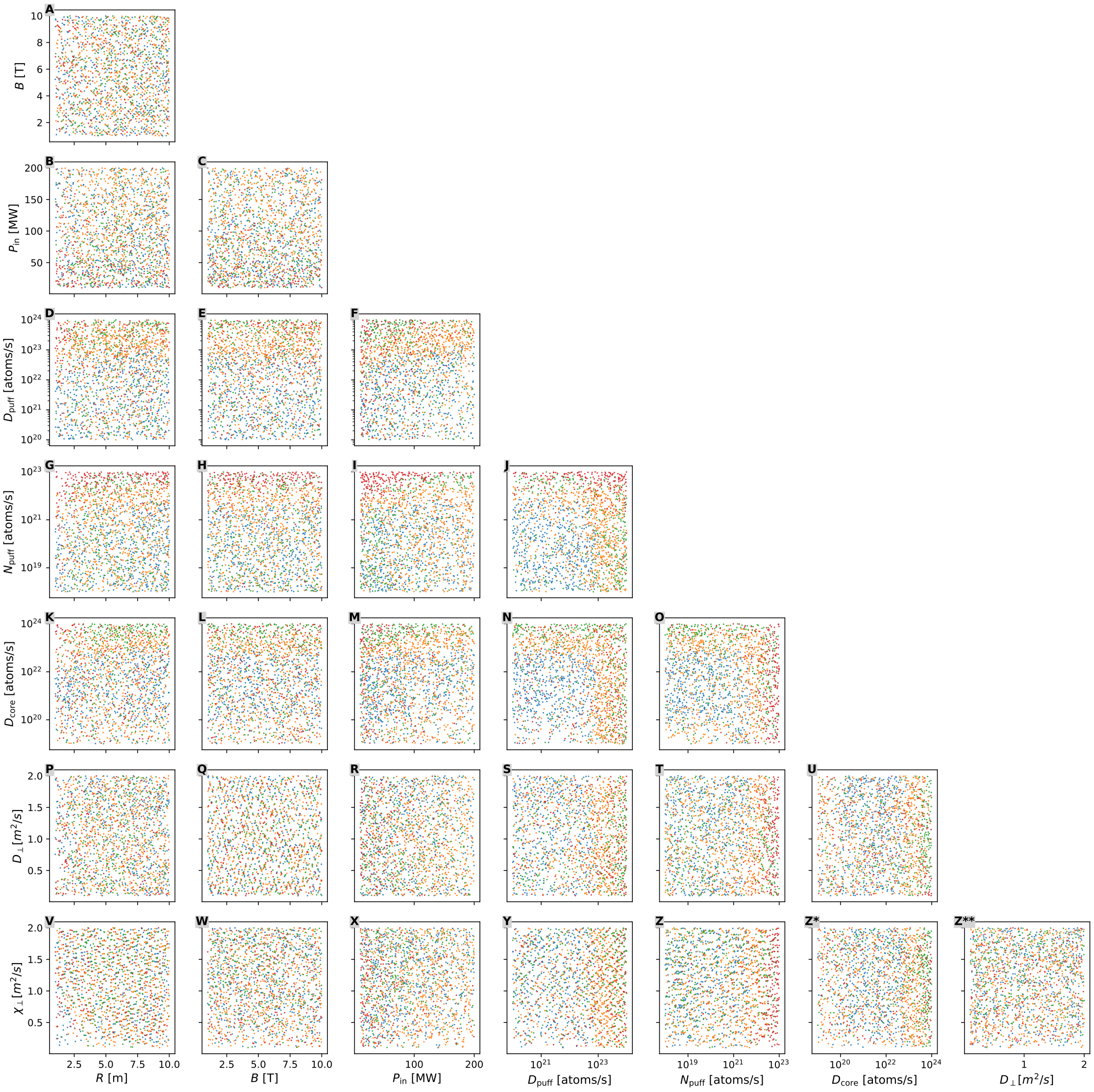


**Figure 7.12:** Each dot represents a non diverging simulation from the training data set, color coded into different physical regimes: Sheath-limited (blue), attached (orange), detached (green) and cold core (red). For visibility only 30% randomly selected from all non diverging training simulations are shown.

occurance of cold core cases. In the top left corner of Figure 7.12I, a region exclusively occupied by cold core cases emerges for large nitrogen gas puff and low input power. However, a significant number of cold core cases are distributed throughout the entire domain for other values of input power and nitrogen gas puff. While simulations of this regime appear anywhere in the parameter space, the majority of such cases occurs at high nitrogen gas puff values (Figure 7.12I). Similar as in Figure 7.11 also here for all three density-controlling parameters ($D_{\mathrm{core}}$, $D_{\mathrm{puff}}$, $N_{\mathrm{puff}}$) threshold values can be seen below which simulations are mostly in sheath-limited regimes and above it almost no sheath-limited cases exist (e.g. Figure 7.12J, N, O). Further increase of any of the three parameters then leads to a shift from a majority of attached cases to more predominantly detached situations. Both deuterium ion influx as well as deuterium gas puff require the same value of $7 \cdot 10^{22}$ atoms/s for the transition out of the sheath-limited regime. Decreasing the core fueling below $10^{20}$ atoms/s will cause more simulation crashes and therefore a lower density of successful simulations if neither of the gas puffs is high enough to lift the density out of the sheath-limited regime (Figure 7.12N, O). In the cases where this is achieved solely through a nitrogen gas puff, it is likely that the plasma will primarily consist of nitrogen instead of deuterium ions.

The most important finding of this analysis is that the simulation setup produces results in all plasma regimes. Specifically it is possible to achieve each regime for all tokamak sizes, input powers and transport coefficients. The trends seen in these regimes generally follow the expectations. For some simulations (e.g cold core or very high temperature sheath-limited) it is questionable whether these provide physically reasonable results if factors beyond the solved fluid equations are taken into account. Nevertheless, a surrogate model might still benefit from such cases as it can learn which regions of the parameter space lead to unreasonable results. But the majority of the simulations provide results in both a reasonable and relevant regimes. So overall the database seems suitable for developing surrogate models.

## 7.5 PFR temperature spike

The statistical analysis in the previous section has come to the conclusion that the simulation database shows overall reasonable trends and is thus usable for surrogate development. However, upon closer inspection of simulations with low target temperatures it was found that some of these simulations exhibit temperature profiles at the outer target with an odd shape. Specifically the temperature exhibits a peak at the lower end of the outer target which is inside the PFR (see e.g. Figure 8.8). Similar profile shapes are not usually seen whether in simulations or in experiment. Because the PFR is magnetically isolated, the temperatures in the PFR are usually lower than in the SOL. Therefore, the question arises whether this is an actual physically sound effect or an artifact of the simulation procedure. To find the cause of this behaviour, it is necessary to determine how many cases are affected and to find what these have in common or what distinguishes them from other simulations.

To determine whether such a peak is present the ratio between the electron temperature at the bottom corner in the PFR of the outer target $T_{e,\mathrm{ot}}^{\mathrm{PFR}}$ and the temperature at the separatrix at the outer target $T_{e,\mathrm{ot}}^{\mathrm{sep}}$ can be calculated. Figure 7.13 depicts this

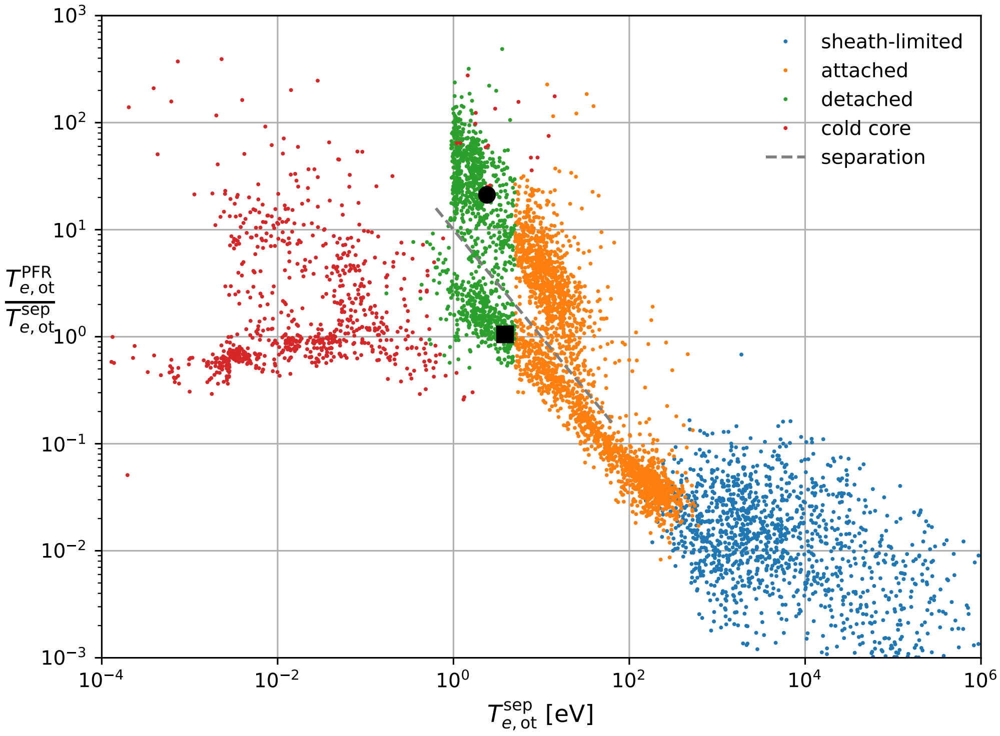


**Figure 7.13:** Ratio between the electron temperature at the outer target at the PFR boundary and the separatrix against the electron temperature at the outer target separatrix in the training simulations. The colors highlight the different physical regimes: Sheath-limited (blue), attached (orange), detached (green) and cold core (red). The visualization is zoomed in such that not all sheath-limited simulations are seen. For the analysis in Figures 7.14 and 7.15 the detached simulations are separated into the group above or below the marked grey separation line. The black circle (above) and square (below) highlight the positions of the two simulations investigated in Table 7.2 and Figures 7.16 and 7.17.

ratio against the temperature at the outer target separatrix $T_{e,\mathrm{ot}}^{\mathrm{sep}}$. Disregarding the cold core cases, where the relations are all over the place, a clear trend is visible. With increasing temperature at the separatrix, the temperature at the PFR becomes an increasingly smaller fraction. For target temperatures between $1 - 30\,\mathrm{eV}$, a split emerges into two distinct groups. The first branch, below the separation line in Figure 7.13, follows the trend which is already present for higher target temperatures. This trend leads to only slight temperature increases in the PFR such that these reach $2 - 3\,\mathrm{eV}$ for a separatrix temperature of $1\,\mathrm{eV}$. In the second branch, above the separation line in Figure 7.13, the simulations with separatrix temperatures of $1\,\mathrm{eV}$ have temperatures at the PFR boundary of the target between $10 - 200\,\mathrm{eV}$. While the magnitude of this relative temperature increase is greatest for the lowest temperature detached cases, the same mechanisms are also present at much higher temperatures in the attached regime in this second branch of simulations.

To investigate the differences between the two branches of simulations, all cases in the detached regime are sorted into either of two groups depending on which side of the separating line they are one in Figure 7.13. The group "below" exhibits no or only small PFR temperature peak, while the group "above" exhibits the large peak. In a similar analysis as before, the dependence of the occurrence of these two groups on the simulation input parameters is analysed. Here it is only important how these two groups differ because common dependencies will match the trends for overall detached cases in Figure 7.11. Overall the number of detached cases with this temperature peak is higher (715 in total) than the number of cases without it (450 in total). Similar as before, the magnetic field strength has no influence on the separation of these branches (Figure 7.14B) For all other parameters clear influences can be seen. Simulations from the group "below" tend to have larger tokamak sizes while smaller sizes are more likely to be in the "above" group (Figure 7.14A). Because the computational grids of the tokamaks are scaled proportional to the major radius, a smaller size also reduces the absolute distance between the domain boundary in the PFR and the separatrix. This means in smaller tokamaks the PFR boundary conditions are closer to the separatrix, which could be a reason for the temperature spike. Concerning the input power both groups show qualitatively the same behaviour but the distribution of the "above" cases is shifted to slightly higher powers (Figure 7.14C). Also the three particle input parameters show clearly different statistics between these two groups (Figure 7.14D, E, F). At high deuterium input far more cases belong to the "above" group while for high nitrogen gas puff most simulations are in the "below" group. Cases with a temperature peak in the PFR also tend to have both higher particle and energy cross-field transport coefficients (Figure 7.14G, H). However, not all of these correlations must stem from a causal relationship but instead can stem simply from a correlation between the input parameters in detached simulations. Nevertheless, it seems that the deciding factor for the development of this PFR temperature peak is by which mechanism the temperature detachment is reached. In cases where this is achieved predominantly through the nitrogen gas input, no peak appears. This is further substantiated by a comparison of the statistics of the plasma properties. Cases which include this temperature peak tend to have higher electron densities and lower nitrogen concentrations at the outer midplane separatrix (Figure 7.15A, B). The nitrogen concentration is defined as the sum of the density of all nitrogen particles (ions and neutrals) divided by the

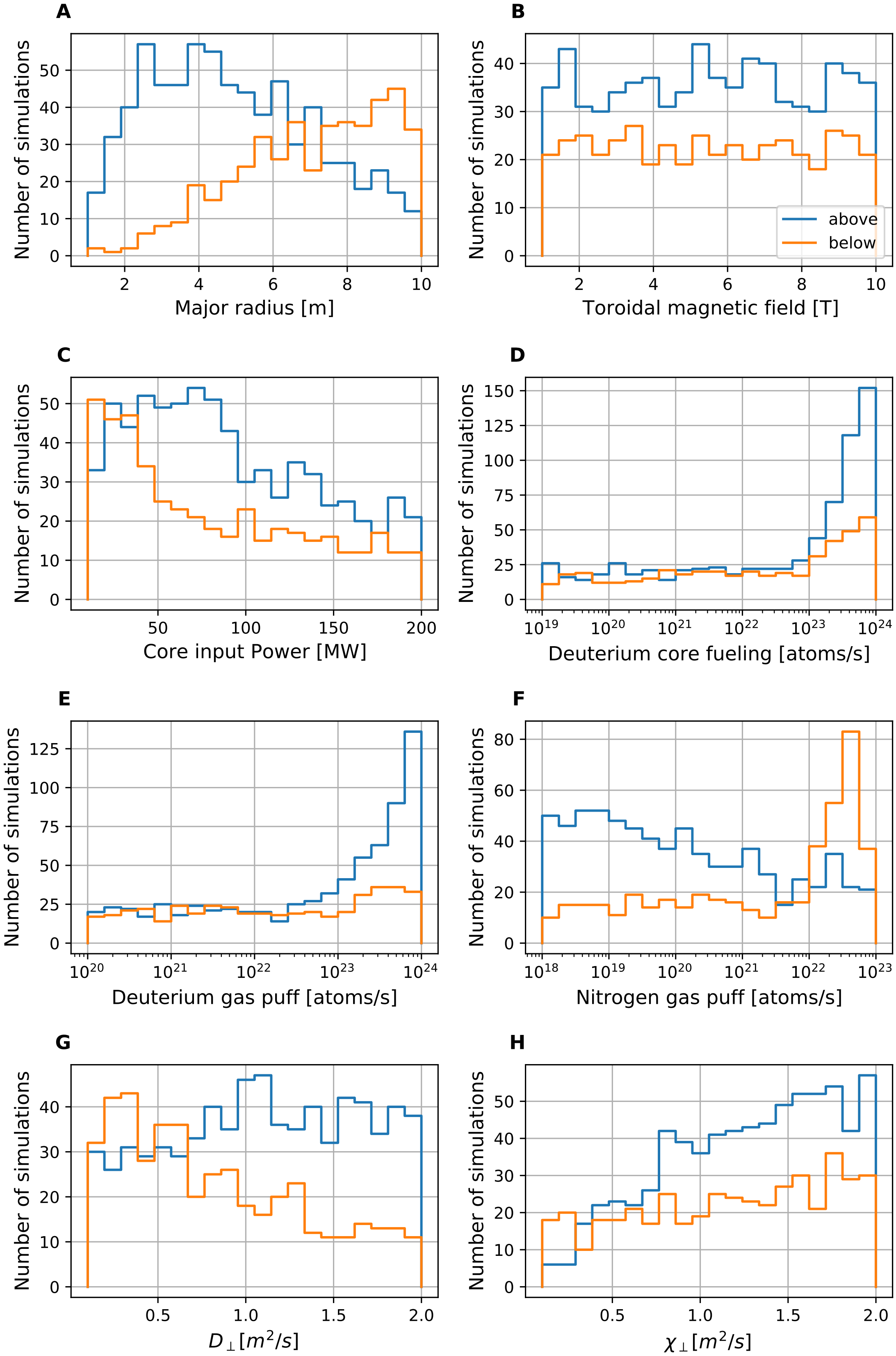


**Figure 7.14:** Histograms of the number of simulations over varying parameter values in the detached simulations from the training set. The colors denote whether the detached simulations lie above (blue) or below (orange) the separation line in Figure 7.13.

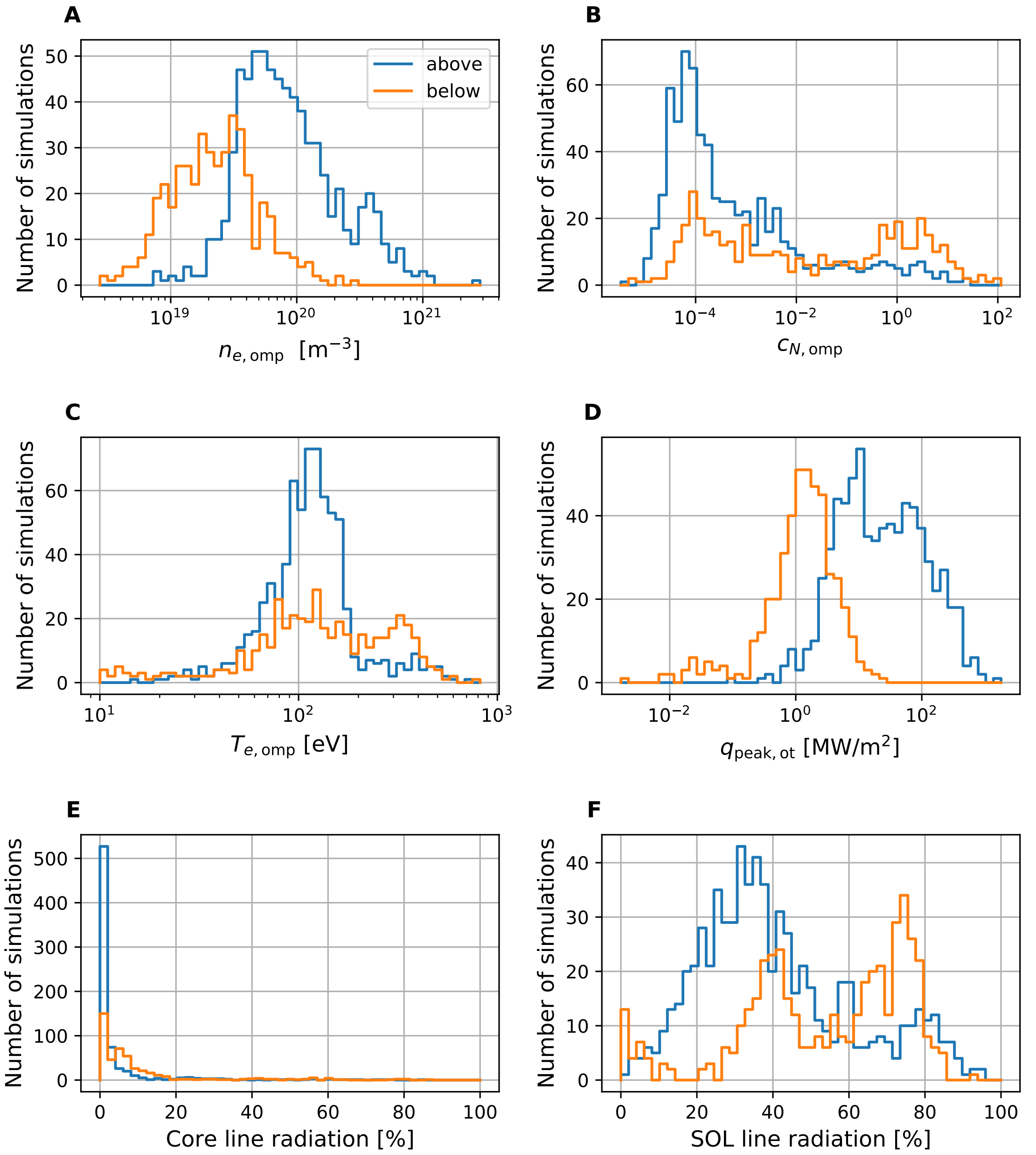


**Figure 7.15:** Histograms of the number of detached simulations over the electron density (A), impurity concentration (B) or electron temperature (C) at the outer midplane separatrix and the peak heat flux at the outer divertor target $q_{\mathrm{peak,ot}}$ (D), the line radiation in the core plasma (E) and the line radiation in the SOL (F; including divertor and PFR). The colors denote whether the detached simulations lie above (blue) or below (orange) the separation line in Figure 7.13.

| | above | below |
|---|---:|---:|
| $R$ [m] | 2.9 | 6.5 |
| $B$ [T] | 3.5 | 2.3 |
| $P_{\text{in}}$ [MW] | 3.3e+01 | 1.2e+02 |
| $D_{\text{puff}}$ [atoms/s] | 2.4e+22 | 1.5e+23 |
| $N_{\text{puff}}$ [atoms/s] | 1.1e+21 | 2.1e+22 |
| $D_{\text{core}}$ [atoms/s] | 1.6e+23 | 2.6e+23 |
| $D_{\perp}$ $[m^2/s]$ | 1.2 | 0.13 |
| $\chi_{\perp}$ $[m^2/s]$ | 1.7 | 0.97 |
| $n_{e,\text{omp}}$ [m$^{-3}$] | 7.1e+19 | 6.1e+19 |
| $c_{N,\text{omp}}$ | 0.004 | 0.01 |

**Table 7.2:** Parameters of the two simulations marked in Figure 7.13. The simulation "above" is marked by a black circle, the simulation "below" by a black square.

number of deuterium ions inside one cell of the computational grid. The distributions seen in Figure 7.15A, B are not distinct, which means additional other factors must contribute. Simulations which contain a majority of nitrogen are less relevant for reactor operation because maximum fusion power is achieved at lowest impurity concentrations [187], [200]. Although some of the simulations without the PFR temperature peak contain plasmas with a majority of nitrogen (impurity concentrations above 100%), it is important to notice that this is not exclusive (Figure 7.15B). This demonstrates that also for reactor relevant nitrogen concentrations, detached regimes are possible without exhibiting a strong PFR temperature peak. The differences in the upstream temperatures between simulations with or without this peak are small but some simulations without this peak tend to have higher temperatures at the outer midplane separatrix (Figure 7.15C). Figure 7.15D shows drastically how the peak heat flux observed at the outer target changes in the presence of the PFR temperature peak. The heat fluxes in the detached simulations without this peak are almost entirely below the critical value of 10 MW/m$^2$. In simulations with this peak the heatfluxes reach higher values. In these cases the peak heat flux occurs at the divertor target boundary in the PFR (see Figure 7.16H). Figure 7.15E, F further show that simulations with the PFR temperature peak tend to have lower line radiation both in the core as well as in the SOL and divertor areas. This seems plausible, given the lower nitrogen concentrations in these cases.

In addition to the statistical analysis two exemplary cases are compared in the following. From each group a case is selected as marked by a black circle and black square in Figure 7.13. The parameters of the two cases are shown in Table 7.2. While these cases differ in machine size and input power, the case with the PFR temperature peak has a lower nitrogen gas puff and stronger cross-field transport coefficients. The cases were selected to exhibit similar upstream densities and impurity concentrations (Table 7.2). Figure 7.16 and Figure 7.17 compare the plasma profiles and neutral densities between the two cases. The modelled extend of the SOL in the below case is larger because of the larger tokamak size (Figure 7.16A). The lower density transport coefficients in the "below" case lead to higher gradients at the outer midplane. This causes higher plasma densities inside the confined region but lower values in the SOL (Figure 7.16B). At the separatrix the conditions are almost identical in both simulations. At the target both simulations have similar temperatures in the SOL but large differences in the PFR (Figure 7.16C). In the "above" case the temperature at the target increases at the PFR domain boundary up to 50 eV while in the "below" case the temperature inside the PFR stays below 10 eV and does not peak directly at the domain boundary. The plasma densities at the target in both cases are very much alike (Figure 7.16D). Only due to the larger extend of the simulation domain in the "below" case the densities are lower at both ends of the target. The neutral densities at the target show a sharp decrease at the PFR boundary due to the pumping boundary condition (Figure 7.16E, F). In the "above" case the decrease in neutral densities is continuous in the PFR while in the "below" case the density performs a jump directly at the boundary. As expected from the gas puffs values, the "above" case has a higher deuterium neutral density (Figure 7.16E) while the "below" simulation has a higher nitrogen neutral density in the SOL (Figure 7.16F). The neutral densities for both species show a sharp increase at the outer side of the outer target because this is adjacent to the far-SOL boundary where the gas puffs enter the domain. Outside of the separatrix, both the deuterium ion flux as well as the heat flux to the target are almost identical (Figure 7.16G, H). Only inside the separatrix the ion flux in the "above" case stays high, and the heat flux increases close to the domain boundary in the PFR. So while both cases have a similar heat flux outside of the separatrix, the peak value is far greater in the "above" case.

Figure 7.17 compares the simulations in the whole divertor area. In the "above" case the electron temperature profile is much broader and extends more into the PFR (Figure 7.17A). In the "below" case the plasma profile is much more constrained closer to the separatrix both inside the PFR and the SOL (Figure 7.17B). The cause of this could lie in the drastically different cross-field transport strengths, because these settings also effect the cross-field transport inside the PFR. While both cases have a separatrix temperature below 5 eV at the outer target, a large part of this temperature decreases is only achieved directly at the target, where the neutral density is increased due to the particle recycling. At the inner target a stronger detachment is seen where the temperature reaches values below 1 eV at a greater distance to the target (Figure 7.17A, B). Although the deuterium neutral density at the targets and the far-SOL domain boundary is higher in the "above" case (Figure 7.17C), in the "below" simulation it extends further towards the separatrix especially at the outer divertor side (Figure 7.17D). While both cases have similar nitrogen densities directly at both targets, the "below" case has a way higher nitrogen density in the PFR (Figure 7.17E, F). The line radiation in the "below" case is strongly increased in the whole plasma in comparison to the "above" case (Figure 7.17G, H). The radiation is highest at the divertor targets and follows the contours of the neutral densities. Such radiation patterns are common in fluid neutral simulations [182], [183].

To summarize this analysis: It is not entirely clear what the root causes of the high temperature tail in some simulations are. It has been seen that this occurs predominantly for smaller tokamak sizes with high densities and high cross-field transport. The influence of the tokamak size could be linked to the employed size

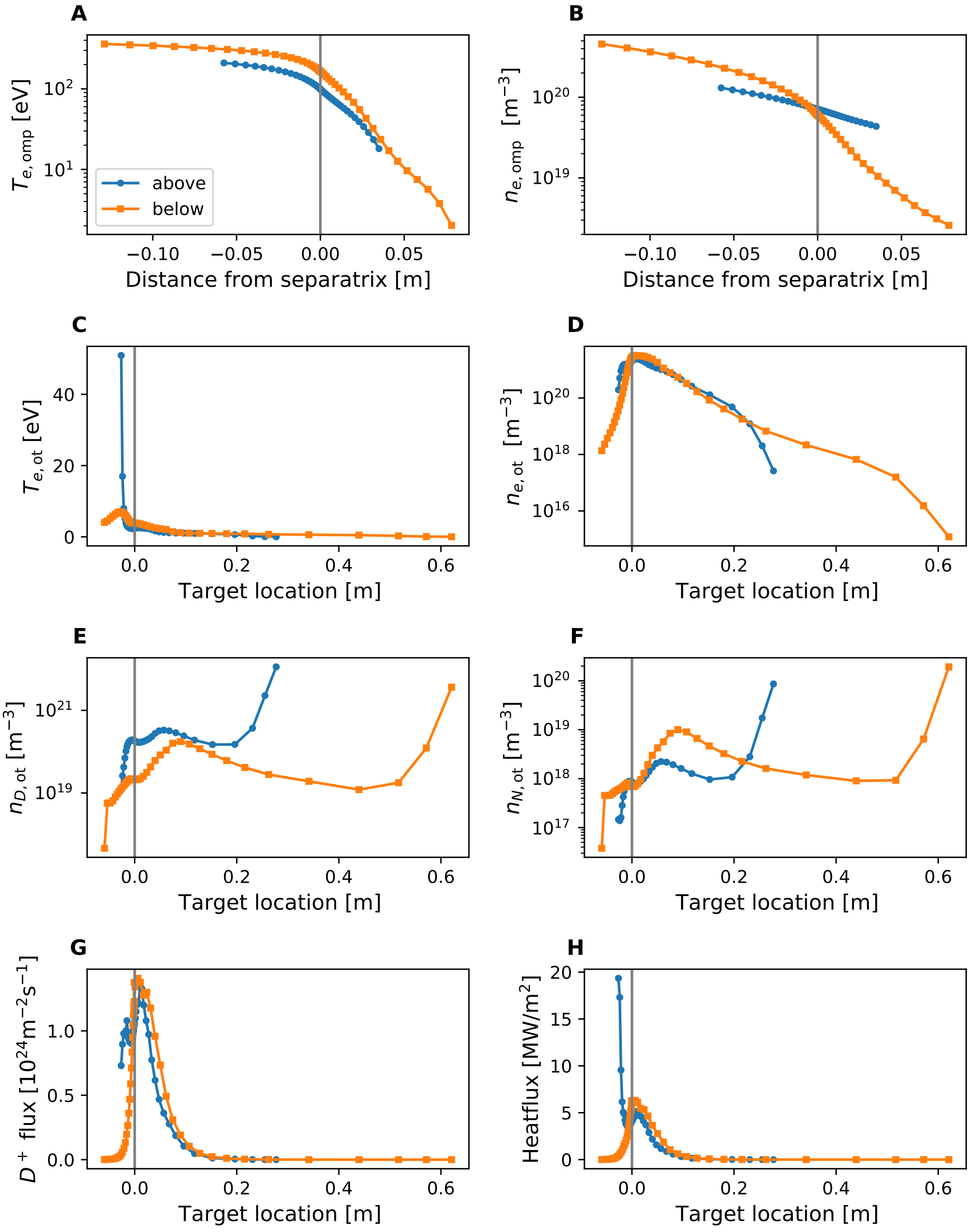


**Figure 7.16:** Plasma and neutral profiles at the outer midplane and outer target in the simulation ”above” (blue) and ”below” (orange) from Table 7.2. At the outer midplane, the electron temperatures (A) and densities (B) are shown. At the outer target, the electron temperatures (C) and densities (D), the deuterium (E) and nitrogen (F) neutral densities and the deuterium ion flux (G) and heat flux (H) are shown. Because the simulation domain is scaled proportionally with the tokamak major radius, the radial extend of the profiles differ between the two simulations.

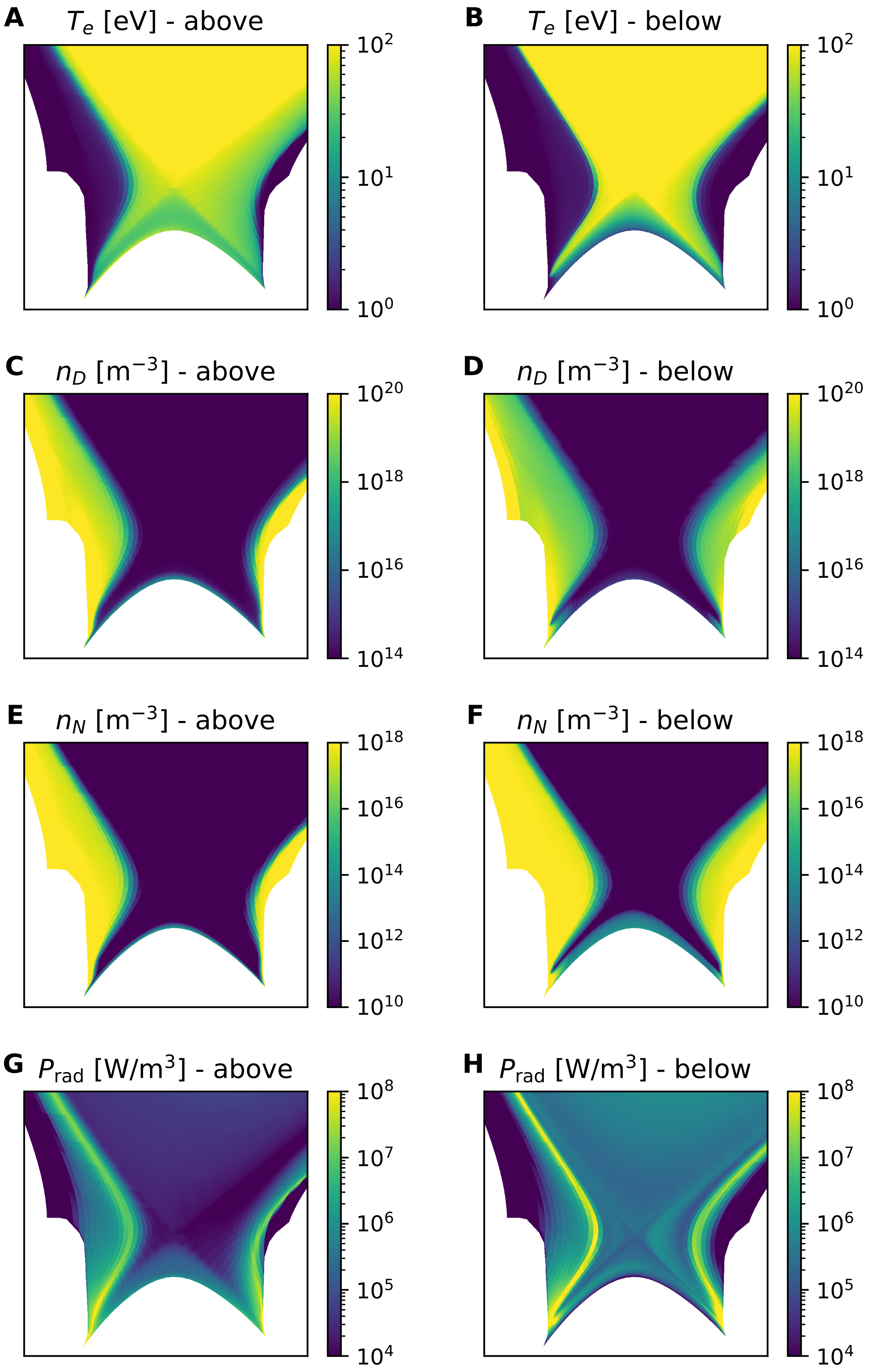


**Figure 7.17:** 2D profiles of the electron temperatures (A, B), neutral deuterium density (C, D), neutral nitrogen density (E, F) and line radiation rate (G, H) in the example simulations "above" (A, C, E, G) and "below" (B, D, F, H) from Table 7.2.

scaling, which decreases the width of the SOL and PFR with decreasing major radius. Then in small tokamaks the pumping boundary condition in the PFR might be too close to the separatrix. Another key factor seems to be the nitrogen density in the PFR. For higher nitrogen densities the temperature peak in the PFR vanishes. But all these aspects are linked to the implementation of the reduced fidelity fluid neutral model and consequently the implementation of the gas puffs. A different neutral treatment or inserting the gas puff in the PFR instead of at the far-SOL boundary might totally eradicate the correlation observed with the major radius. Studies that use a similar fluid neutral implementation as here do not report the shape of target temperature profiles [180], [183] or contain no gas puff and pump [201], [202]. This prevents a comparison to determine whether this effect is unique to the setup used here or generally a property arising in the neutral model. It has to be noted that the neutral density distributions seen in Figure 7.17 are drastically different than what is seen in simulations with kinetic neutrals or more advanced fluid neutral implementations [184]. A more extensive study of these effects is likely not worthwhile because the effects might vanish with more exact neutral representations. Although it is not shown here, it was also tested whether changes to the strength of the pump in the private-flux region mitigate the effect but the results remain inconclusive. The key requirement for this study is that the employed simulations are able to achieve reasonable plasma profiles in all regimes and conditions. The simulation database fulfills this requirement. In relevant plasma scenarios, detachment can be achieved both with as without this high temperature tail (Figures 7.13,7.14, 7.15) and the overall statistics of the regimes follow the expected trends (Figure 7.10).

# Chapter 8

# Surrogate modeling

Finally this chapter uses the dataset from the previous chapter to develop surrogate models. Because the main goal of this thesis is to compare different methods to provide recommendations for future studies, many different models with different architectures and scopes are created and compared. Section 8.1 explains the training and optimization procedure that is used for model development in the following sections. In Section 8.2 different models are developed and evaluated that predict the electron temperatures at the 1D surface of the outer target. This is followed in Section 8.3 by models that are capable of predicting the electron temperatures in the whole 2D domain. Section 8.4 analyses how the accuracy of these surrogate models depends on the number of training simulations that are available. Different modeling approaches to include more plasma properties than just the electron temperature are compared in Section 8.5. And lastly Section 8.6 compares strategies to deduce additional derived quantities from the model predictions.

## 8.1 Workflow for model training and optimization

As explained in Section 3.1.5, for training a machine learning models and especially neural networks it is helpful if the data is scaled to values close to zero. For the eight simulation parameters this is done by applying a standardization as defined in Section 3.1.5 to each of the parameters. Because the parameters $D_{\text{puff}}$, $N_{\text{puff}}$ and $D_{\text{core}}$ are varied on a logarithmic scale, the standardization is applied on the logarithm (base 10) of these values. The standardized simulation parameters then constitute the input to the machine learning models.

The pre-processing of the plasma states requires more effort. Two factors have to be considered: the large spatial variation of the plasma properties in the domain and the large variation between different simulations. Figure 8.1A depicts the temperatures in the simulation domain averaged across all simulations in the training set. It demonstrates (which is also known a priori) that the average temperature inside the confined core plasma is several orders of magnitude larger than in the SOL and divertor. Figure 8.1B further demonstrates that the statistics of temperatures occuring at the outer midplane across simulations differ drastically from the statistics of the temperatures at the outer target. Because of this, we expect a surrogate model to achieve higher accuracy when the plasma properties are scaled independently for each location. Fundamentally the machine learning model should predict whether the temperature at some location (e.g. the divertor) is high or low in comparison to the temperature range that is expected at that location. Therefore, this position dependent scaling takes some of the challenge away from the machine learning model. If instead an identical scaling were used across the simulation domain, the machine learning model would spend a lot more effort to first learn that temperatures at the divertor are in most cases cooler than in the core plasma. But also at a fixed location the temperatures across the different training simulations vary over more than 10 orders of magnitude (Figure 8.1B). This clearly advocates the use of a non-linear function to reduce this range such as applying a logarithm. Instead a quantile transformation (Section 3.1.5) is used here, because with this transformation the parts of the temperature scale with the most simulations occupy a larger fraction of the scaled temperature range. Because the number of simulations in the reasonable temperature range is large than the number of simulations with extreme temperatures (Figure 8.1B), this approach should benefit the surrogate accuracy in the more relevant range. Because statistics for the quantile transformation are calculated for each location in the simulation domain independently, the scaled temperatures at each location resemble a standard normal distribution (Figure 8.1C). To reverse these back to the original temperatures, of course for each location in the domain a different inverse function is used.

Some preliminary tests (not shown here) hinted that indeed this approach increases the model accuracies compared to other ways of pre-processing, but more rigorous comparison studies are necessary to determine whether it is the ideal approach. Nevertheless, this approach is applied in all consecutive model developments throughout this thesis (except for the tests conducted in Appendix A.4).

As introduced in Section 3.1.4, machine learning models contain many hyperparameters that are not learned during training but are prescribed by the researcher. Because the choice of hyperparameters can influence the performance of the model drastically, it is worthwhile to optimize the hyperparameters by testing and comparing many different configurations. To perform such optimizations during the model development, first the general procedure of this optimization process needs to be decided.

The first major decision is: Which metric is used to evaluate and rank the different model implementations? Preliminary tests showed that for the tasks here the median absolute error is most suited. But one can also understand intuitively why the median error is more suitable than the mean error. When using the models in practice, a user is most commonly not interested in the mean error because this measures the average error experienced when performing multiple predictions and adding all errors up. Instead a user typically performs a single prediction and wants to know how accurate that one is, in other words, which error can be commonly expected. This is quantified by the median error which provides the information that in 50% of the predictions the error will be lower. Additionally, the mean is more sensitive to outliers such that singular bad predictions outweigh many accurate ones. Similarly, the absolute error is less influenced by extreme values than the squared error. Because the plasma observables seen in the simulations stretch across vastly different scales (Figure 8.1), it is advisable to measure the absolute error.

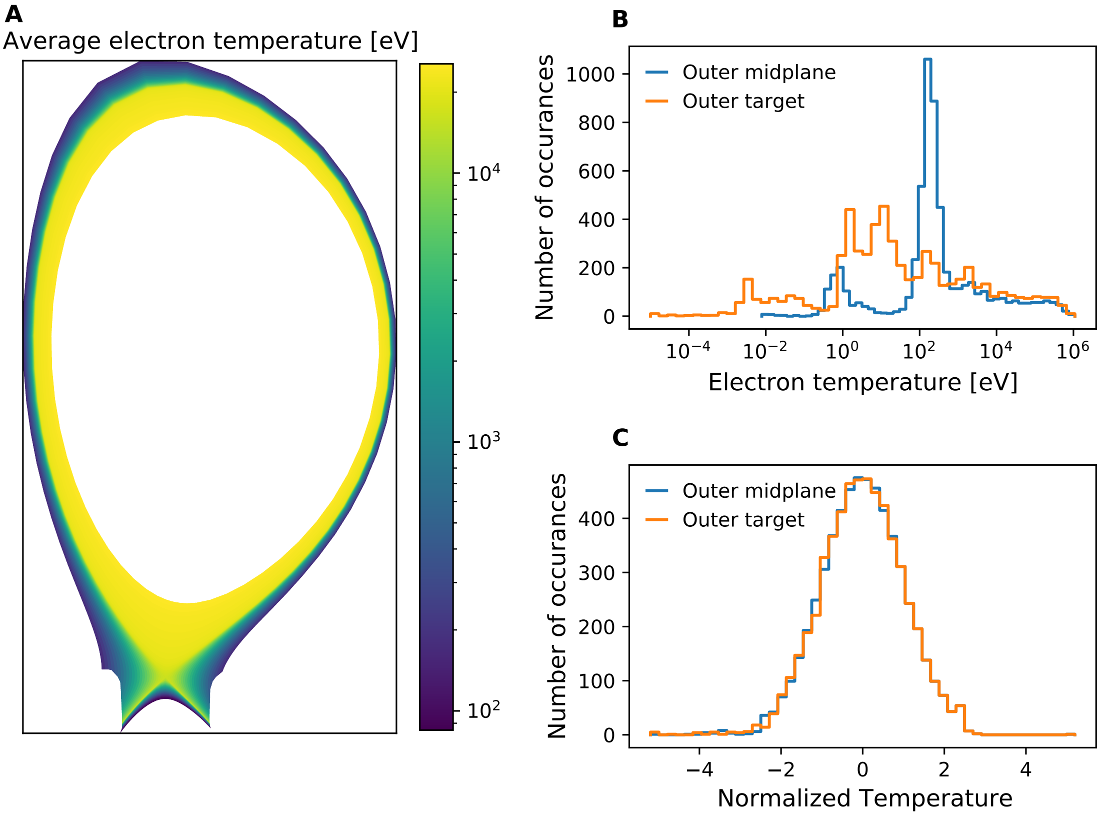


**Figure 8.1:** A: Electron temperature at each location in the simulation domain averaged over all simulations in the training set. B: Histogram of the electron temperatures at the outer midplane (blue) and outer target (orange) separatrix in the training simulations. C: Histogram of the electron temperatures at the outer midplane (blue) and outer target (orange) separatrix in the training simulations after applying the quantile transformation.

To achieve best performance all of the available data should be used to train the model, which requires avoidance of a designated holdout validation set. A common way to avoid using a holdout validation set is using cross-validation (Section 3.1.4). The cross-validation score (CV-score) that is used to compare different hyperparameter configurations is thus the median absolute error on the validation sets averaged over all 5 folds in the cross-validation. Since our primary goal is to find models that provide accurate predictions on the actual physical scales, errors are always assessed after back-transforming the model predictions to the original scales using the quantile transformation.

A crucial question when using cross-validation to optimize a models hyperparameters is how to construct the final model with the best hyperparameters. In the following sections, the final models based on gradient boosting are retrained with the best hyperparameters using all training simulations. For all neural network based models, the final models are constructed by averaging the predictions of the 5 networks trained on the different folds of the training data. This approach is taken for the neural networks because during the hyperparameter optimization early stopping is used on the validation sets in each fold. This leaves it ambiguous for how many iterations a final model an all training data needs to be trained to achieve similar accuracy. In preliminary tests this approach provided more accurate models than replacing early stopping by other forms of strong regularization like dropout. These preliminary tests also found the mean absolute error (MAE) as optimal loss function for training the neural networks. Therefore, this loss is used for all neural networks in the following sections.

## 8.2 Modeling the 1D divertor target

In theory it should be possible to train a surrogate model that predicts all properties of the SOL plasma at once. However, in practice it is unclear a priori which model architecture to choose and whether the highest levels of accuracy can be reached for this complicated task. Therefore, we start by solving simpler problems to get some intuition and as reference to see if in practice a model for ”everything” is similarly good as models for specific aspects. In this section models are trained to predict (only) the electron temperature profiles at the outer divertor target (the last row of cells in the computational grid). For this task, two basic model concepts are compared: Fully connected feedforward neural networks (NN, Section 3.2) and Gradient Boosted Regression Trees (GBRT, Section 3.3). For each of the two model concepts, two different variants are trained. In the first variant the models receive the eight parameters of each simulation as input and are trained to output the whole temperature profile at once. In the case of the neural network this requires an output layer with 48 neurons (equal to the number of target grid cells in the simulation) For the GBRT this requires training 48 models in parallel because each outputs only a scalar quantity. The trained models thus represent functions of the form $f : \mathbb{R}^8 \to \mathbb{R}^{48}$. These models will be called NN1D and GBRT in the following. In the second variant, which will be called ”positional”, the models have only a single scalar output which predicts the temperature in a predefined grid cell. The models receive one additional input to the eight simulation parameters, which specifies the index of the respective grid cell (1-48). The trained models thus represent functions of the form $f : \mathbb{R}^9 \to \mathbb{R}^1$. These model will be called NNpos1D and GBRTpos in the following. In the case of the NNpos1D, this network architecture is similar as used in Physics Informed Neural Networks [203]. The idea behind the positional variant is that the models might use the available training data more efficiently if they are explicitly forced to include the location in the domain rather than treating each location independently. Instead of seeing 5756 training samples, which each have an output dimensionality of 48, these models see $5756 \times 48 = 276288$ samples of output dimensionality 1.

To optimize the hyperparameters of each model, a random search with the 5-fold cross-validation procedure explained in Section 8.1 is applied. Random search is not the necessarily the fastest way to find the best possible model, but it is reliable in

| | GBRT/GBRTpos |
|---|---|
| Trials | 160 |
| Tree depth | [5,20] |
| Learning rate | $[10^{-3}, 10^{-1}]$ |
| Number of Estimators | [200,1000] |
| Loss function | {MAE,MSE} |
| Subsample | [0.2,1.0] |

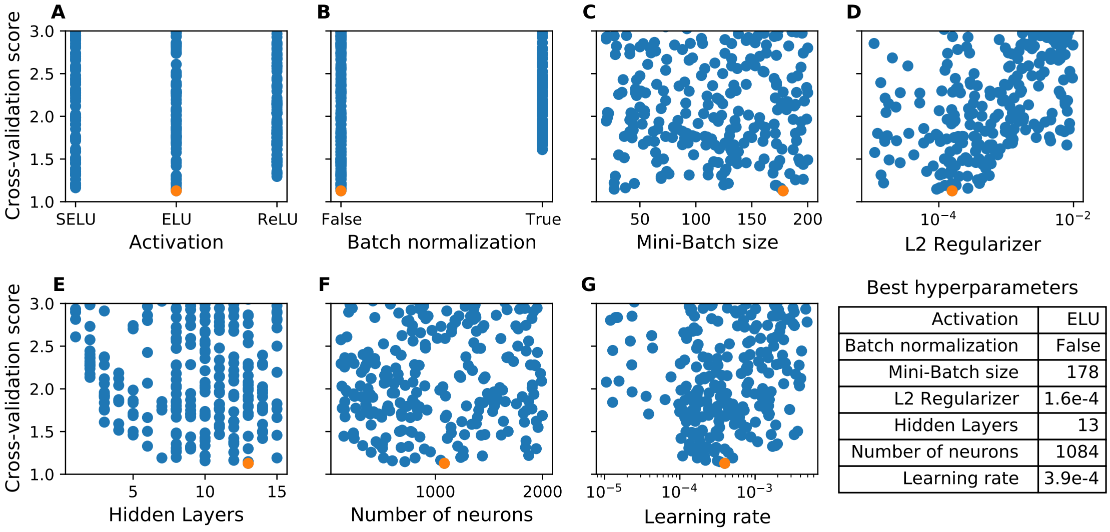
**Table 8.1:** Hyperparameters varied in the hyperparameter searches for the GBRT and GBRTpos models. [...] denote ranges in which parameters are varied, while {...} are discrete sets. The learning rate is sampled in logarithmic domain. All other model hyperparameters remain at the default values set in scikit-learn.

| | NN1D | NNpos1D |
|---|---|---|
| Trials | 160 | 54 |
| Number of layers | [1,15] | [5,15] |
| Neurons per layer | [100,2000] | [100,2000] |
| Learning rate | $[10^{-5}, 5 \cdot 10^{-3}]$ | $[10^{-4}, 5 \cdot 10^{-3}]$ |
| L2 regularization | $[10^{-5}, 10^{-2}]$ | $[10^{-5}, 10^{-2}]$ |
| Activation | {ReLU,ELU,SELU} | {ReLU,ELU,SELU} |
| Mini-Batch size | [20,200] | [20,200] |
| Batch normalization | {True,False} | {True,False} |
| Loss function | MAE | MAE |
| Patience [Epochs] | 50 | 50 |

**Table 8.2:** Hyperparameters varied in the hyperparameter searches for the NN1D and NNpos1D models. [...] denote ranges in which parameters are varied, while {...} are discrete sets. The learning rate and L2 regularization were sampled in logarithmic domain. Batch normalization was only tested with ReLU and ELU activations. Each hidden layer has the same number of neurons. All other model hyperparameters remain at the default values set in TensorFlow.

finding a good model and allows us to gain an intuition about suitable hyperparameter ranges. Table 8.1 lists the varied hyperparameters and their maximum range for both GBRT and GBRTpos, while Table 8.2 lists the same for NN1D and NNpos1D.

To assess the viability of a hyperparameter configuration the median absolute validation error (averaged over the folds after backtransformation into eV-scale, see Section 8.1) is used as cross-validation score. The best hyperparameter configurations are those which achieve the lowest cross-validation score. Figure 8.2 depicts the random search cross-validation scores for the NN1D model. Each point represents one tested configuration. Because the random search algorithm does not optimize towards a specifically suitable architecture but tries random combinations of hyperparameter values, many bad performing configurations with high scores are tested. Of relevance are only the hyperparameters that lead to the best scores. The Mini-Batch size has apparently no impact on the model accuracy (Figure 8.2C) and can therefore be set to higher values for training efficiency. For both the number of hidden layers and the number of neurons, lower limits of 7 hidden layers (Figure 8.2E) with each 500 neurons (Figure 8.2F) exist. With smaller networks the error increases while larger networks have no added benefit. If the searched hyperparameter space would be extended to include even higher number of layers and neurons, it is likely that at some point the model error starts increasing again. Batch normalization harms the model accuracy and should not be used (Figure 8.2B). Similarly also the self-normalizing properties of the SELU activations offer no added benefit beyond the plain ELU activation functions. However, both are slightly better than ReLU activations (Figure 8.2A). The networks achieve more accurate predictions when the regularization is minimal, which is the case for small values of the L2 regularizer (Figure 8.2D) and rather small learning rates between $1 \cdot 10^{-4}$ and $6 \cdot 10^{-4}$ (Figure 8.2G).



Best hyperparameters

| | |
|---|---|
| Activation | ELU |
| Batch normalization | False |
| Mini-Batch size | 178 |
| L2 Regularizer | 1.6e-4 |
| Hidden Layers | 13 |
| Number of neurons | 1084 |
| Learning rate | 3.9e-4 |

**Figure 8.2:** Cross-validation score of the NN1D model architecture with varying hyperparameters. Each dot represents one tested hyperparameter configuration in the random search. The best hyperparameter configuration is the one with the lowest score (marked in orange). The corresponding hyperparameter values are given in the attached table. Several configurations exhibit higher cross-validation scores than visible in these plots.

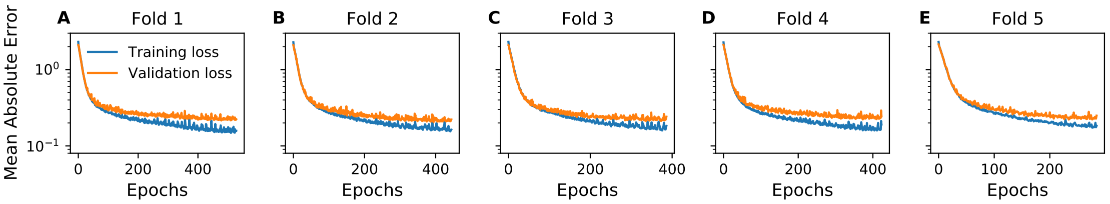
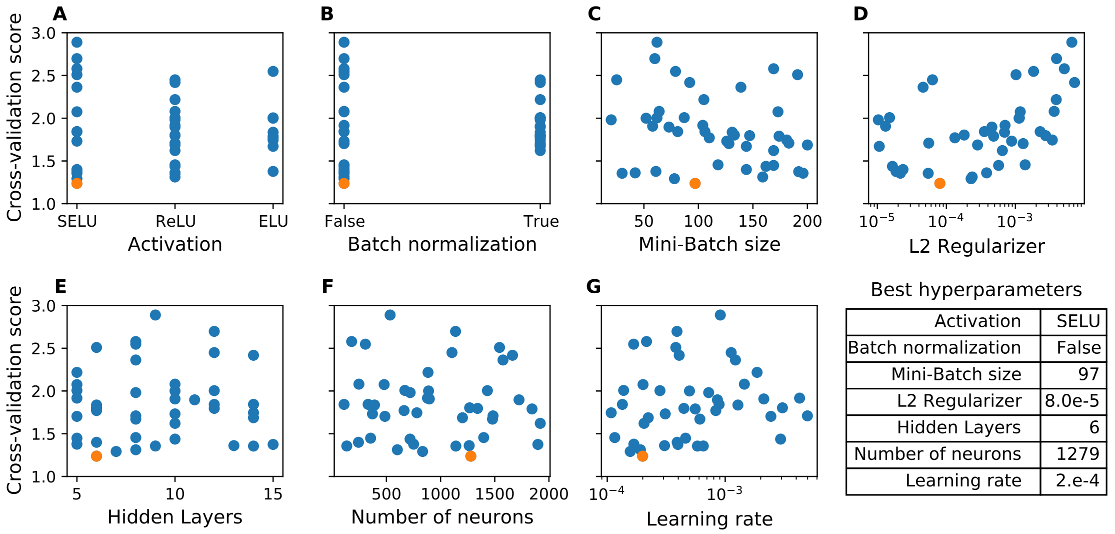


**Figure 8.3:** The MAE loss on the training (blue) and validation (orange) data in each of the five folds as function of the epochs during training of the best NN1D model.



Best hyperparameters

| | |
|---|---|
| Activation | SELU |
| Batch normalization | False |
| Mini-Batch size | 97 |
| L2 Regularizer | 8.0e-5 |
| Hidden Layers | 6 |
| Number of neurons | 1279 |
| Learning rate | 2.e-4 |

**Figure 8.4:** Cross-validation score of the NNpos1D model architecture with varying hyperparameters. Each dot represents one tested hyperparameter configuration in the random search. The best hyperparameter configuration is the one with the lowest score (marked in orange). The corresponding hyperparameter values are given in the attached table. Several configurations exhibit higher cross-validation scores than visible in these plots.

Figure 8.3 depicts the progression of the MAE loss during training on the scaled data for the best model found in the hyperparameter search. The gap between the validation and training loss indicates that the low L2 regularization value is not sufficient on its own to prevent overfitting, but the additional early stopping is required. The number of training epochs before stopping are different in each fold, but all are of the same magnitude ($300 - 600$ epochs).

Because the training data for the positional models is formatted such that each scalar temperature value is seen by the models as independent sample, a single epoch consists of many more batches than before. For the NNpos1D model this means a higher number of weight updates and less parallel computations, which results in longer training times. To find suitable hyperparameter configurations with testing less trials, the search space was decreased in size based on the limits that can be expected from the results of the NN1D search (Table 8.2).

The best hyperparameter configurations in the NNpos1D model are very similar as in the NN1D model before. Also for the NNpos1D model, the Mini-Batch size has, inside the tested range, no influence on the accuracy (Figure 8.4C) and can be set to the larger values. Again Batch normalization should be avoided (Figure 8.4B). For the activation functions, SELU performs the best although the gap to ReLU and ELU is small (Figure 8.4A). Inside the tested ranges, both the number of neurons and the number of hidden layers have little influence on achieving optimal performance (Figure 8.4E, F). The strongest impact comes from the L2 regularizer, which should be set to values around $10^{-4}$ (Figure 8.4D) with a similar value for the learning rate (Figure 8.4G). Due to the low number of trials in the hyperparameter search, the model error could likely be still improved by performing more trials or

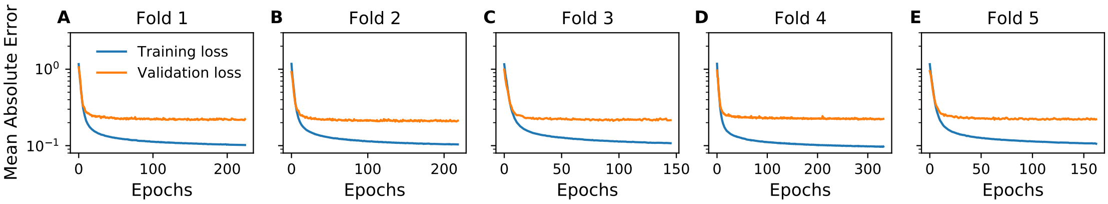
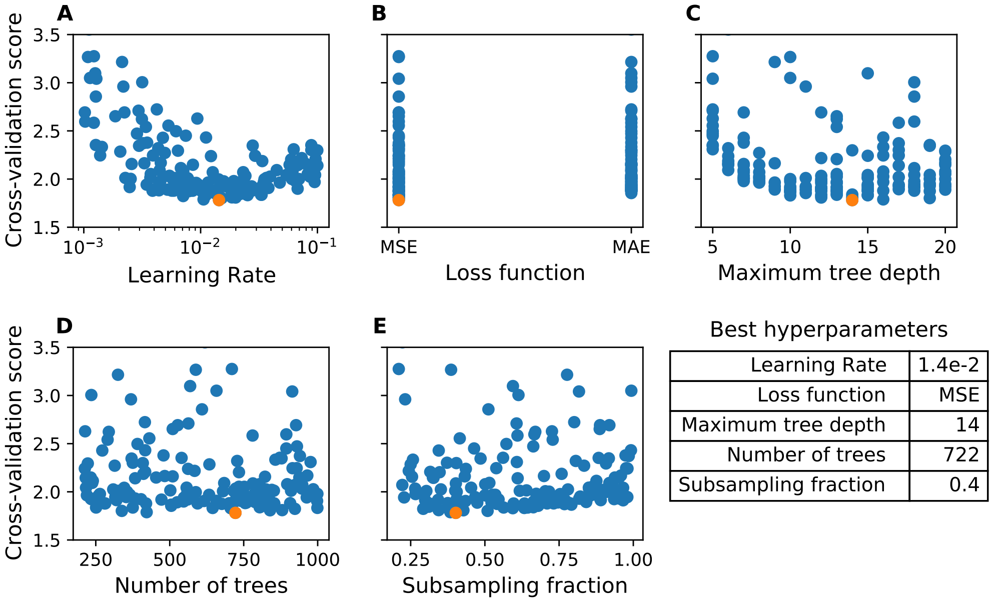


**Figure 8.5:** The MAE loss on the training (blue) and validation (orange) data in each of the five folds as function of the epochs during training of the best NNpos1D model.



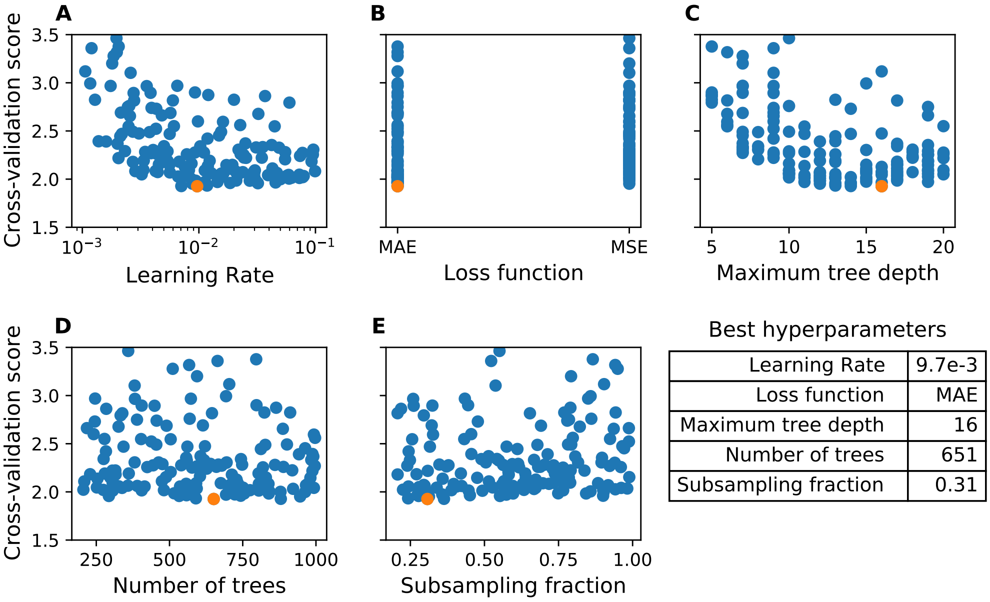
| | |
|---|---|
| Learning Rate | 1.4e-2 |
| Loss function | MSE |
| Maximum tree depth | 14 |
| Number of trees | 722 |
| Subsampling fraction | 0.4 |

**Figure 8.6:** Cross-validation score of the GBRT model architecture with varying hyperparameters. Each dot represents one tested hyperparameter configuration in the random search. The best hyperparameter configuration is the one with the lowest score (marked in orange). The corresponding hyperparameter values are given in the attached table. Several configurations exhibit higher cross-validation scores than visible in these plots.

extending the tested hyperparameter ranges especially concerning the number of neurons and hidden layers.

Because the patience value for training the NNpos1D model is the same as for the NN1D, this results with similar Mini-Batch sizes to many more weight updates in the NNpos1D model. This might be the reason why the difference between training and validation loss for the best NNpos1D model (Figure 8.5) is much larger than for the NN1D model. For the same reason also the total number of training epochs is smaller (150-400).

Both gradient boosting architectures (GBRT and GBRTpos) show almost identical influences of the hyperparameters (Figures 8.6 and 8.7). The strongest impacts have the learning rate and the maximally allowed tree depth (Figures 8.6A, C and 8.7A, C). Both models achieve best performance at learning rates around 0.01 and a maximum tree depth of 14 – 16. The number of trees in the boosting process seems to have no effect (Figures 8.6D and 8.7D), which hints that less than 200



| | |
|---|---|
| Learning Rate | 9.7e-3 |
| Loss function | MAE |
| Maximum tree depth | 16 |
| Number of trees | 651 |
| Subsampling fraction | 0.31 |

**Figure 8.7:** Cross-validation score of the GBRTpos model architecture with varying hyperparameters. Each dot represents one tested hyperparameter configuration in the random search. The best hyperparameter configuration is the one with the lowest score (marked in orange). The corresponding hyperparameter values are given in the attached table. Several configurations exhibit higher cross-validation scores than visible in these plots.

trees should be enough already. Similarly, the subsampling fraction seems to have only a moderate impact on model accuracy. But for both models the optimum is around 0.3-0.4 (Figures 8.6E and 8.7E). Regarding the loss function used to determine the splitting criteria in the trees, both least squares and least absolute error perform equally well (Figures 8.6B and 8.7B).

While the tables in Figures 8.2-8.7 give the best found hyperparameter configurations for each model, these values lie in ranges where many similar configurations perform almost equally well. Therefore, these configurations should not be seen as the sole optimum, but the analysis should serve as a guideline to determine which hyperparameters are more important and which ranges of values are suitable.

To produce predictions on the separate test set using the best hyperparameter configurations, the gradient boosted models (GBRT and GBRTpos) are retrained on the entire training data. For the neural networks (NN1D and NNpos1D), the predictions of the models trained in the different folds are averaged. Figure 8.8 shows the predicted temperature profiles for eight example simulations from the test set. The predictions from all models are in most cases close to the profiles seen in the simulations. For several test simulations, the predictions of the neural network models are closer to the simulation output than the GBRT based models (Figure 8.8A, B, C). Yet in other cases also the GBRT model is more accurate (Figure 8.8E). And sometimes all models predict different profiles then seen in the simulation (Figure 8.8H). All models can generally cope with the vastly differing temperature scales and profile shapes across the different examples. The profiles from the simulations always have an inherent smoothness. This is not always recovered by the models, and the predicted profiles can have less natural shapes (Figure 8.8D, E). In these examples the positional models perform better in that regard because their profiles are often smoother. This property can be expected from these models because they have to inherently integrate the correlations between the spatial locations.

Instead of evaluating the surrogate accuracy solely on a few examples, Figure 8.9 compares the predicted temperatures at all outer target locations in all test cases against the simulations. The models are most accurate when the predicted temperatures are close to the diagonal. For all four models the predicted temperatures are highly correlated with the temperatures seen in the test simulations. For both neural network models the predictions are generally closer to the simulations than for the GBRT models. Below 1 eV the accuracy in all models seems to decrease and more model predictions divert from the values seen in the ground truth simulations. Nevertheless, also for some higher temperature cases the model predictions can be orders of magnitude too high or too low. Interestingly, all four models show similar patterns in some of their highest errors. When one searches for the widest deviations from the optimal diagonal in Figure 8.9, multiple examples can be observed, which show similar point patterns in all four subplots. This means in some cases all four models make exactly the same mistakes. One possible explanation for this effect is that there are "bad" simulations in the test set. Such could be cases which were stopped prematurely by one of convergence metrics while they were still in a transient phase (see Section 7.2). In these situations the simulations results do not correspond to the trends expected by the input parameters. When all four models are correct in their prediction and only this specific test simulation is inaccurate, this would lead to similar error patterns for the models. But this is only a hypothesis, which can not be verified. Therefore, it can not be ruled out that all four models make the same incorrect predictions in certain cases althought it seems less likely. As discussed in Section 7.4, the simulations with the most extreme temperatures are not physically reasonable or relevant for reactor design when effects outside of the SOLPS-ITER simulations are considered. Figure 8.10 shows a close up of the results in the more relevant temperature range $10^{-1} - 10^{3}$ eV. Here it becomes clearly visible that the NN1D (Figure 8.10A) and NNpos1D (Figure 8.10B) models are more accurate than the regression tree models because the predictions are more densely grouped close to the optimal diagnal.

To provide a quantitative comparison between the different models, Table 8.3 summarizes the median absolute errors and median relative errors on the test set. In both metrics the neural network models perform better than the gradient boosted models. Also the positional models are slightly worse than their regular counterparts. Therefore, the best model is the NN1D architecture with an overall median absolute error of 1 eV and relative error of 13% (Table 8.3). While these results are not perfect, it will be in most cases accurate enough for first scoping studies and to validate general trends. Because even highly tuned SOLPS-ITER simulations regularly reproduce experiments only qualitatively (see e.g. [133], [202], [204]–[206]), the discrepancy between the model and the simulations is likely smaller than the gap between the fluid neutral simulations and reality. However, this assessment is based here solely on the median errrors. No guaranteed upper limit for the model

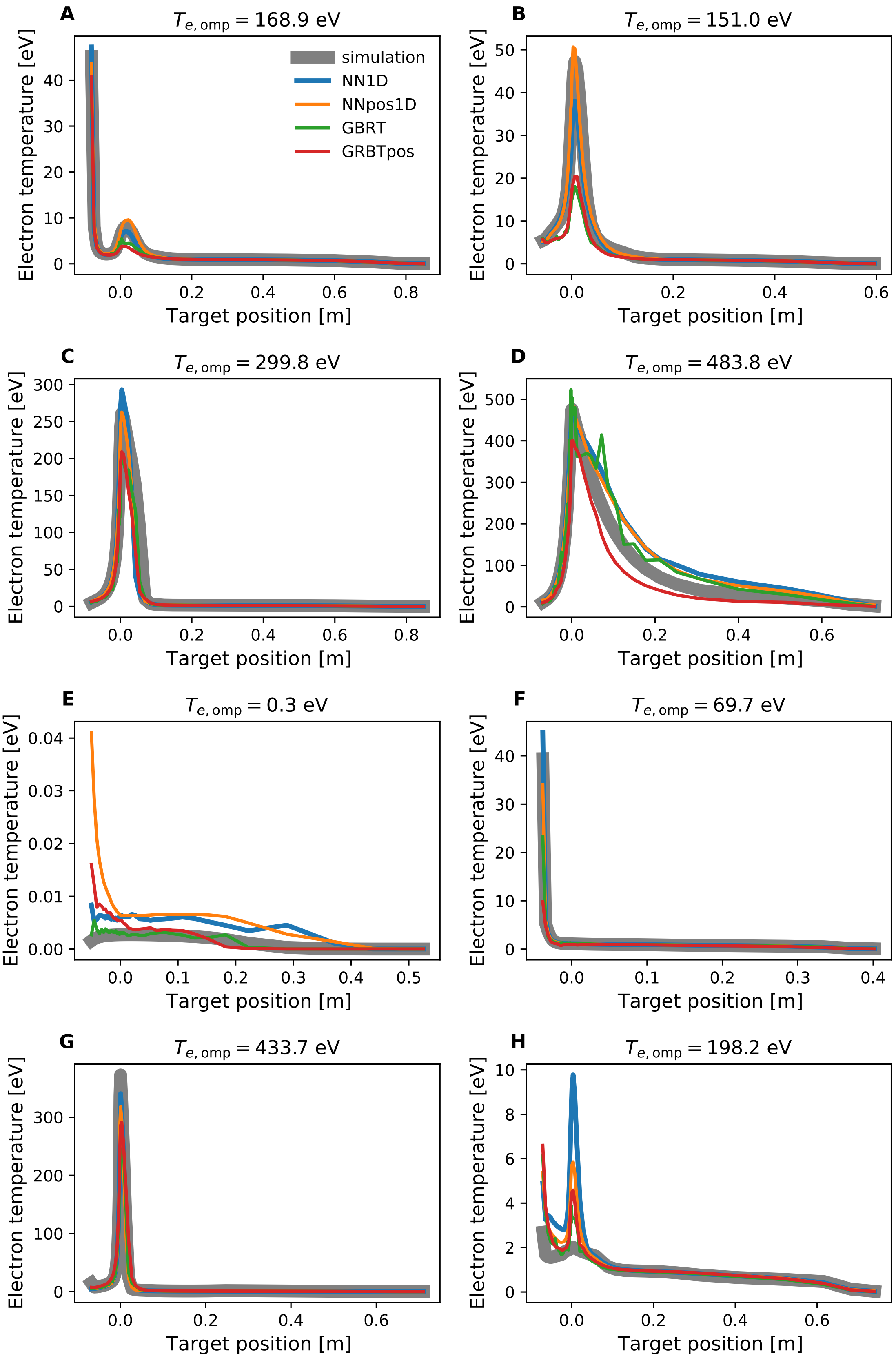


**Figure 8.8:** Comparison between the temperature profiles predicted by the various models at the outer target for eight simulations from the test set. The eight examples are hand selected to represent different regimes. For reference the temperature at the outer midplane separatrix in each simulation is given. Shown are the simulation results (grey) and the predictions by the NN1D (blue), NNpos1D (orange), GBRT (green) and GBRTpos (red) models.

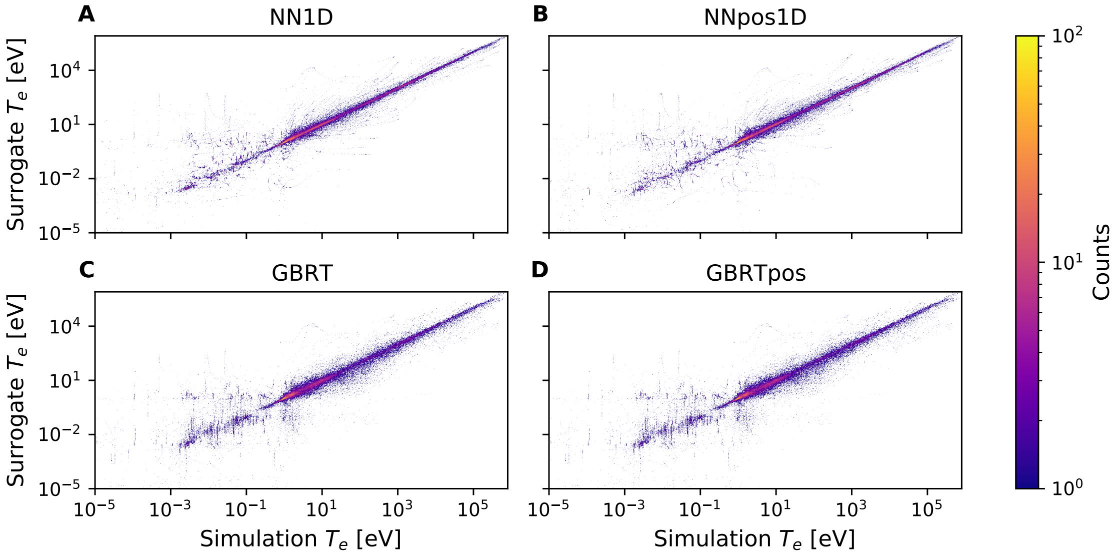


**Figure 8.9:** The model predicted temperatures in all outer target grid cells against the respective temperatures in the simulations for all simulations in the test set. The color denotes the number of overlapping points at the same plot position. A model is more precise the closer the predictions are to the diagonal.

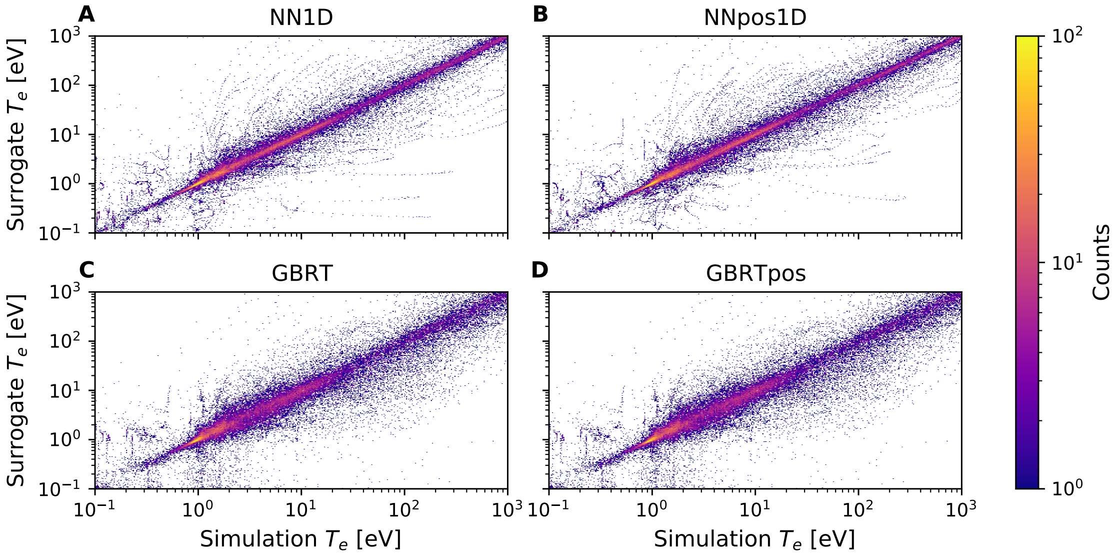


**Figure 8.10:** Close up of Figure 8.9 in the temperature range $10^{-1}$ eV – $10^{3}$ eV. Everything else is identical to Figure 8.9.

| | | all | sheath-limited | attached | detached | cold core |
|---|---|---|---|---|---|---|
| NN1D | abs | 1.0 | 1e+02 | 0.92 | 0.19 | 0.015 |
| | rel | 0.13 | 0.1 | 0.12 | 0.12 | 0.75 |
| NNpos1D | abs | 1.1 | 1e+02 | 0.93 | 0.18 | 0.014 |
| | rel | 0.14 | 0.1 | 0.12 | 0.12 | 0.74 |
| GBRT | abs | 1.6 | 2.3e+02 | 1.6 | 0.29 | 0.018 |
| | rel | 0.27 | 0.25 | 0.23 | 0.19 | 0.91 |
| GBRTpos | abs | 1.7 | 2.4e+02 | 1.8 | 0.35 | 0.026 |
| | rel | 0.3 | 0.28 | 0.26 | 0.22 | 1.0 |

**Table 8.3:** Median absolute and relative errors on the test set, with the median calculated either over the whole test set or only simulations in certain regimes (as defined in Section 7.4). The absolute errors are given in eV.

errors exists, so in rare cases the model predictions can be off by far more (Figure 8.9) or contain unphysical correlations. The data points in the test set are split into regimes based on the criteria described in Section 7.4. This way the errors of the surrogate models can be evaluated for each of these regimes independently (Table 8.3). The absolute errors scale with the temperatures present in the different regimes. So in regimes with lower temperature the absolute errors tend to be smaller while the relative errors become larger. The best median absolute error under attached conditions is with 0.92 eV only slightly better than the error over the whole test set. But the absolute errors in the detached regime are a lot smaller, with the best median values obtained by the NNpos1D model at 0.18 eV. While the cool core regime has diminishing absolute errors the relative errors explode due to the small scales. For all other three regimes, the relative errors of each model change only slightly between regimes. Apparently, it is not generally more difficult to yield accurate predictions in one regime than in another. The neural network based models (NN1D and NNpos1D) achieve very similar scores in all regimes and are always significantly better than the boosted trees (GBRT and GBRTpos). The small benefit of the NN1D model compared to the NNpos1D architecture might only stem from the higher number of trials in the hyperparameter search (Table 8.2).

While the positional network NNpos1D is similar in accuracy, it takes much longer to train than the NN1D model. The best NNpos1D model took 27 minutes to train (on average for each fold) on a single Nvidia A100 GPU, while the NN1D model took less than 2 minutes (on average for each fold). The GBRT model took 11 minutes to train on a single CPU, but since this trains 48 models independently after another this can be trivially parallelized The GBRTpos model training which cannot be parallelized took even longer 23 minutes. Overall the training times of the individual models are tiny compared to the compute time it took to generate the SOLPS database, but a faster trainable model allows for more trials in the hyperparameter optimization, which makes it easier to optimize further.

In conclusion, the results of this section can be summarized as follows: It is possible to create a surrogate with acceptable prediction accuracy for the electron temperature at the outer divertor target. In this task, fully connected neural networks are more accurate than models based on Gradient Boosted Regression Trees. Both the NN1D and the NNpos1D network architectures are equally accurate although the former are faster to train. The best neural networks use SELU or ELU neurons, no batch normalization, between 6-13 hidden layers, around 1000 neurons per hidden layers, learning rates between $1 \cdot 10^{-4} - 6 \cdot 10^{-4}$ and L2 regularizers around $10^{-4}$. The gradient boosted models are optimal at learning rates around $10^{-2}$ and maximum tree depths between $14 - 16$. In the most relevant attached and detached conditions, the best network model obtains a median relative prediction error of 12% on the independent test data. With median absolute test errors of 0.93 eV in attached and 0.18 eV in detached cases, the discrepancy between the model and the underlying simulations is probably smaller than between the simulations and actual experiments.

## 8.3 Modeling the 2D scrape-off layer

While the divertor targets are one of the most relevant areas, a full picture of the SOL is necessary to gain an understanding of the physics leading to favorable target conditions. In addition, also the plasma conditions at the separatrix need to fit confinement requirements, and different codes (e.g. ERO2.0 [86]) require the plasma conditions in the whole SOL as background for their calculations. Therefore, this section increases the problem complexity by testing models which are capable of predicting the electron temperature in whole domain that is simulated by SOLPS-ITER. SOLPS-ITER implements boundary conditions in artificial extra guard cells around the computational grid (Section 2.7). Although the temperatures in these cells have no physical meaning, the models in this section also predict the temperatures in these cells. This way the model predictions can be used as initial states for further SOLPS-ITER simulations without experiencing numerical restart artifacts (see Sections 8.6 and 9.2).

Three different model types are tested. The first model is a fully connected neural network (NN2D) which receives the eight simulation parameters as input and contains an output layer of 5200 neurons (gridsize 104x50). This way the model outputs the temperatures in the whole SOL in one pass. The second model is a positional network (NNpos2D) with scalar output which receives the R and Z coordinates of each simulation grid cell (the cell center is given by the average of the coordinates of all 4 corners) as additional input together with the eight simulation parameters. The third model is a positional Gradient Boosted Regression Tree model implemented using XGBoost (XGBoost2D). Regular GBRT based architectures are not tested because with the implementations used here (scikit-learn/XGBoost) this would result in training 5200 models independently as done before at the outer divertor target. Therefore, the individual sub-models responsible for the target prediction in the 2D model would be entirely identical to the GRBT sub-models from the 1D case and no improvement is possible. Since the 1D GBRT models offered inferior prediction accuracy (Table 8.3), it makes no sense to build a 2D model from these inferior components. For all three model types, a hyperparameter random search is run similarly as in the previous section. The details of the searches are given in Table 8.4 for the neural networks and Table 8.5 for XGBoost2D. Based on the same architecture as the NN2D model, also an additional smaller hyperparameter search is run (NN2D-small). In this search, sensible values found in the NN1D optimization (Section 8.2) are selected for most of the hyperparameters and only the number of hidden layers and neurons are varied.

For the NN2D model very similar hyperparameters are optimal as for the NN1D model before (Figure 8.11). SELU and ELU activation perform better than ReLU (Figure 8.11A), Batch normalization should be avoided (Figure 8.11B) and the Mini-Batch size has no strong impact (Figure 8.11C). For the other hyperparameters the results are similar to the NN1D model but slightly different. Best performance is reached for L2 regularizers and learning rates below $10^{-4}$ (Figure 8.11D, G), which were slightly higher in the 1D case. Since the best performing model is found at the minimum of the tested learning rates, a better model could likely be found by including even lower learning rates in the search (Figure 8.11G). The number of

| | NN2D | NNpos2D | NN2D-small |
|---|---|---|---|
| Trials | 320 | 64 | 208 |
| Number of layers | [1,15] | [8,12] | [1,15] |
| Neurons per layer | [100,1500] | [100,1500] | [100,2000] |
| Learning rate | $[10^{-4}, 5 \cdot 10^{-3}]$ | $5 \cdot 10^{-4}$ | $3 \cdot 10^{-4}$ |
| L2 regularization | $[10^{-5}, 10^{-2}]$ | $5 \cdot 10^{-5}$ | $10^{-4}$ |
| Activation | {ReLU,ELU,SELU} | ELU | ELU |
| Mini-Batch size | [20,200] | 500 | 128 |
| Batch normalization | {True,False} | False | False |
| Loss function | MAE | MAE | MAE |
| Patience | 50 | 5 | 50 |

**Table 8.4:** Hyperparameters varied in the hyperparameter searches for the NN2D, NNpos2D and NN2D-small models. [...] denote ranges in which parameters are varied, while {...} are discrete sets. The learning rate and L2 regularization were sampled in logarithmic domain. Batch normalization was only tested with ReLU and ELU activations. Each hidden layer has the same number of neurons. All other model hyperparameters remain at the default values set in TensorFlow.

| | XGBoost2D |
|---|---|
| Trials | 50 |
| Tree depth | [5,20] |
| Learning rate | $[10^{-3}, 2 \cdot 10^{-1}]$ |
| Number of Estimators | [200,1000] |
| Loss function | {MAE,MSE} |
| Subsample | [0.2,1.0] |

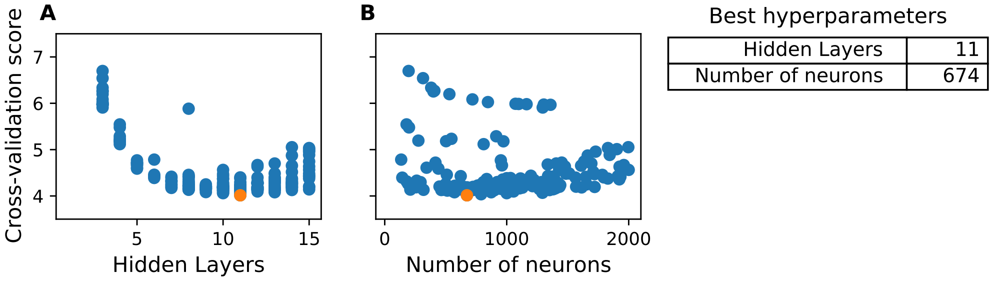
**Table 8.5:** Hyperparameters varied in the hyperparameter search for the XGBoost2D model. [...] denote ranges in which parameters are varied, while {...} are discrete sets. The learning rate is sampled in logarithmic domain. All other model hyperparameters remain at the default values set in XGBoost.

| | |
|---|---|
| Activation | SELU |
| Batch normalization | False |
| Mini-Batch size | 37 |
| L2 Regularizer | 3.4e-5 |
| Hidden Layers | 9 |
| Number of neurons | 1427 |
| Learning rate | 1.0e-4 |

**Figure 8.11:** Cross-validation score of the NN2D model architecture with varying hyperparameters. Each dot represents one tested hyperparameter configuration in the random search. The best hyperparameter configuration is the one with the lowest score (marked in orange). The corresponding hyperparameter values are given in the attached table. Several configurations exhibit higher cross-validation scores than visible in these plots.

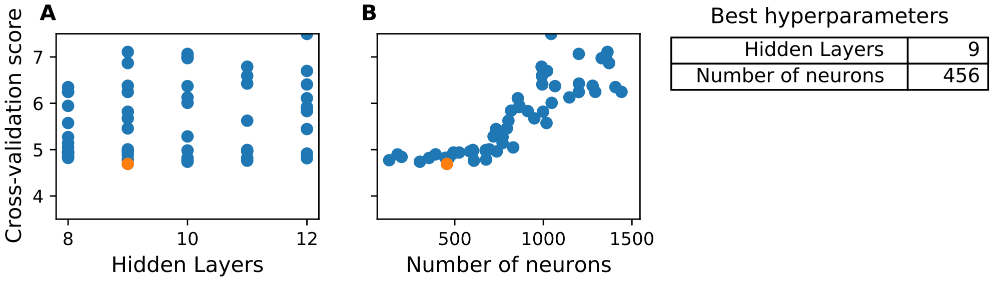
hidden layers has a clear optimum around 10 (Figure 8.11E) and while the number of neurons again has only minor effects, it appears that values between 1000-1500 neurons per layer are suitable (Figure 8.11F). The smaller hyperparameter search for the same architecture, NN2D-small, comes to similar results with an optimum of 11 hidden layers with each around 700 neurons (Figure 8.12). Due to the high number of trials with only two varied hyperparameters, this smaller search demonstrates clearly an optimal number of hidden layers around 10 (Figure 8.12A). For less than 7 hidden layers the obtained scores worsen drastically and also for 15 hidden layers the scores worsen slightly.

As explained in the previous section, the positional formatting of the training data increases the number of training samples a model sees. In the case of the 2D models, this effect is drastically increased because the $104 \times 50$ temperature values in each of the 5756 training simulations are transformed into $5756 \times 104 \times 50 = 29.9 \cdot 10^6$ independent samples. This causes drastically longer training times for the NNpos2D model. To compensate for this, the Mini-Batch size was increased and many hyperparameters were fixed to sensible values as estimated by the NN2D or NNpos1D optimization. Therefore, only the number of hidden layers and number of neurons per hidden layers are varied. Additionally the range of parameter variation was reduced to find well performing configurations with less trials in the hyperparameter optimization (Table 8.4). For the NNpos2D architecture smaller networks seem to perform better (Figure 8.13) The number of hidden layers has no effect in the tested range, which hints that less than 8 hidden layers are enough (Figure 8.13A). The number of neurons per hidden layer is much more influential. Above 800 neurons per layer the model accuracy decreases strongly (Figure 8.13B).



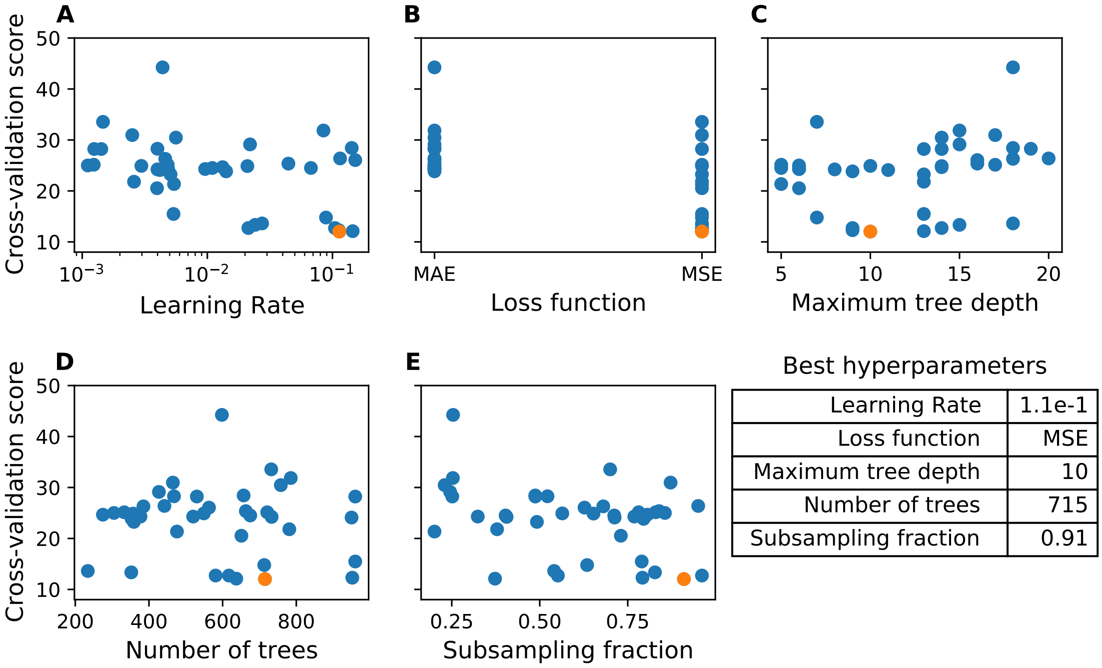
| | |
|---|---|
| Hidden Layers | 11 |
| Number of neurons | 674 |

**Figure 8.12:** Cross-validation score of the NN2D model architecture with varying hyperparameters in the small hyperparameter search (NN2D-small). Each dot represents one tested hyperparameter configuration in the random search. The best hyperparameter configuration is the one with the lowest score (marked in orange). The corresponding hyperparameter values are given in the attached table. Several configurations exhibit higher cross-validation scores than visible in these plots.



| | |
|---|---|
| Hidden Layers | 9 |
| Number of neurons | 456 |

**Figure 8.13:** Cross-validation score of the NNpos2D model architecture with varying hyperparameters. Each dot represents one tested hyperparameter configuration in the random search. The best hyperparameter configuration is the one with the lowest score (marked in orange). The corresponding hyperparameter values are given in the attached table. Several configurations exhibit higher cross-validation scores than visible in these plots.



Best hyperparameters

| | |
|---|---|
| Learning Rate | 1.1e-1 |
| Loss function | MSE |
| Maximum tree depth | 10 |
| Number of trees | 715 |
| Subsampling fraction | 0.91 |

**Figure 8.14:** Cross-validation score of the XGBoost2D model architecture with varying hyperparameters. Each dot represents one tested hyperparameter configuration in the random search. The best hyperparameter configuration is the one with the lowest score. Several configurations exhibit higher cross-validation scores than visible in these plots.

While the learning rate and maximum tree depth were the most important hyperparameters for the 1D gradient boosting models, they are less relevant for the XGBoost2D model (Figure 8.14A, C). Potentially the higher learning rates around $10^{-1}$ perform better, but this can not be determined without doubt given the few trials in the search (Figure 8.14A). While the GBRT and GBRTpos models worked with both loss functions, the XGBoost2D model achieves better CV scores with the squared loss (Figure 8.14B). This difference might stem from the differences in splitting algorithms used in scikit-learn and XGBoost. There is not visible trend or optimum regarding the total number of trees or the subsampling fraction (Figure 8.14D, E). It is likely that the subsampling fraction has no influence on the model accuracy, as was found for the 1D gradient boosting models. Since the NNpos2D model also favored a smaller number of neurons in comparison to the NNpos1D model, the XGBoost2D model might also favor smaller sizes such that the optimum is below the tested range of 200 trees. Overall the best XGBoost2D model achieves only a CV score which is more than twice the best neural network scores. Therefore, it is already apparent that the XGBoost2D model offers inferior accuracy compared to the neural networks. The best configurations for each of the architectures are now used to make predictions on unseen examples from the test set.

Figure 8.15 shows the predicted temperatures by the various models compared to the simulation for one exemplary case from the test set. All models yield predictions which resemble the simulation data and by eye it is difficult to find any mistakes. In this example, the simulation contains low temperatures at the far-SOL boundary above the inner target. This low temperature zone extends much further upstream

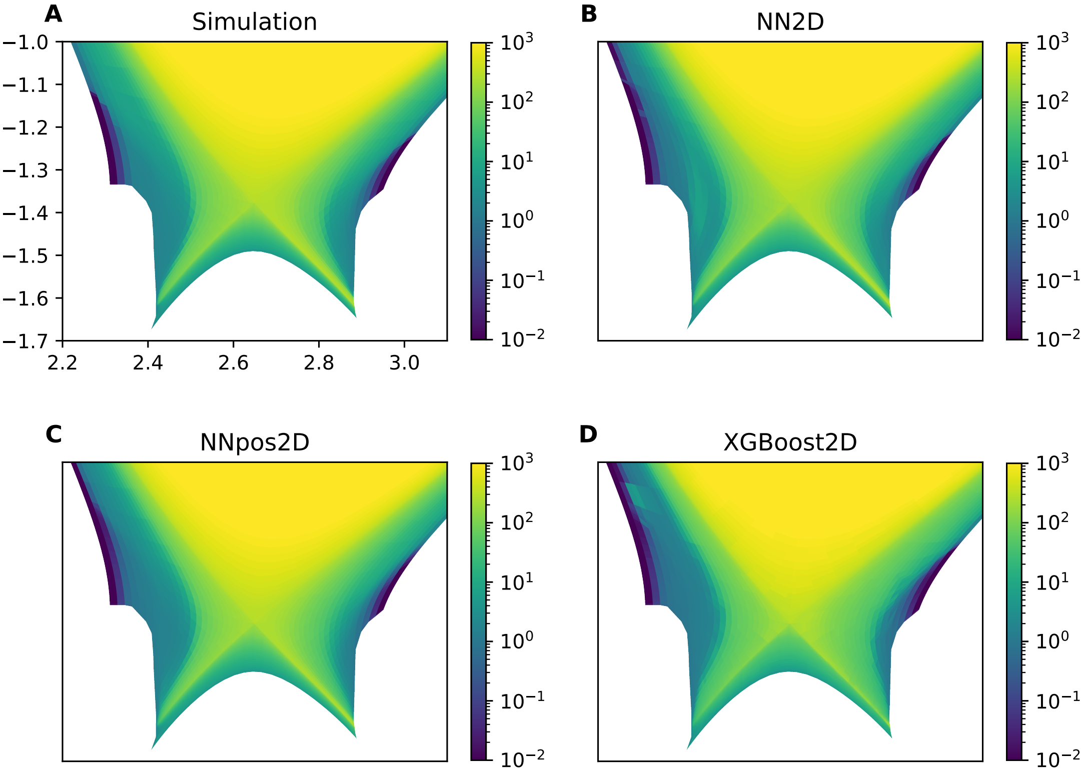


**Figure 8.15:** Comparison between the electron temperatures in the simulation and predictions by the NN2D (B), NNpos2D (C) and XGBoost2D (D) models for one case from the test set. The temperatures are given in eV. The strongest deviations between the models and the simulation can be seen above the inner target.

in all model predictions than it does in the simulation. Additionally, the NN2D model has a speck of visibly higher temperatures in front of the inner target. This mistake is not present in both other models, but the XGBoost2D model has several other visible temperature discrepancies above the inner target. While in this example the most visible deviations all appear close to the inner target, the magnitude and location of such errors change in every test example.

Figure 8.16 shows the 2D model predictions at the outer target for the same exemplary cases as in Figure 8.8. All models generally show similar profiles as the simulations, except for the example in Figure 8.16H. The most and most drastic deviations are seen in the XGBoost2D model (Figure 8.16B, D, G, H), but also both neural network models show visible deviations from the simulations in most examples. Some of the errors done by the XGBoost2D model look very similar as the mistakes of the GBRT and GBRTpos models (comparing subplots B, D, H in Figure 8.16 and Figure 8.8), while other mistakes are new (Figure 8.16G). In comparison to the 1D models NN1D and NNpos1D (Figure 8.8), the 2D network models perform equally well.

Figure 8.17 depicts the model predictions at the outer midplane for the same exemplary test simulations. Also here, the strongest deviations are displayed by the

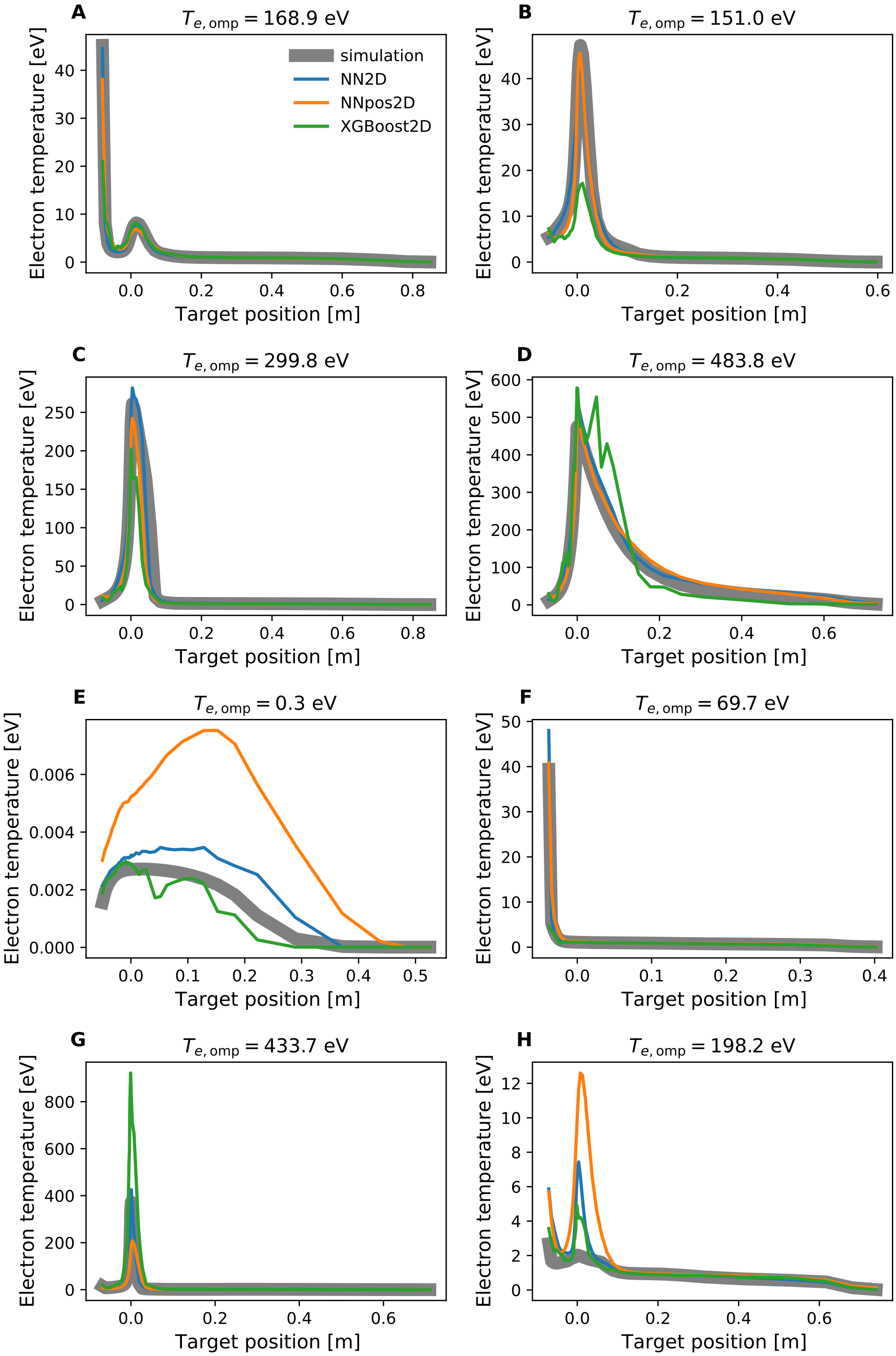


**Figure 8.16:** Comparison between the temperature profiles predicted by the various models at the outer target for eight simulations from the test set. The eight examples are the same as in Figure 8.8. For reference the temperature at the outer midplane separatrix in each simulation is given. Shown are the simulation results (grey) and the predictions by the NN2D (blue), NNpos2D (orange) and XGBoost2D (green) models.

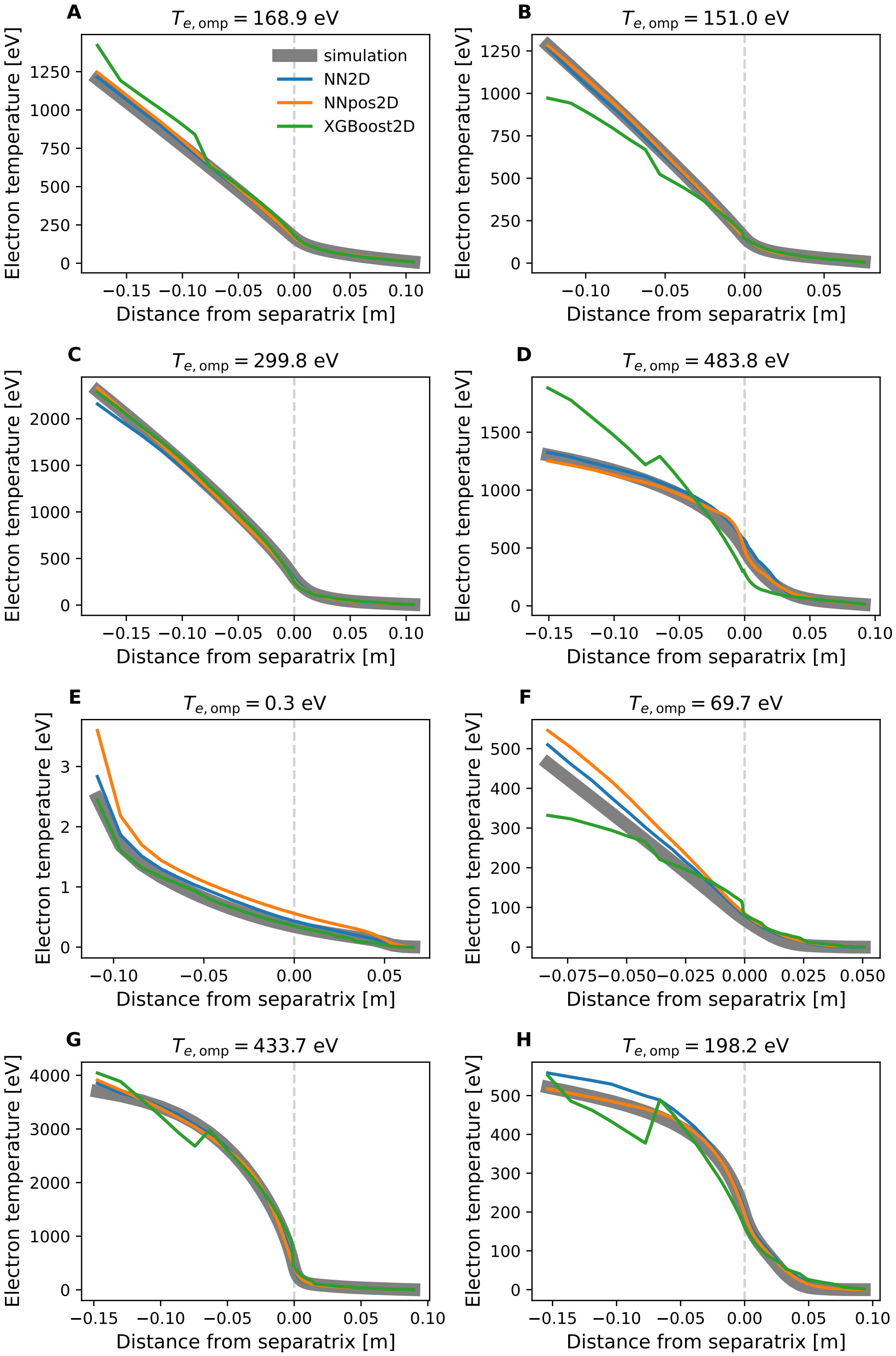


**Figure 8.17:** Comparison between the temperature profiles predicted by the various models at the outer midplane for eight simulations from the test set. The eight examples are the same as in Figure 8.16. For reference the temperature at the outer midplane separatrix in each simulation is given. Shown are the simulation results (grey) and the predictions by the NN2D (blue), NNpos2D (orange) and XGBoost2D (green) models.

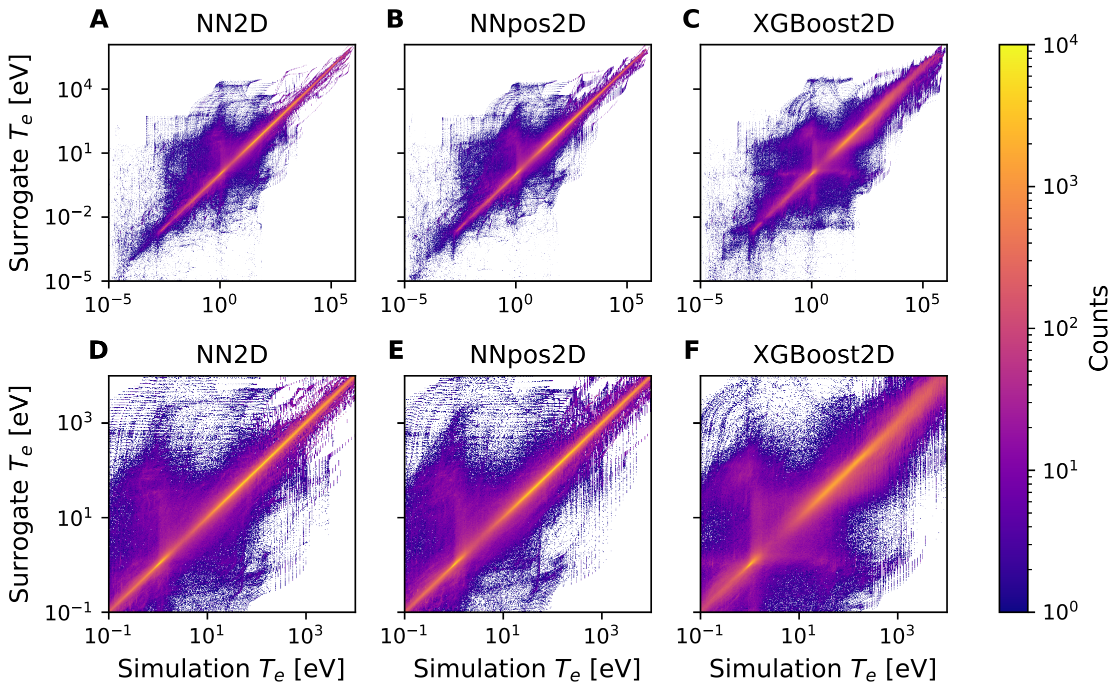


**Figure 8.18:** The temperatures predicted by the NN2D (A), NNpos2D (B) and XGBoost2D (C) models in all grid cells against the respective temperatures in the test simulations. D, E, F show close ups of A, B, C in the relevant temperature range $10^{-1} - 10^4$ eV. The color denotes the number of overlapping points at the same plot position. A model is more precise the closer the predictions are to the diagonal.

XGBoost2D model, while the neural networks match the simulation temperatures fairly well in all cases. In all cases both neural network models capture the shape of the temperature profiles and the deviations that exist are shifts of the whole profiles (Figure 8.17E, F).

Figure 8.18 depicts all predicted temperatures at all domain locations in all test samples against the respective simulation. All three models cover the general systematic in the data and the predicted temperatures are highly correlated with the ground truth in the simulations. Most points lie very close to the optimal diagonal of a perfect prediction, but also many predicted temperatures are orders of magnitude too high or too low. It is clear that the XGBoost2D model does more errors. Specifically around 1 eV a cross like pattern appears (Figure 8.18C, F). The reason behind exactly this pattern is likely because temperatures of 1 eV occur very often in the simulation dataset (Figure 7.7). At extremely high temperatures ($10^5$ eV) the deviations from the simulations are smaller but this is outside of the realistic temperature range.

For a direct comparison to the previous 1D models, Figure 8.19 performs the same visualization just for the temperatures at the outer target. Also in this visualization it is apparent that XGBoost2D has the lowest precision of all three 2D models. NN2D and NNpos2D follow the optimal diagonal closely for the most part, but similar as in the 1D models (Figure 8.9) below 1 eV the predictions become less accurate and more obfuscated (Figure 8.19A, B). In the most relevant temperature

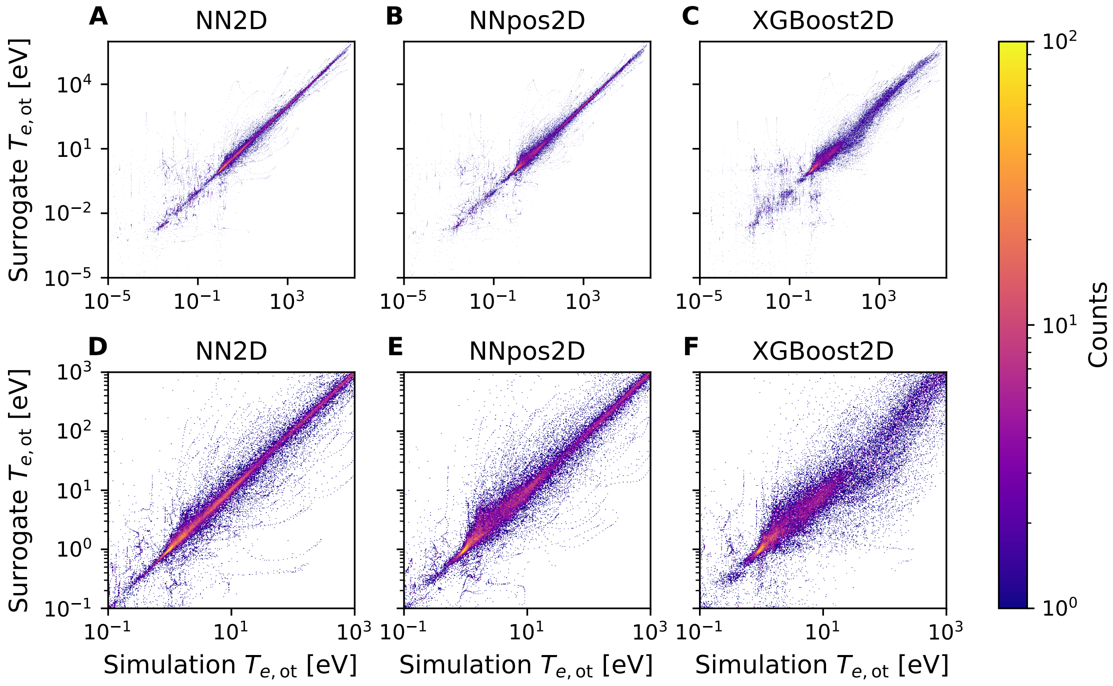


**Figure 8.19:** The temperatures predicted by the NN2D (A), NNpos2D (B) and XGBoost2D (C) models in all outer target grid cells against the respective temperatures in the test simulations. D, E, F show close ups of A, B, C in the relevant temperature range $10^{-1} - 10^3$ eV. The color denotes the number of overlapping points at the same plot position. A model is more precise the closer the predictions are to the diagonal.

range $1 - 100$ eV, the NN2D model is slightly more precise than NNpos2D because the predictions are grouped a little closer to the optimal diagonal (Figure 8.19D, E). In many of the most extreme outliers, NN2D and NNpos2D show similar patterns, which are identical to the ones observed for the 1D network models. Some of these patterns are also picked up by the XGBoost2D model. This further increases the suspicion that some of these large errors stem from poorly converged simulations in the test set.

Figure 8.20 compares the predicted temperatures at the separatrix at the outer midplane to the test simulations. While all three models predictions are highly correlated with the simulation data, it is clear that the XGBoost2D model has the strongest errors (Figure 8.20C, F). In both the NN2D and the NNpos2D model, the simulations in the cold core regime lead to the highest errors while in the other regimes both models perform almost perfect predictions (Figure 8.20A, B). Together with Figure 8.17, this demonstrates that the model predictions at the OMP are generally more reliable than at the outer target. This is anticipated because especially the temperature at the outer midplane is rather stiff and has drastically smaller variance than the temperatures at the outer target (Figure 7.7).

For a quantitative comparison of the different models, Table 8.6 summarizes their median errors on the test set. The NN2D model performs the best by achieving an absolute median error of 3.3 eV over the whole 2D domain with a relative error of only 6.5%. The positional network NNpos2D performs slightly worse but this

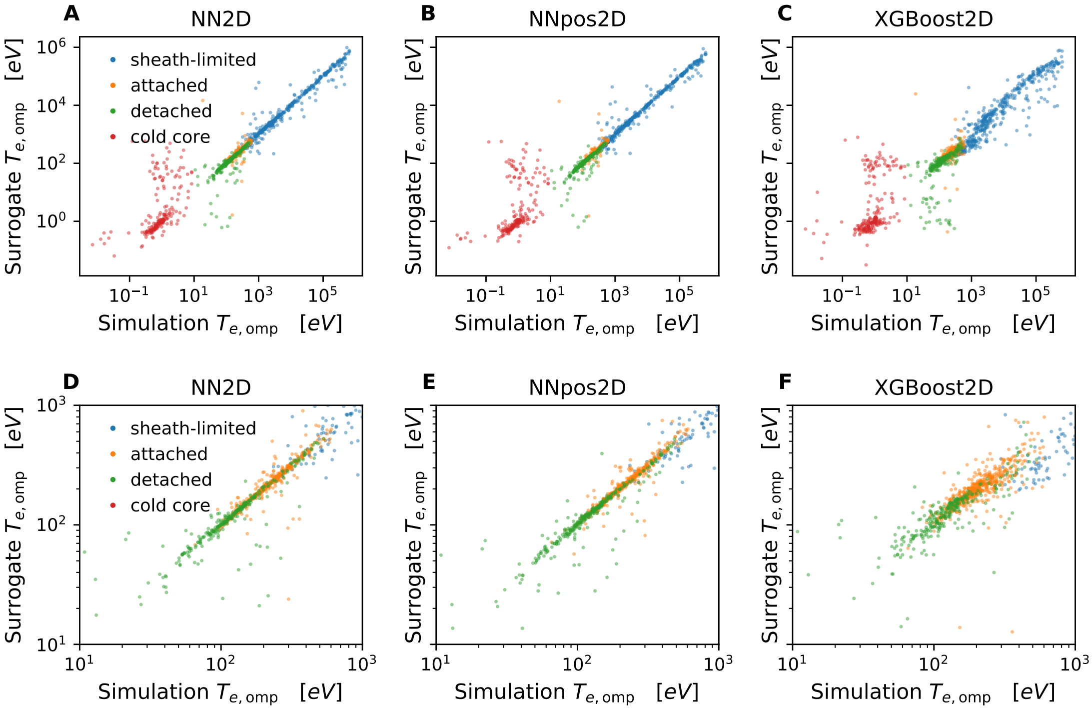


**Figure 8.20:** The temperatures predicted by the NN2D (A), NNpos2D (B) and XGBoost2D (C) models at the outer midplane separatrix against the respective temperatures in the test simulations. D, E, F show close ups of A, B, C in the relevant temperature range $10^1 - 10^3$ eV. The color denotes the regime of the simulation in the test set (as defined in Section 7.4): Sheath-limited (blue), attached (orange), detached (green) and cold core (red). A model is more precise the closer the predictions are to the diagonal.

| | | 2D | 1D | OMP |
|---|---|---|---|---|
| NN2D | abs | 3.3 | 0.97 | 7.1 |
| | rel | 0.065 | 0.12 | 0.043 |
| NNpos2D | abs | 3.8 | 1.6 | 7.8 |
| | rel | 0.068 | 0.19 | 0.041 |
| XGBoost2D | abs | 1.2e+01 | 2.2 | 2.8e+01 |
| | rel | 0.25 | 0.4 | 0.18 |
| NN2D - smaller HPO | abs | 3.5 | 1.0 | 6.8 |
| | rel | 0.068 | 0.13 | 0.043 |

**Table 8.6:** Median absolute and relative errors on the test set, calculated either over all grid points (2D), at the outer target (1D) or directly outside the separatrix at the outer midplane (OMP). The absolute errors are given in eV.

might be attributed to the lower number of trials performed in the hyperparameter optimization. The training times of the best NNpos2D model are with 34 min drastically higher than for the NN2D model with training times of 6 min. The NN2D model with the smaller hyperparameter search yields similar errors as with the wider search. The XGBoost model obtains by far the worst test errors and also requires 10 min to train. The same ranking of the models also holds true when computing the median errors solely at the outer target (1D) or the outer midplane separatrix (OMP). The NN2D model is most accurate whether with the full or the smaller hyper parameter search (Table 8.6). The median errors of the NN2D model at the 1D target (1 eV and $12-13\%$) are identical to the best designated 1D model NN1D. Splitting the analysis of the errors into the different SOL regimes (as done in Table 8.3 for the 1D models) reveals that in all regimes the NN2D model is similarly accurate at the divertor target as the designated NN1D model (not shown here). The NN2D model simultaneously maintains high level of accuracy at the OMP with a median relative error of less than 5%.

To summarize the results of this section: It is possible to develop surrogate models which predict the temperature in the whole SOL at once without compromising the precision at the outer divertor target. Therefore, it can be recommeded to only develop such full 2D models because they offer a wider range of applications without any downsides compared to the target specific models. The best model architecture found so far is a fully connected neural network with independent outputs for each location in the 2D grid (NN2D). The optimal hyperparameters for such a model are similar to the target specific 1D networks.

## 8.4 Influence of the training data size

In sections 8.2 and 8.3 over 5000 successful SOLPS-ITER simulations were used as training data for the surrogate models. These simulations are spread in an exhaustive 8 dimensional parameter space, which covers different machine parameters and physical regimes. For this parameter space 5000 simulations are actually fairly few, and similar projects with lower dimensional parameter spaces have used orders of magnitude more simulations [94]. However, to obtain the results of several thousand SOLPS-ITER simulations, it is required to reduce the complexity of the simulations, which comes at the expensive of lowering the physical accuracy of the simulations. In this thesis the complexity is reduced by using computationally faster fluid neutral models and the neglect of plasma drift effects. But the biases and limitations of the reduced physics description propagate into the trained surrogate models. Two principle approaches are possible to tackle this issue: A surrogate models can be trained on simulations with a mixture of fidelities. In this approach a high number of fast simulations is used to explore the high dimensional parameter space sufficiently dense, and some few slow and complex simulations are added to inform the model about higher fidelity effects. One variant of such a procedure is explored in Section 10. The other approach aims at finding model architectures which can be trained with minimal amounts of training simulations. If the number of necessary training simulations is sufficiently low, these models could be trained on a small dataset consisting entirely of high fidelity simulations. Therefore, this section investigates how the models developed in the previous two sections perform given fewer simulations as training data.

The training simulations are distributed according to a low-discrepancy Sobol sequence [189]. The datapoints in this sequence are not distributed randomly but follow a specific pattern. The sequence is designed in such a way that the first $2^N$ points are distributed to explore the whole parameter space uniformly. Increasing $N$ leads to finer exploration of the parameter space but always maintains a uniform distribution. This property can be used now by employing only the first $2^N < 8192$ training simulations as training data for the surrogate models. The selection based on the Sobol sequence needs to be conducted on the started simulations to ensure the uniform distribution across the parameter space. Because some of the simulations diverge, a selection of $2^N$ started simulations will yield a smaller number of actual simulations with an output that can be used for training. Not only the surrogate models themselves but also the data-preprocessing routines (standard scaling of input parameters, quantile transformation of temperatures) are fitted to the training data. To fully emulate the situation when only the finite amount of training data is available, these routines also need to be refitted on the reduced number of training simulations. Therefore, a comparison of the models trained with different numbers of simulations is only possible after backtransformation into the eV-scale.

The optimal choice of hyperparameters for a model varies with the amount of training data which is available. Therefore, the same hyperparameter searches for all 1D models (as given by Table 8.2 and 8.1) are repeated with varying amounts of training simulations. The only difference is the patience used in neural network training, which is scaled such that the same number of weight updates are waited as in the

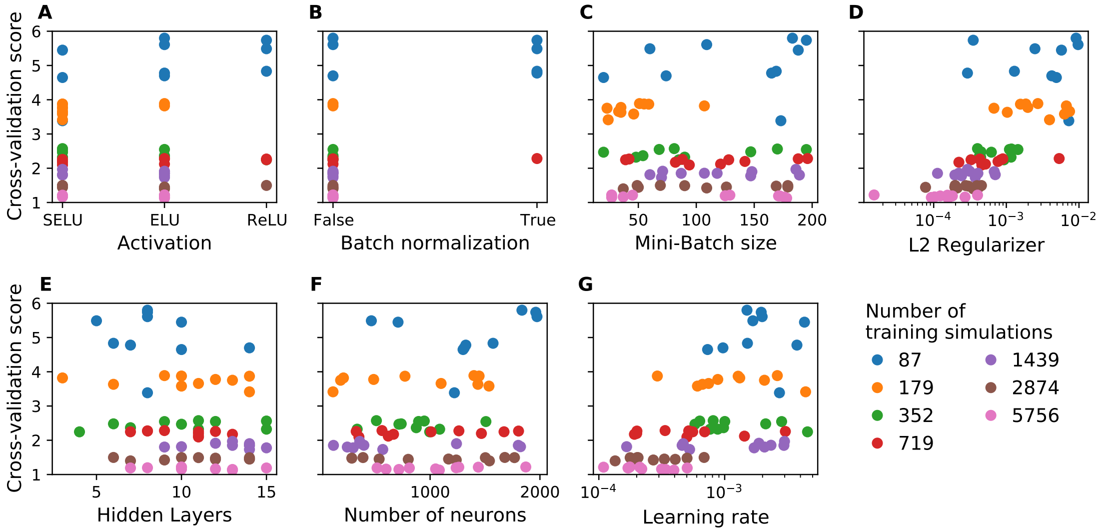


**Figure 8.21:** Cross-validation scores of the ten best NN1D hyperparameter configurations found for each number of training simulations as highlighted by the colors. Each dot represents one tested hyperparameter configuration from the random search.

training with all training simulations. Figure 8.21 depicts the cross-validation score of the ten best NN1D hyperparameter configurations found in the searches for each number of training simulations. The optimal number of neurons or hidden layers is similar for any number of training simulations (Figure 8.21E, F). Also the batch size remains irrelevant for the NN1D performance regardless of the number of training simulations (Figure 8.21C). But with smaller number of training simulations, the stronger regularized networks perform better. With smaller number of simulations the optimal values of the L2 regularizer increases (Figure 8.21D) and similarly the learning rate (Figure 8.21G). Such trends are expected because with a smaller training dataset the risk of overfitting increases, which requires a stronger regularization to counteract. It is interesting that this tendency for stronger regularization does not cause a decrease of the optimal number of hidden layers or neurons. Already the CV scores show a continuous improvement of the achievable model accuracy with increasing number of training simulations.

When training the GBRT model with less simulations, optimal performance is reached at lower learning rates (Figure 8.22A). Surprisingly, the optimal loss function changes with the number of training simulations. With fewer simulations the absolute error seems to be advantageous while for higher number of simulations the squared error is better (Figure 8.22B). Similar as for the neural network also for the GBRT model the overall size of the model changes little. Both the optimal maximum tree depth (Figure 8.22C) and the total number of trees (Figure 8.22D) changes little for the varying numbers of training simulations. There might be a slight shift to smaller tree sizes for lower numbers of training simulations, but this cannot be accurately determined in this test (Figure 8.22C). The strongest change in hyperparameter values is seen for the optimal subsampling fraction (Figure 8.22E). For low number of training simulations, it is beneficial to use most/all training samples when determining data splitting criteria. However, this trend only starts for less than 352 training simulations, while for any higher number of training simulations

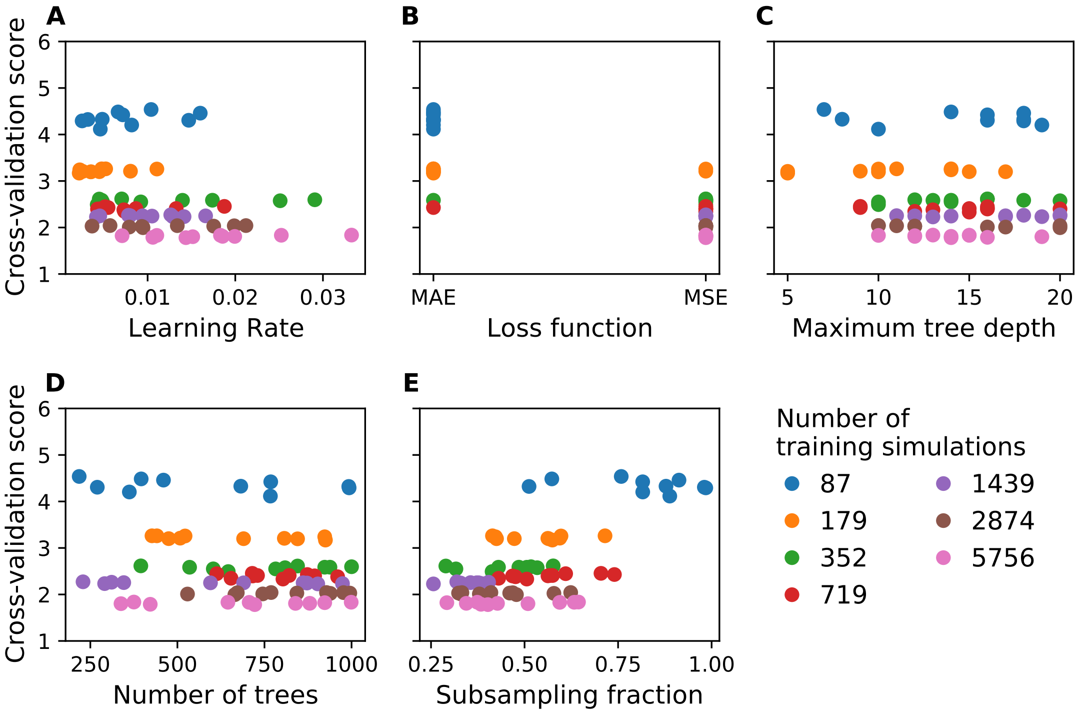


**Figure 8.22:** Cross-validation scores of the ten best GBRT hyperparameter configurations found for each number of training simulations as highlighted by the colors. Each dot represents one tested hyperparameter configuration from the random search.

the optimal value is around 50% subsampling.

Similar full hyperparameter searches were also conducted for the NNpos1D and GBRTpos models. Similar figures analysing their hyperparameters are found in the Appendix (Figures A.2 and A.3). The trends seen in these models are similar to the findings for the NN1D and GBRT model. Although, the increase in subsampling fraction, which is seen for lower number of simulations in the GBRT model, does not appear for the GBRTpos model.

To reduce the computational demand of the hyperparameter optimizations with varying number of training simulations for the 2D models, the hyperparameter searches were limited to varying only the number of hidden layers and neurons. As seen by the comparison between the NN2D and NN2D-small models in Section 8.3, this restriction should not degrade the final model accuracy if reasonable values are chosen for the other hyperparameters. The other hyperparameters of the neural networks are fixed to sensible values, which are estimated from the 1D network optimizations with varying training data. A comparison of the optimal network sizes is found in the Appendix (Figures A.4 and A.5). For the XGBoost2D model no tests regarding the influence of the number of training simulations were conducted. Because the accuracy of XGBoost2D with all simulations is so drastically low compared to the neural network models, it is unlikely that it will outperform them with fewer training examples.

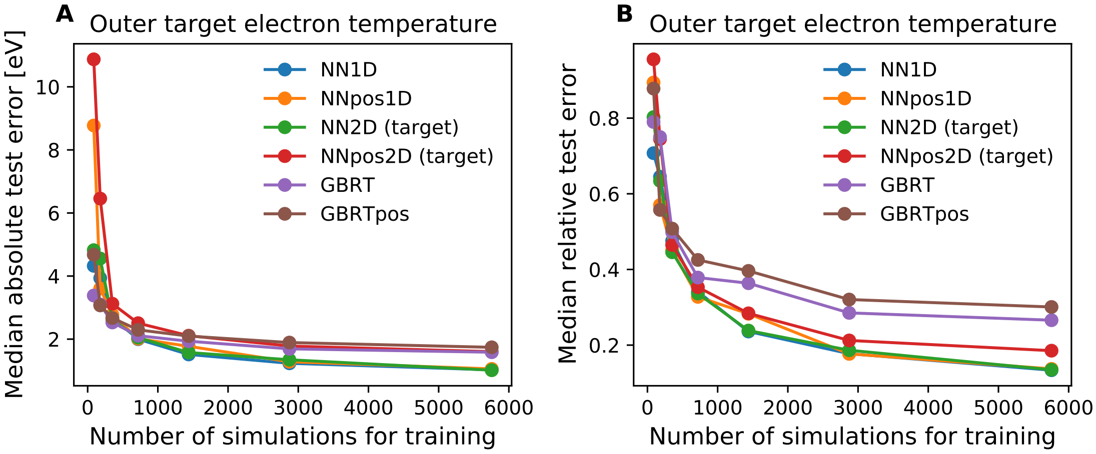


**Figure 8.23:** Median absolute (A) and relative (B) errors of the predicted temperatures at the outer target on the fixed test set for models trained with varying amount of training simulations. The colors denote the type of model. The evaluation on the test set is always performed with the best hyperparameter configuration for each model at the given number of training simulations. For comparison the test scores of the 2D models are only evaluated at the outer target.

| | 87 | 179 | 352 | 719 | 1439 | 2874 | 5756 |
|---|---|---|---|---|---|---|---|
| NN1D | 4.3 | 3.9 | 2.7 | 2.0 | 1.5 | 1.2 | 1.0 |
| NNpos1D | 8.8 | 3.6 | 2.8 | 2.0 | 1.8 | 1.3 | 1.1 |
| NN2D | 4.8 | 4.6 | 2.6 | 2.0 | 1.6 | 1.3 | 1.0 |
| NNpos2D | 1.1e+01 | 6.5 | 3.1 | 2.5 | 2.1 | 1.8 | 1.6 |
| GBRT | 3.4 | 3.1 | 2.5 | 2.1 | 1.9 | 1.7 | 1.6 |
| GBRTpos | 4.7 | 3.1 | 2.7 | 2.3 | 2.1 | 1.9 | 1.7 |

**Table 8.7:** Median absolute errors of the predicted temperatures at the outer target on the fixed test set for models trained with varying amount of training simulations. The evaluation on the test set is always performed with the best hyperparameter configuration for each model at the given number of training simulations. For comparison the test scores of the 2D models are only evaluated at the outer target. The median absolute errors are given in eV.

| | 87 | 179 | 352 | 719 | 1439 | 2874 | 5756 |
|---|---|---|---|---|---|---|---|
| NN1D | 0.71 | 0.65 | 0.48 | 0.34 | 0.24 | 0.18 | 0.13 |
| NNpos1D | 0.89 | 0.57 | 0.45 | 0.33 | 0.28 | 0.18 | 0.14 |
| NN2D | 0.8 | 0.63 | 0.45 | 0.34 | 0.24 | 0.19 | 0.13 |
| NNpos2D | 0.96 | 0.74 | 0.46 | 0.35 | 0.28 | 0.21 | 0.19 |
| GBRT | 0.79 | 0.75 | 0.5 | 0.38 | 0.36 | 0.29 | 0.27 |
| GBRTpos | 0.88 | 0.56 | 0.51 | 0.43 | 0.4 | 0.32 | 0.3 |

**Table 8.8:** Median relative errors of the predicted temperatures at the outer target on the fixed test set for models trained with varying amount of training simulations. The evaluation on the test set is always performed with the best hyperparameter configuration for each model at the given number of training simulations. For comparison the test scores of the 2D models are only evaluated at the outer target. The median absolute errors are given in eV.

To compare the different models with optimal hyperparameters, their predictions aer evaluated on the test set. The set of test simulations is the same as in the previous sections, and independent of the training simulations all models are evaluated on the same full test set. The median errors on the test set are shown in Figure 8.23 and reported in Table 8.7 and Table 8.8. For comparison with the 1D models, the 2D model errors are only evaluated at the outer target. For high number of simulations the NN2D, NN1D and NNpos1D models are almost equally good in terms of the median absolute and the relative errors. The NNpos2D, GBRT and GBRTpos models have very similar median absolute errors for high number of simulations, but the NNpos2D model has lower relative errors. But all three models are worse than the other neural network models. For 352 and 719 training simulations, the differences between all models are smaller (Figure 8.23A). For all models the errors increase with smaller number of training simulations. For less than 100 training simulations, both NNpos1D and NNpos2D show distinctly higher absolute errors compared to all other models and also the highest relative errors although there the difference to the other models is less (Figure 8.23). For 87, 179 or 352 training simulations the GBRT model achieves slightly lower absolute errors than all other models, but its relative errors are still higher than that of the NN1D network.

Overall Figure 8.23 shows that going below 1000 simulations will decrease the model accuracy drastically while the improvements above 1000 simulations are smaller. All models would improve further if more training data is provided. However, the GBRT, GBRTpos and NNpos2D models would benefit less because the gradient of the median errors with regard to the training simulations is smaller than in the other neural network models. Comparing 1D and 2D models, the 1D models show no benefit for any level of training simulations. While the NN2D network is slightly worse than the NN1D model for 87 training simulations, this small difference might vanish when a more complete hyperparameter search is conducted for the 2D model. So for low or high number of available training simulations, NN2D architecture seems like a suitable choice. However, based purely on the median errors it is difficult to determine how many simulations are enough. Especially because the required level of accuracy can differ between different applications.

To get a more intuitive impression of the changes in precision with varying number of training simulations, Figure 8.24 and Figure 8.25 depict the NN2D predictions at the outer midplane and at the outer target for examplary simulations from the test set. The model trained with all available training simulations ($N_{\text{sims}} = 5756$) is identical to the one from Figures 8.17 and 8.16. Clearly the predicted temperature profiles at the outer midplane become more accurate the more training simulations are used in model development (Figure 8.24). With 1439 or 5756 simulations, the model predicts the correct shapes of temperature profiles in all examples and also the absolute values are often correct. Nevertheless, an improvement from 1439 to 5756 simulations is clearly seen. While the model with 352 training simulations is often closer to the simulation than the model trained on 87 simulations, both have always clearly visible discrepancies to the simulations. The absolute deviations of the profiles are far larger inside of the separatrix than in the SOL. At the outer target the assessment shows similar findings as at the outer midplane. The models trained with 87 and 352 simulations perform drastically worse than the other two models (Figure 8.25). The temperature profiles predicted by the models trained with 87 and 352 simulations often clearly differ from the simulations. However, the improvement going from 87 to 352 simulations is more clear than at the outer midplane. In several examples the model with 352 simulations produces profiles which are atleast roughly similar to the simulations (all except B, E, H in Figure 8.25). The model based on 87 simulations does so only in the first example (Figure 8.25A). A clear difference between the predictions of the models with 1439 and 5756 simulations is only visible in three examples (Figure 8.25B, C, D). In all other cases they predict roughly the same.

In summary: The surrogate model accuracy depends tightly on the number of simulations available. Also for smaller number of training simulations, the neural networks with independent outputs for each simulation grid cell (NN1D/NN2D) are the most accurate models. Only for less than 400 training simulations the gradient boosted model (GBRT) becomes comparably accurate. Clear trends regarding the optimal model hyperparameters are visible. For lower number of simulations stronger regularizers and larger learning rates are better. The model error on the test set improves for all models drastically until 1000 training simulations and than more slowly. Also for smaller training datasets, it is not worthwhile to train a model only for the temperatures at the divertor target but the whole SOL can be predicted at once. Based on the comparison with some example profiles, the NN2D model based on 1439 training simulations already offers a high enough precision to be applicable for first scoping studies of different machine parameter combinations.

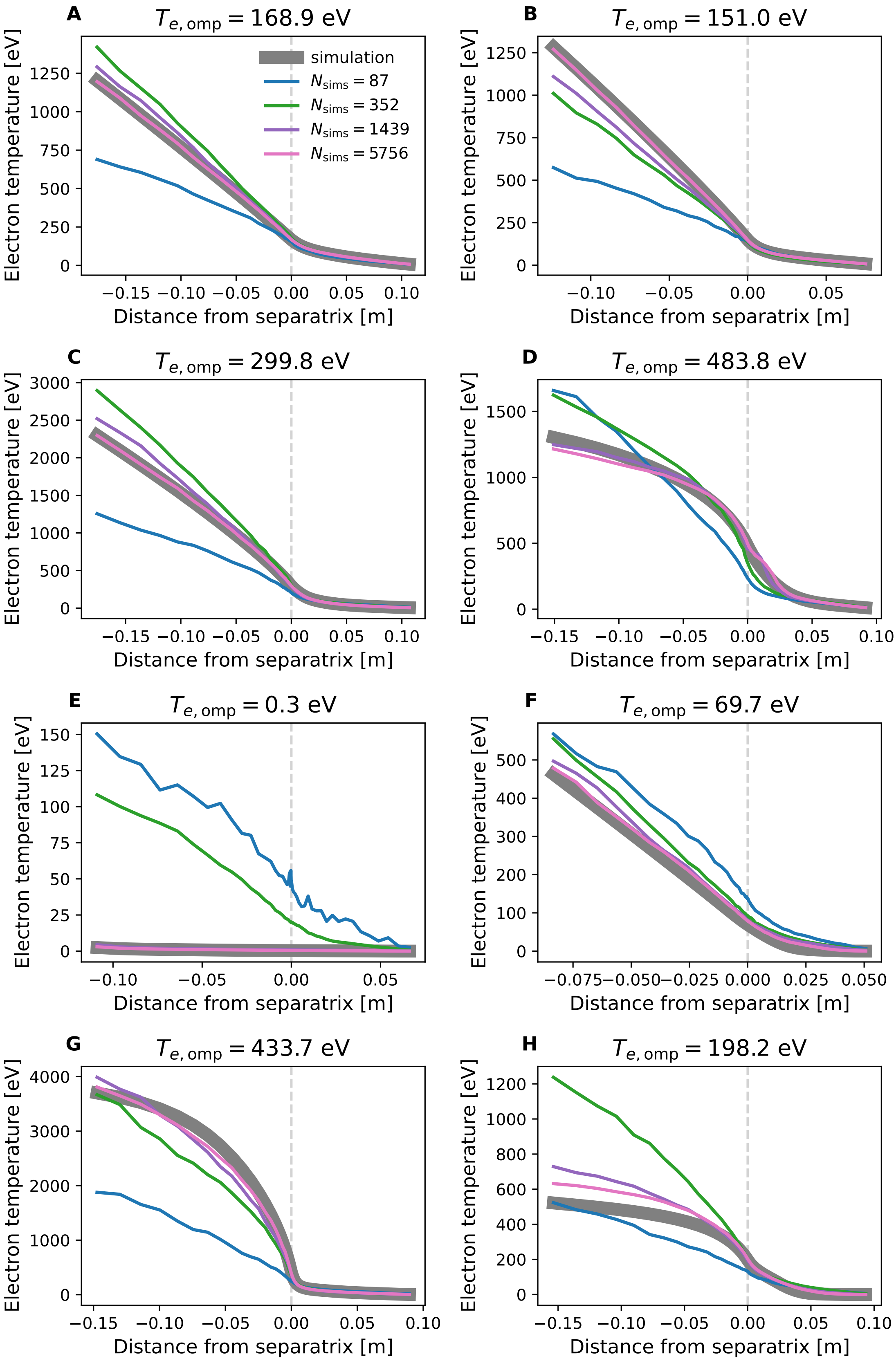


**Figure 8.24:** Comparison between the temperature profiles predicted by NN2D models trained with varying number of training simulations at the outer midplane for eight simulations from the test set. The eight examples are the same as in Figure 8.8. For reference the temperature at the outer midplane separatrix in each simulation is given. Shown are the simulation results (grey) and the predictions by the NN2D model trained with 87 (blue), 352 (green), 1439 (purple) or 5756 (pink) training simulations.

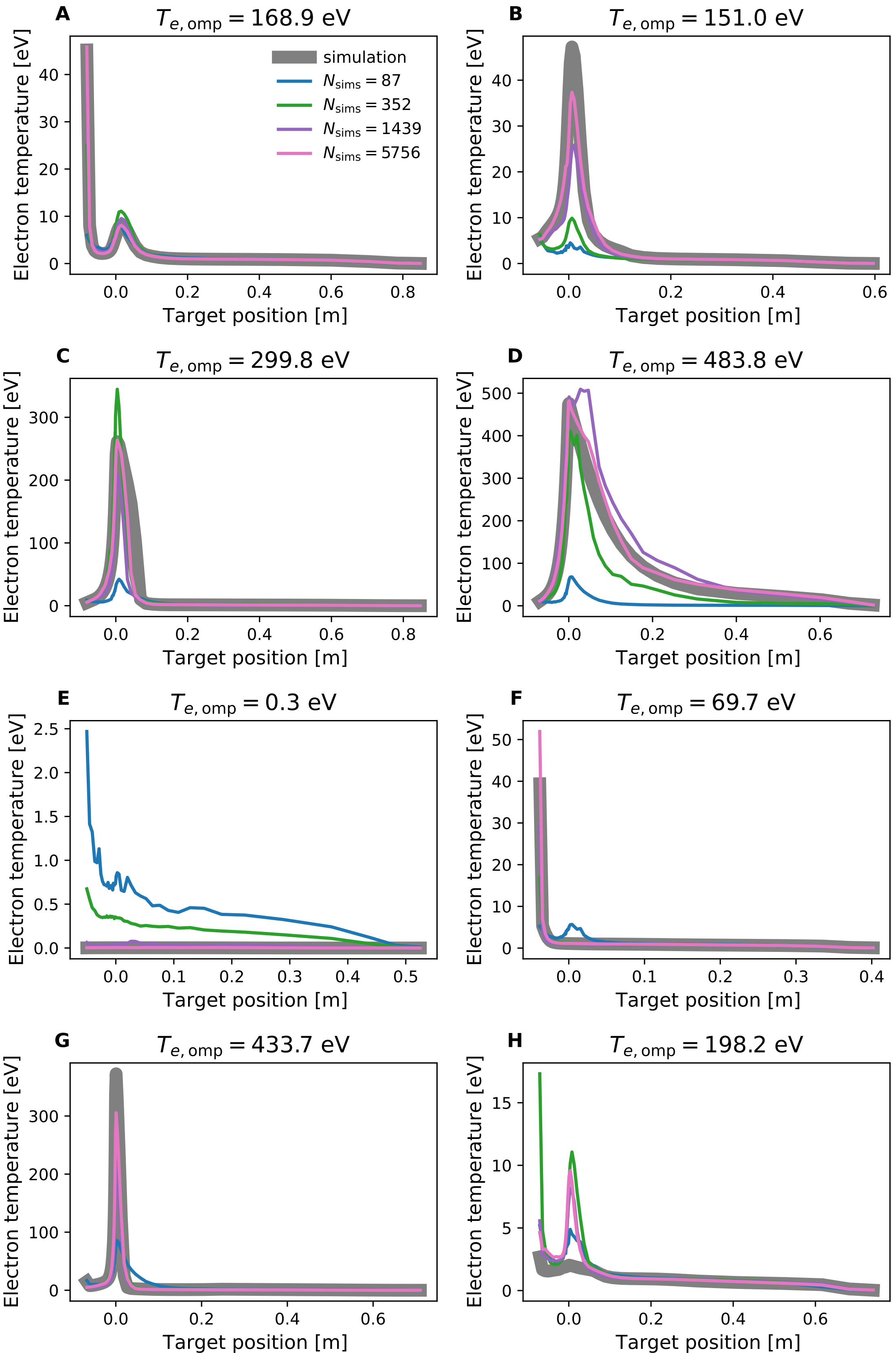


**Figure 8.25:** Comparison between the temperature profiles predicted by NN2D models trained with varying number of training simulations at the outer target for eight simulations from the test set. The eight examples are the same as in Figure 8.8. For reference the temperature at the outer midplane separatrix in each simulation is given. Shown are the simulation results (grey) and the predictions by the NN2D model trained with 87 (blue), 352 (green), 1439 (purple) or 5756 (pink) training simulations.

## 8.5 Incorperation of several plasma properties

The models from the previous sections concentrate on predicting only the electron temperature. For a complete description of the plasma processes in the SOL, many more observables are necessary. Including more properties in the surrogate model raises the same question as in the step of moving from 1D to 2D models: Is it better to include everything into one model or split it into several distinct models? A single model might preserve the correlations between observables better but can be more difficult to train and optimize. Which of these factors outweighs the other is specific to each problem. The goal of this section is to compare both approaches and find out which strategy is better for the SOLPS-ITER dataset here.

In addition to the existing NN2D model for the electron temperature $T_e$, three additional networks with the same NN2D architecture are trained independently for the electron density $n_e$, deuterium neutral gas density $n_D$ and nitrogen neutral gas density $n_N$. For each of these networks a full hyperparameter search is conducted with the same procedure as in Section 8.3. Many more observables are relevant in the SOL, but these four are selected as proxies to test the approaches. Later a full model for all quantities of interest can be build (Section 8.6). As comparison, two networks are trained which each predict all four quantities $T_e, n_e, n_D, n_N$ in the whole 2D domain at once. These networks are also of the NN2D type but the sizes of the last layers are changed to accommodate for the higher number of outputs. In the first network (NN2D-all-in-one) only the size of the output layer is increased to 4x5400 neurons. Because the output layer has no activation function itself, the outputs are weighted linear combinations of the neuron outputs in the last hidden layer. Because the number of neurons in the hidden layers is drastically lower than in the output layer, this last step corresponds to a decompression of a lower dimensional representation of the plasma state to the final high dimensional result. If this decompression can only be poorly represented by the linear operation in the last layer, this sudden change in number of neurons could bottleneck the network performance. Therefore, the second network (NN2D-all-in-one-2) has an increased size of 4x5400 neurons in the last two layers. Also for these two networks a hyperparameter search is conducted as in Section 8.3. Additionally, a network of the NNpos2D architecture is trained which also includes all four observables in one. This network (NNpos2D-all-in-one) has in principle the same architecture as the NNpos2D in Section 8.3, but instead of a single scalar output it has four scalar outputs, one for each observable. The hyperparameters of this network are not optimized but prescribed. The network contains 10 hidden layers with each 300 neurons and a batch size of 4096 and a 10 epochs patience in the early stopping. All other hyperparameters are identical to the NNpos2D in Table 8.4. Like in the previous sections, the plasma observables are scaled using quantile transformations independently for each observable and each grid location, and the tokamak parameters use the regular standardization.

In order not to make the analysis unnecessarily lengthy, only the model predictions at the outer midplane are compared and evaluated in this section. However, the conclusions drawn from this analysis are the same if the models are compared in the whole SOL or at the outer divertor target. The figures and tables comparing

| | | OMP | | | | |
|---|---|---|---|---|---|---|
| | | $T_e$ | $n_e$ | $n_D$ | $n_N$ | $p_e$ |
| NN2D-all-in-one | abs | 8.5 | 9.5e+17 | 5.4e+13 | 4.8e+11 | 3.1e+01 |
| | rel | 0.046 | 0.065 | 0.11 | 0.077 | 0.062 |
| NN2D-all-in-one-2 | abs | 9.0 | 1e+18 | 5.8e+13 | 4.3e+11 | 3.1e+01 |
| | rel | 0.054 | 0.073 | 0.14 | 0.082 | 0.068 |
| NN2D-separated | abs | 7.1 | 7.6e+17 | 4.3e+13 | 3.9e+11 | 2.9e+01 |
| | rel | 0.043 | 0.062 | 0.1 | 0.058 | 0.06 |
| NNpos2D-all-in-one | abs | 8.4 | 7.7e+17 | 4.4e+13 | 3.6e+11 | 2.8e+01 |
| | rel | 0.048 | 0.054 | 0.1 | 0.065 | 0.058 |

**Table 8.9:** Median absolute and relative errors on the test set of the model predictions at the outer midplane separatrix. The absolute errors are given in the units $[T_e] = \mathrm{eV}$, $[n_e] = [n_D] = [n_N] = \mathrm{m}^{-3}$,$[p_e] = \mathrm{Pa}$. The models do not predict the pressure $p_e$ directly but it is computed as $p_e = n_e T_e$.

the models at these different locations are shown in Appendix A.3. In addition to the four predicted quantities, the comparisons of the model predictions include also the static plasma pressure $p_e = n_e \cdot T_e$. The pressure is derived from the electron temperatures and densities which are predicted by the models. This is introduced to test whether the combined models capture the correlations between $n_e$ and $T_e$ more accurately than the separate models. In that case the combined models would deliver more precise predictions for $p_e$.

Table 8.9 depicts the median errors on the test of the final trained models (with optimized hyperparameters). The median errors are calculated only for the first grid cell outside the separatrix at the outer midplane. For almost all observables the NN2D-all-in-one-2 network achieves the worst results. So the anticipated bottleneck in the NN2D-all-in-one architecture poses not a problem. The second worst results are obtained by the NN2D-all-in-one model, which is surpassed by both the NN2D-separated and the NNpos2D-all-in-one model in all metrics. Also for the plasma pressure the NN2D model with the separated networks obtains slightly lower test errors than the NN2D-all-in-one model, although the differences are small. The NN2D-separated and the NNpos2D-all-in-one model are very similar in the median errors on all three densities and the plasma pressure. But concerning the electron temperature, the NN2D-separated model is slightly more accurate. For all four models the median relative errors of the predicted temperatures are lower than for the other quantities. This hints that the temperature might be easier to predict than the particle densities.

For all models, except NN2D-all-in-one-2, the predictions at the outer midplane separatrix are shown in Figure 8.26. In this visualization none of the three models appears more accurate than another. All models share a pattern of incorrect predictions for nitrogen densities of $10^{10}\,\mathrm{m}^{-3}$ (Figure 8.26J, K, L). This values was set as the numerical lower limit for the nitrogen density in any grid cell in the simulations.

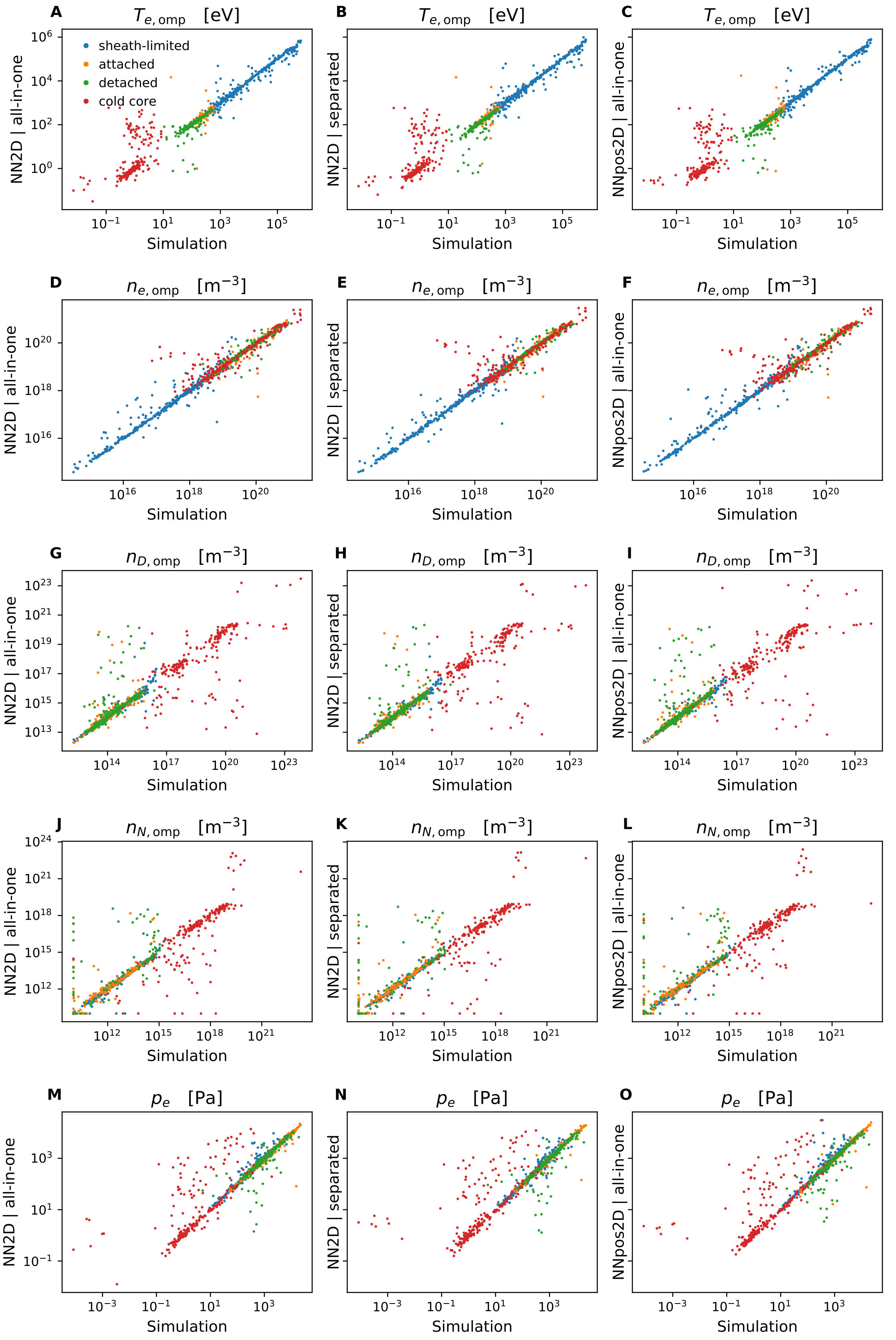


**Figure 8.26:** The model predictions at the outer midplane separatrix against the results in the simulations in the test set. The color denotes the regime of the simulation in the test set (as defined in Section 7.4): Sheath-limited (blue), attached (orange), detached (green) and cold core (red). A model is more precise the closer the predictions are to the diagonal. The models do not predict the pressure $p_e$ directly but it is computed as $p_e = n_e T_e$. The vertical and horizontal lines in J, K, L appear because $10^{-10}\,\mathrm{m}^{-3}$ was set as lower limit for $n_N$ in the simulations.

Also comparing the predicted plasma pressure profiles at the outer midplane (Figure 8.27) shows that all three models are equally capable of predicting accurate plasma pressures. Similar visualizations for the other observables can be found Appendix A.3.

Relying on the median errors, the NN2D-separated model is the most accurate model by a tight margin. To include even more observables, the NN2D-all-in-one model needs to be retrained with an even larger output layer in the network. As the current increase in network output size has slightly decreased the model accuracy, it is likely that a further size increase would further decrease the model accuracy. In contrast, adding more observables to the NN2D-separated model is trivial. This simply requires the training of additional networks, without any impact on the already existing. The accuracy in the temperature prediction of the NNpos2D-all-in-one has decreased compared to the positional network that predicted solely the temperature (comparing Table 8.9 to Table 8.6). Therefore, also for the NNpos2D-all-in-one model the accuracy would likely decrease further in a model containing more observables. This might be counteracted by a hyperparameter optimization, but the longer training times of the positional architecture make extensive hyperparameter optimizations computationally expensive.

One of the theoretical benefits of the NN2D-separated model is that the hyperparameters of each of the independent networks can be tailored to the specific observable. Interestingly this does not occur, but instead all networks in the NN2D-separated model have similar hyperparameters. Table 8.10 shows the hyperparameters of the five best networks found in the hyperparameter searches for each of the observables. All of them use either the SELU or ELU activation function and similar values for the number of hidden layers, L2 regularizer and learning rate. A higher variation is found in the number of neurons per layer and the batch size used during training. But these two parameters are less relevant for the model performance and can be arbitrarily chosen inside some sensible ranges (see Figure 8.11). Therefore, the values which appear in the best hyperparameter configurations are just random.

In conclusion: To predict more plasma properties besides the electron temperature, all can be included in a singular neural network or in separate networks for each quantity. Both approaches obtain similar levels of accuracy, but using separate networks can be trivially scaled to include even more observables. Additionally, the separate networks all use a similar set of optimal hyperparameters. Therefore, it is not necessary to run new hyperparameter optimizations for each new observable, but the same hyperparameters can be reused in each separate network.

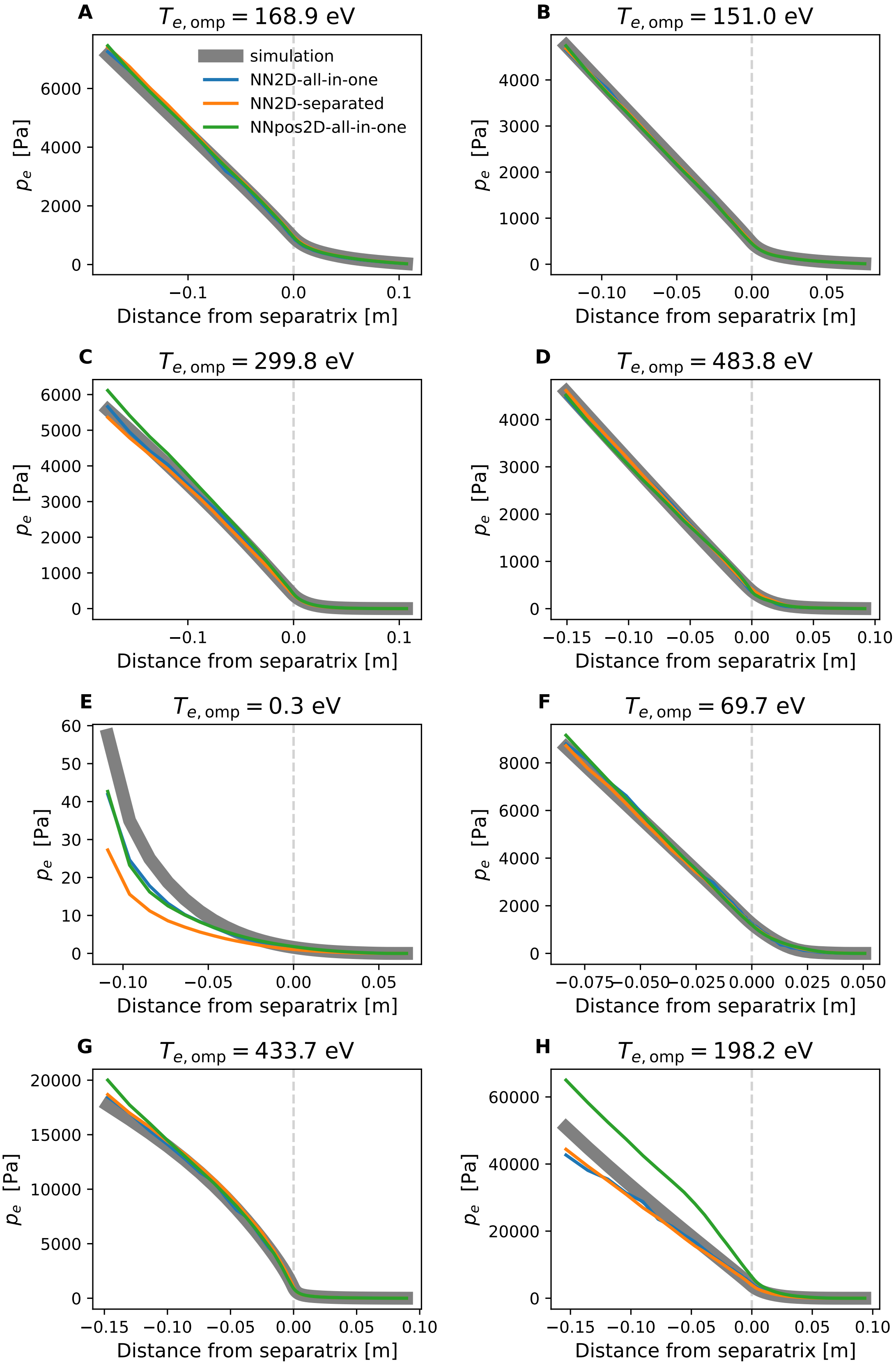


**Figure 8.27:** Comparison between the electron pressure profiles computed from model predictions at the outer midplane for eight simulations from the test set. The eight examples are the same as in Figure 8.16. For reference the temperature at the outer midplane separatrix in each simulation is given. Shown are the simulation results (grey) and the predictions computed from the NN2D-all-in-one (blue), NN2D-separated (orange) and NNpos2D-all-in-one (green) models.

| | CV score | Activation | Batch size | L2 | Hidden Layers | Number of neurons | Learning rate |
|---|---|---|---|---|---|---|---|
| $T_e$ | 3.9e+0 | SELU | 37 | 3.4e-5 | 9 | 1427 | 1.0e-4 |
| | 3.9e+0 | SELU | 50 | 8.7e-5 | 11 | 1057 | 1.0e-4 |
| | 4.1e+0 | ELU | 106 | 5.2e-5 | 8 | 464 | 1.4e-4 |
| | 4.2e+0 | SELU | 61 | 1.4e-4 | 11 | 1205 | 1.1e-4 |
| | 4.2e+0 | SELU | 47 | 3.8e-5 | 11 | 944 | 2.7e-4 |
| $n_e$ | 1.6e+18 | SELU | 86 | 1.0e-4 | 11 | 965 | 2.8e-4 |
| | 1.6e+18 | ELU | 115 | 2.1e-5 | 10 | 1097 | 1.6e-4 |
| | 1.7e+18 | SELU | 195 | 1.0e-4 | 10 | 1478 | 3.2e-4 |
| | 1.7e+18 | SELU | 197 | 8.3e-5 | 8 | 772 | 3.7e-4 |
| | 1.7e+18 | SELU | 192 | 4.8e-5 | 15 | 1023 | 4.e-4 |
| $n_D$ | 5.0e+14 | SELU | 192 | 8.7e-5 | 10 | 1238 | 1.5e-4 |
| | 5.1e+14 | ELU | 96 | 1.1e-4 | 15 | 1291 | 1.7e-4 |
| | 5.3e+14 | SELU | 119 | 9.8e-5 | 11 | 918 | 2.6e-4 |
| | 5.4e+14 | ELU | 77 | 2.5e-4 | 12 | 861 | 1.0e-4 |
| | 5.4e+14 | SELU | 176 | 4.5e-5 | 14 | 1139 | 2.5e-4 |
| $n_N$ | 5.5e+12 | ELU | 127 | 6.0e-5 | 12 | 1254 | 1.3e-4 |
| | 5.6e+12 | ELU | 151 | 3.9e-5 | 9 | 1020 | 2.1e-4 |
| | 6.0e+12 | SELU | 118 | 3.4e-5 | 15 | 536 | 1.1e-4 |
| | 6.2e+12 | ELU | 89 | 7.8e-5 | 13 | 317 | 1.6e-4 |
| | 6.3e+12 | SELU | 181 | 1.5e-5 | 12 | 641 | 1.2e-4 |

**Table 8.10:** The hyperparameters of the five best network architectures found for each observable in the NN2D-separated model. None of the models use Batch normalization, therefore the column was removed in this overview. A hyperparameter configuration is better when it has a smaller CV score.

## 8.6 SOLPS-ITER in the loop

The goal of this section is to go forward and build a surrogate model for all plasma quantities. Based on the findings in the previous section, each quantity is predicted by a separate NN2D network with a fixed network architecture (10 hidden layers with each 1000 SELU neurons, learning rate $10^{-4}$, L2 regularizer $10^{-4}$, batch size 64). For each independent network, the output quantities are scaled using a quantile transformation. This approach can lead to predicting the wrong sign for plasma velocities, but also more complicated setups mitigate this error source only slightly in comparison to a plain NN2D model with quantile transformations (Appendix A.4).

A list of all quantities of interest (QoI) in the SOL can become quite extensive because different applications require different things. For exhaust control the heat and particle fluxes at the divertor are relevant, coupling to pedestal models requires densities and temperatures at the separatrix, comparison with experiment might require radiation profiles or line integrated quantities, etc. Training independent models to predict all of these QoI can become exhaustive. This would require users with very specialized applications to train their own models from scratch based on the simulation database. Instead of predicting all QoI directly, another approach is to predict all independent state variables of the SOLPS-ITER simulations because from these all other plasma quantities can be derived. This section compares the two approaches by predicting: the power crossing the separatrix $P_{\mathrm{SOL}}$, the peak heat flux at the outer divertor target $q_{\mathrm{peak,ot}}$ and the integrated deuterium ion flux at the outer divertor target $\Gamma_{D_+,\mathrm{ot}}$.

In the simulation setup in this thesis, the state variables are the electron temperature $T_e$, ion temperature $T_i$, densities of all ion and neutral particles $(n_D, n_{D+}, n_N, n_{N+}, ..., n_{N7+})$ and the parallel velocities of all ions and neutral particles $(u_D, u_{D+}, u_N, u_{N+}, ..., u_{N7+})$. The electron density follows from quasineutrality and the potential equation is not solved but a potential of $3.1T_e/e$ is assumed. For each of these 22 state variables a NN2D network is trained with fixed hyperparameters. To calculate all derived quantities and extra outputs from the state variables, SOLPS-ITER can be used (NN2D+SOLPS). Thus, the outputs of these networks are written into input files for SOLPS-ITER (all other fields in the input files are set to zero). Instead of using own implementations for calculating the derived quantities, using SOLPS-ITER ensures that the same assumptions are used in all derivations as in the underlying simulations. But depending on the exact SOLPS-ITER version, one has to be careful that all dependent quantities are recalculated from scratch instead of using the wrong values in the input file. The procedure that is used here is given in Chapter 13. This workflow is slower than predicting the QoI directly, but since the code has to compute only the auxiliary outputs this is still orders of magnitudes faster than actually running a simulation over many iterations (seconds instead of days).

To predict the three QoI directly, a network is trained for each of them. The networks hyperparameters are optimized like the NN1D network in Table 8.2. The only difference is that the output of each of the three networks is only a single scalar which corresponds to the respective QoI.

| | | | all | sheath-limited | attached | detached | cold core |
|---|---|---|---|---|---|---|---|
| Direct | $P_{\mathrm{SOL}}$ | abs | 0.51 | 0.38 | 0.51 | 0.61 | 1.3 |
| | | rel | 0.0076 | 0.0048 | 0.0057 | 0.0096 | 0.46 |
| | $q_{\mathrm{peak,ot}}$ | abs | 0.98 | 0.71 | 1.7 | 1.9 | 5e-05 |
| | | rel | 0.13 | 0.055 | 0.092 | 0.34 | 0.98 |
| | $\Gamma_{D^+,\mathrm{ot}}$ | abs | 6.5e+22 | 5.4e+20 | 2.1e+23 | 2.5e+23 | 3.6e+19 |
| | | rel | 0.15 | 0.15 | 0.085 | 0.15 | 6.7 |
| NN2D | $P_{\mathrm{SOL}}$ | abs | 1.1e+01 | 1.7e+01 | 1.1e+01 | 9.3 | 0.77 |
| | | rel | 0.19 | 0.24 | 0.13 | 0.16 | 0.72 |
| | $q_{\mathrm{peak,ot}}$ | abs | 1.1e+01 | 4.4e+01 | 8.7 | 2.5 | 0.00076 |
| | | rel | 1.7 | 3.1 | 0.43 | 0.41 | 1.5e+02 |
| | $\Gamma_{D^+,\mathrm{ot}}$ | abs | 7.1e+22 | 4.4e+20 | 1.9e+23 | 2.5e+23 | 2.3e+19 |
| | | rel | 0.15 | 0.13 | 0.089 | 0.16 | 1.0 |
| NNpos2D | $P_{\mathrm{SOL}}$ | abs | 9.4 | 1.4e+01 | 1e+01 | 7.8 | 0.86 |
| | | rel | 0.15 | 0.18 | 0.12 | 0.13 | 0.49 |
| | $q_{\mathrm{peak,ot}}$ | abs | 9.8 | 1.6e+01 | 1.7e+01 | 2.7 | 0.025 |
| | | rel | 0.91 | 0.98 | 0.6 | 0.49 | 3.4e+03 |
| | $\Gamma_{D^+,\mathrm{ot}}$ | abs | 7e+22 | 5e+20 | 3e+23 | 3.1e+23 | 2.1e+19 |
| | | rel | 0.17 | 0.12 | 0.13 | 0.18 | 1.0 |
| NN2D+1000 | $P_{\mathrm{SOL}}$ | abs | 0.39 | 1.4 | 0.14 | 0.19 | 0.53 |
| | | rel | 0.011 | 0.023 | 0.0016 | 0.0033 | 0.24 |
| | $q_{\mathrm{peak,ot}}$ | abs | 0.79 | 0.75 | 1.4 | 2.3 | 3.6e-05 |
| | | rel | 0.12 | 0.046 | 0.074 | 0.23 | 0.99 |
| | $\Gamma_{D^+,\mathrm{ot}}$ | abs | 7e+22 | 3.2e+20 | 2.1e+23 | 2e+23 | 7e+18 |
| | | rel | 0.12 | 0.08 | 0.084 | 0.12 | 1.3 |

**Table 8.11:** Median absolute and relative errors on the test set of the three model predicted QoI. For the NN2D, NNpos2D and NN2D+1000 models the quantities are not predicted directly but computed by inserting the model predicted plasma state variables into SOLPS-ITER. The errors are evaluated either on the whole test set or only those simulations in certain regimes (as defined in Section 7.4). The absolute errors are given in the units $[P_{\mathrm{SOL}}] = \mathrm{MW}$, $[q_{\mathrm{peak,ot}}] = \mathrm{MW/m^2}$, $[\Gamma_{D^+,\mathrm{ot}}] = \mathrm{atoms/s}$.

Table 8.11 reports the median test errors for all three QoI in both the NN2D+SOLPS approach and the direct prediction. Both approaches achieve comparable errors in the predicted particle flux $\Gamma_{D+,\mathrm{ot}}$ in all regimes. The direct approach is able to predict $P_{\mathrm{SOL}}$ excellently and obtains median relative errors of less than 1% in almost all regimes. This can be expected as on of the inputs to the model is the input power $P_{\mathrm{in}}$ into the simulation domain at the core boundary, which is highly correlated with $P_{\mathrm{SOL}}$. Strong differences between $P_{\mathrm{in}}$ and $P_{\mathrm{SOL}}$ are only expected when the radiation inside the confined region becomes strong. This is the case in the cold core regime, which explains why the accuracy in predicting $P_{\mathrm{SOL}}$ is so much lower there. The NN2D+SOLPS predictions of $P_{\mathrm{SOL}}$ are less accurate and achieve only median relative errors of more than 10% in all regimes. In the direct approach the predictions for $P_{\mathrm{SOL}}$ in the sheath-limited regime are slightly more accurate than in attached and detached cases. However, NN2D+SOLPS approach is less accurate in the sheath-limited regime than in the attached and detached. A similar pattern but a lot more extreme is found in the errors of the peak target heat flux $q_{\mathrm{peak,ot}}$. The direct approach is much more accurate in the sheath-limited regime than in either the attached or detached cases. In the NN2D+SOLPS approach, it is the other way around and the accuracy is far lower in the sheath-limited regime. The NN2D+SOLPS approach yields a median relative error of 310% for $q_{\mathrm{peak,ot}}$ in the sheath-limited regime. In the attached regime, the NN2D+SOLPS solution still has higher errors than the direct prediction but the differences are smaller than in the sheath-limited regime. In the detached regime the accuracy of both approaches is similar. In the cold core regime the median error of the NN2D+SOLPS approach is drastically increased. But because the heat fluxes in this regime are almost non existent, even small absolute errors can cause these drastic relative deviations.

Figure 8.28 visualizes the predicted QoI against the simulation values in the test simulations, such that for a perfect prediction all points would lie on the diagonal. In the direct approach the predictions of all three QoI lie very close to the values in the simulations, with the highest accuracy for $P_{\mathrm{SOL}}$ (Figure 8.28A, B, C). For $P_{\mathrm{SOL}}$ the predictions of the NN2D+SOLPS approach follow the results from the simulations loosely but with higher deviations than in the direct approach (Figure 8.28D). The ion flux $\Gamma_{D+,\mathrm{ot}}$ is predicted similarly precise as in the direct approach (Figure 8.28F). For the peak target heat flux $q_{\mathrm{peak,ot}}$, the predictions in the detached simulations follow the expected values, but for all sheath-limited and some of the attached cases a systematic shift towards higher values is observed (Figure 8.28E). In the range between $1 - 100\,\mathrm{MW/m^2}$, the NN2D+SOLPS approach predicts peak heat fluxes systematically too large by a factor $\approx 4$. But only sheath-limited and some attached cases are affected by this because in the same range the predictions for detached simulations are accurate. This observation explains, why the median errors of the peak heat fluxes predicted by the NN2D+SOLPS model in Table 8.11 are so much higher in the sheath-limited regime.

To identify the root causes behind these systematic errors in the sheath-limited regime, many simulations and their predictions in the NN2D+SOLPS approach have been inspected in-depth. This allowed the detection of a common property in the erroneous predictions, which is not found in the correct predictions. Figure 8.29 illustrates this property. Figures 8.29A, B, C depict the conditions in a de-

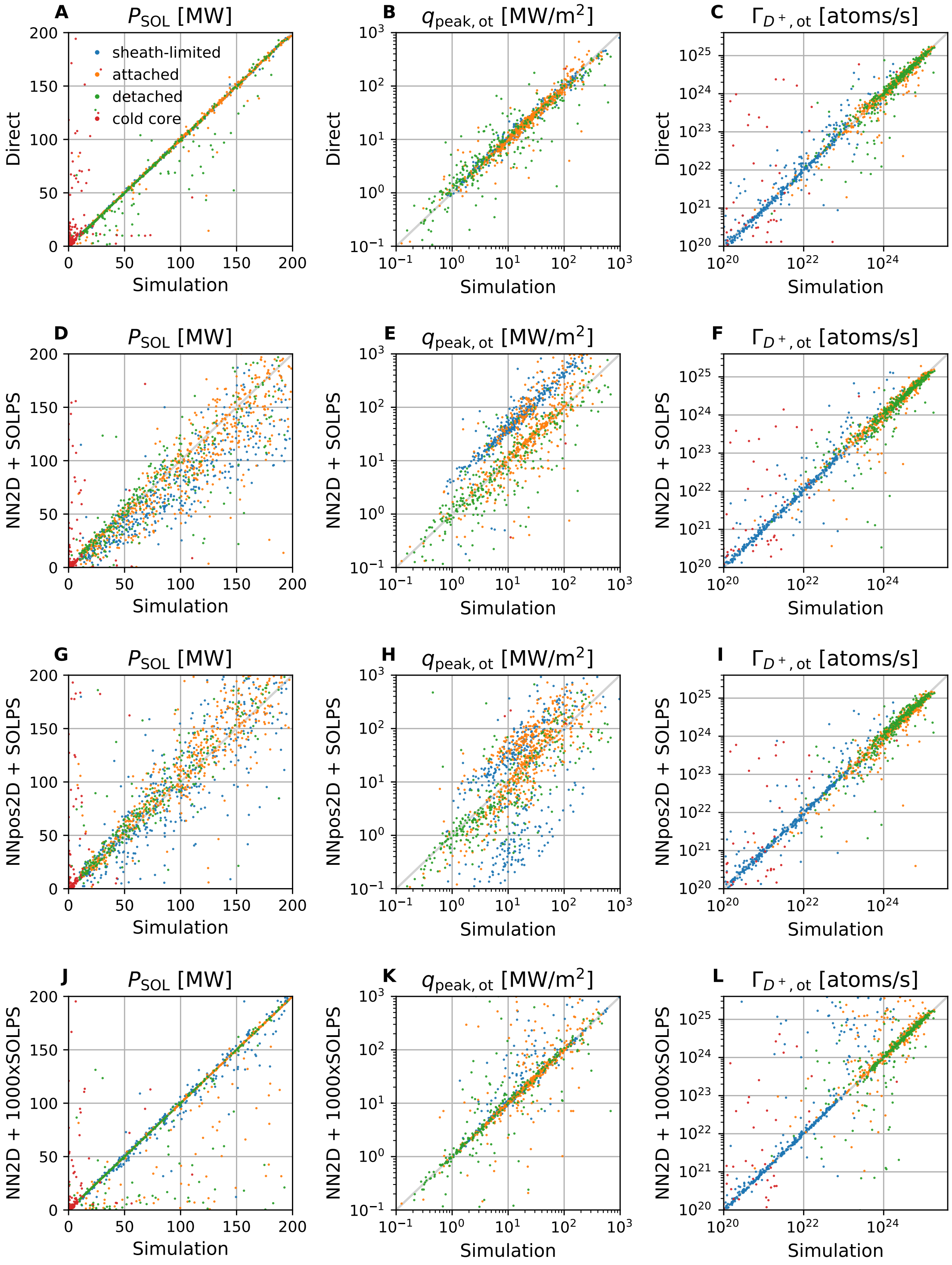


**Figure 8.28:** The model predictions for the three QoI against the results in the simulations in the test set. The color denotes the regime of the simulation in the test set (as defined in Section 7.4): Sheath-limited (blue), attached (orange), detached (green) and cold core (red).. The axis limits in B, E, H, K are adjusted to highlight the systematic shift in the predictions. Therefore the cold core cases with smaller heat fluxes are not seen. A model is more precise the closer the predictions are to the diagonal.

tached simulation, for which the NN2D+SOLPS approach predicts the correct peak heatflux. Like in all detached simulations, this case exhibits strong temperature gradients along the magnetic field lines towards the target. Consequently, there is a visible difference between the temperature profiles in the last (102) and second last poloidal row (101) of grid cells in front of the outer divertor target (Figure 8.29A). The NN2D model predicts the temperatures in both rows of grid cells accurately. For both the temperatures in the simulation and the temperatures predicted by the NN2D model, the poloidal temperature differences between the last and second last row of grid cells are calculated. While the model predicted temperature profiles are not perfect, they are accurate enough that the poloidal temperature differences are similar in the model predictions (Figure 8.29C) as in the simulations (Figure 8.29B). Because the temperature gradients are one of the key defining factors for the poloidal heat fluxes, also the resulting electron heat flux distributions resulting from the model predictions (Figure 8.29C) are alike the simulation (Figure 8.29B) in shape and magnitude. Figures 8.29D, E, F depict the conditions in a sheath-limited case, for which the NN2D+SOLPS approach predicts the peak heatflux too high. In this simulation the poloidal temperature gradients are so small that the temperature profiles in the two last rows of grid cells are almost identical (Figure 8.29D). Even though the temperature differences between the rows of grid cells are almost non existent, the overall high temperatures still yield a non-vanishing heat flux towards the target (Figure 8.29E). While the temperatures predicted by the NN2D network are accurate (Figure 8.29D), the poloidal temperature differences between the rows of grid cells are orders of magnitude higher than in the simulation and follow a random pattern (Figure 8.29F). One can see that the resulting electron heat fluxes always point in the direction of the temperature gradients and therefore randomly point towards or away from the target at the differing radial locations (Figure 8.29F). Because the absolute values of the temperature differences are drastically higher than in the simulations, also the heat fluxes in the NN2D+SOLPS approach have drastically increased magnitudes. These random patterns in the temperature gradients occur because the NN2D architecture predicts temperatures for each grid cell independently. As long as the errors in the predicted temperatures are far lower than the local temperature differences, this will automatically get the temperature gradients correct. Because the temperature gradients are large in the detached regime, the NN2D+SOLPS approach yields the correct heatfluxes in this regime. But in the sheath-limited regime, which has almost no temperature gradients, the local temperature differences between neighboring grid cells are tiny compared to the absolute temperature values present (here $\approx 0.001\,\mathrm{eV}/10^3\,\mathrm{eV} = 10^{-4}\%$). Because these temperature differences are far lower than the accuracy of the network predictions ($\approx 12\%$, see Table 8.6), the random errors in network predictions will lead to random directions and magnitudes of the temperature gradients and resulting heat fluxes. While the examples in Figure 8.29 demonstrate this phenomenon at the outer target, this behaviour occurs anywhere in the domain. The fact that the predictions of $P_{\mathrm{SOL}}$ in the NN2D-SOLPS scheme are less impaired by this behaviour has two reasons. Mainly because the radial temperature differences across the separatrix are larger and secondly because $P_{\mathrm{SOL}}$ is an integrated quantity, which allows for parts of the errors to average out. The peak heat flux $q_{\mathrm{peak,ot}}$, which is defined as the maximum heat flux at any target location, will be systematically increased by the large random heatflux spikes.

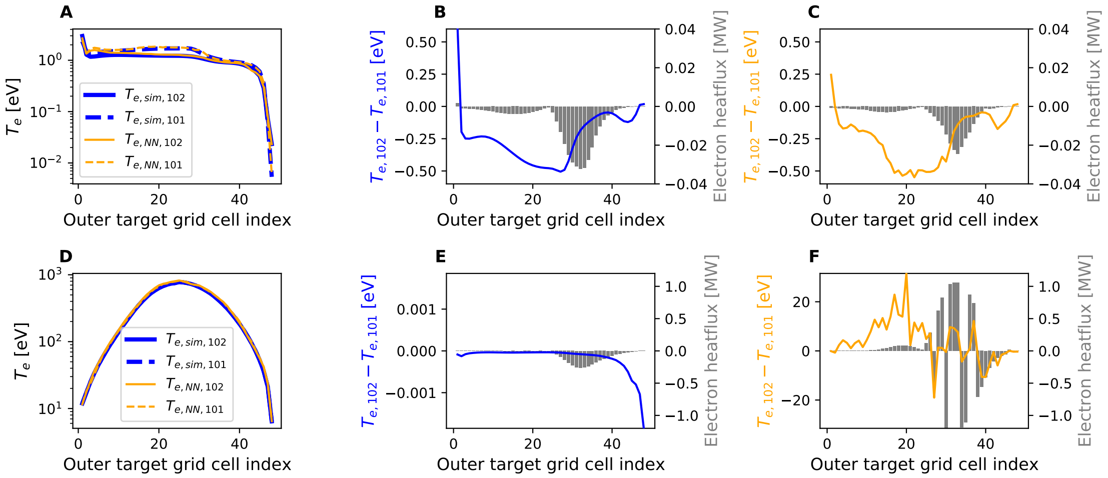


**Figure 8.29:** A: Electron temperatures in the last and second last row of grid cells in front of the outer divertor target for a test simulation in the detached regime. B: Poloidal temperature differences in the simulations between the two rows of grid cells and the resulting electron heat flux. C: Poloidal temperature differences in the NN2D predictions between the two rows of grid cells and the resulting electron heat flux. D, E, F show the same as the top row but for a test simulation in the sheath-limited regime.

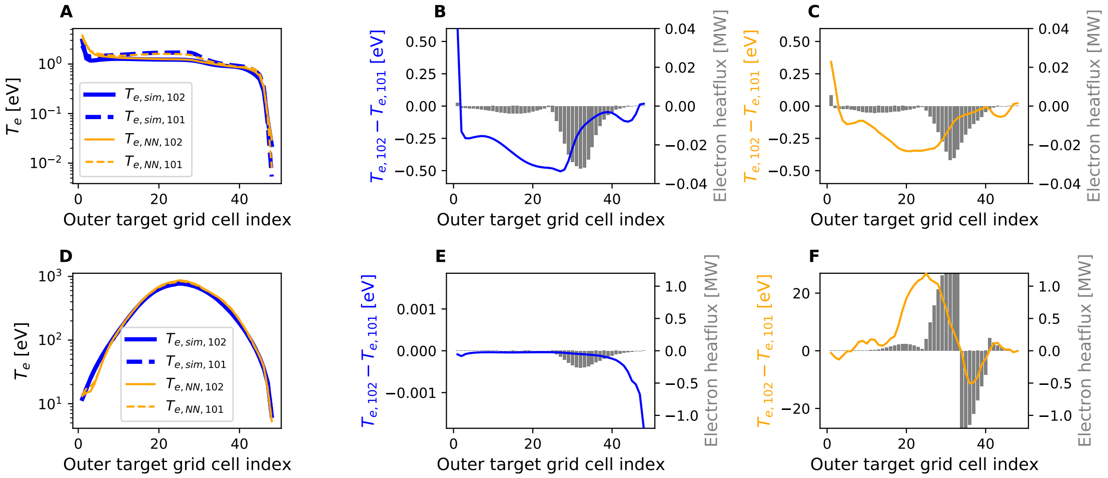


**Figure 8.30:** A: Electron temperatures in the last and second last row of grid cells in front of the outer divertor target for a test simulation in the detached regime. B: Poloidal temperature differences in the simulations between the two rows of grid cells and the resulting electron heat flux. C: Poloidal temperature differences in the NNpos2D predictions between the two rows of grid cells and the resulting electron heat flux. D, E, F show the same as the top row but for a test simulation in the sheath-limited regime.

Even if $q_{\mathrm{peak,ot}}$ were less affected, it is clear that the random heat flux patterns will cause problems for applications which require accurate heat flux profiles at the target, e.g. to calculate the sputtering of divertor target materials. Therefore, different approaches are tested in the following to reduce the random errors in the heat flux profiles.

One potential remedy lies in using a network model based on the NNpos2D architecture. The flaw of the NN2D model is that it does not necessarily conserve the spatial correlations between temperatures. In contrast the NNpos2D architecture is forced by its to design to learn a representation of spatial patterns. Therefore, similarly as in Section 8.5 a NNpos2D model is trained with multiple outputs. Here network contains 22 scalar outputs for all necessary SOLPS state variables. The model predictions are then used in the same procedure with SOLPS-ITER to create the auxiliary outputs (NNpos2D+SOLPS). The predictions on the test set for the three QoI are shown in Table 8.11 and Figure 8.28G, H, I. Compared to the NN2D+SOLPS procedure, the NNpos2D model leads to slightly lower median errors for $P_{\mathrm{SOL}}$ and also some improvement in the errors of $q_{\mathrm{peak,ot}}$ in the sheath-limited regime. But the NNpos2D+SOLPS model leads to slightly higher errors for $q_{\mathrm{peak,ot}}$ in attached and detached conditions. Also in Figure 8.28H it appears that the heat fluxes in attached and detached conditions are less accurately predicted than with the NN2D model in Figure 8.28E. Importantly, Figure 8.28H still shows a systematic shift of the predicted peak heat fluxes in sheath-limited and some attached simulations. Additionally, the heat fluxes in the sheath-limited regime are sometimes also predicted to low by the NNpos2D+SOLPS approach. Figure 8.30 depicts the same examples as Figure 8.29 but with the predictions by the NNpos2D model. Clearly the positional model removed the unnatural random spike patterns from the temperature differences (Figure 8.30F), but because the temperature predictions are slightly less accurate, the absolute magnitudes of temperature differences and heat fluxes are still wrong. So while the idea behind the positional model works, the low accuracy negates the effects such that this model does not pose an effective alternative.

Another idea is to use SOLPS-ITER itself to smooth out the mistakes in the temperature gradients. For this the NN2D predictions from the example in Figure 8.29D are taken, and instead of computing only the depending quantities, the SOLPS-ITER simulation is started from the network prediction for several timesteps. In this example the poloidal temperature differences at the outer target are only exactly recovered after progressing the SOLPS-ITER state for 0.1 s, although the target heat flux profile is already approaching the ground truth simulation results after 0.01 s (Figure 8.31). Therefore, this scheme is applied to the whole test set by iterating the NN2D predictions for 1000 timesteps of size $10^{-5}$ s (NN2D+1000xSOLPS). This approach has the disadvantage that it takes more computational time and given the many necessary iterations 4% of the simulations diverged before reaching this number of timesteps. In these cases the results from the NN2D+SOLPS model without forward progression in time are used as a fallback. The results of this approach on the test set are summarized in Table 8.11 and Figures 8.28J, K, L. Using this procedure, the errors for all three observables have improved compared to the plain

NN2D+SOLPS model. Specifically for the predictions of $q_{\mathrm{peak,ot}}$ this approach yields the highest accuracy in all regimes, even surpassing the direct network predictions. So while this approach serves as a usable remedy it is also partly cheating, because it requires running SOLPS for extensive periods of times. While the iterations required are still far lower than what is needed to arrive at the SOLPS-ITER solutions from scratch, it is unclear how many iterations are required for similar surrogate models constructed on a different SOLPS-ITER datasets with different numerical settings.

In conclusion: Deriving fluxes from the SOLPS-ITER state variables which are predicted by a surrogate, leads to mixed results. For the target particle flux this approach yields similar accuracy as when predicting the particle fluxes directly by a neural network. In contrast, the heat fluxes derived from model predictions are often drastically less accurate then the heat fluxes predicted directly by a network. Especially in the sheath-limited regime, the peak target heat flux is predicted systematically too large. These errors occur because the local temperature differences in the simulations are smaller than the accuracy of the network. In detached conditions with stronger temperature gradients, the derived fluxes are more precise. To derive the fluxes from the surrogate predictions, the SOLPS-ITER code can be used. Running SOLPS-ITER for multiple iterations starting from the surrogate predictions provides the most accurate predictions. But this comes with a decrease in the computational advantage gained by the surrogate model.

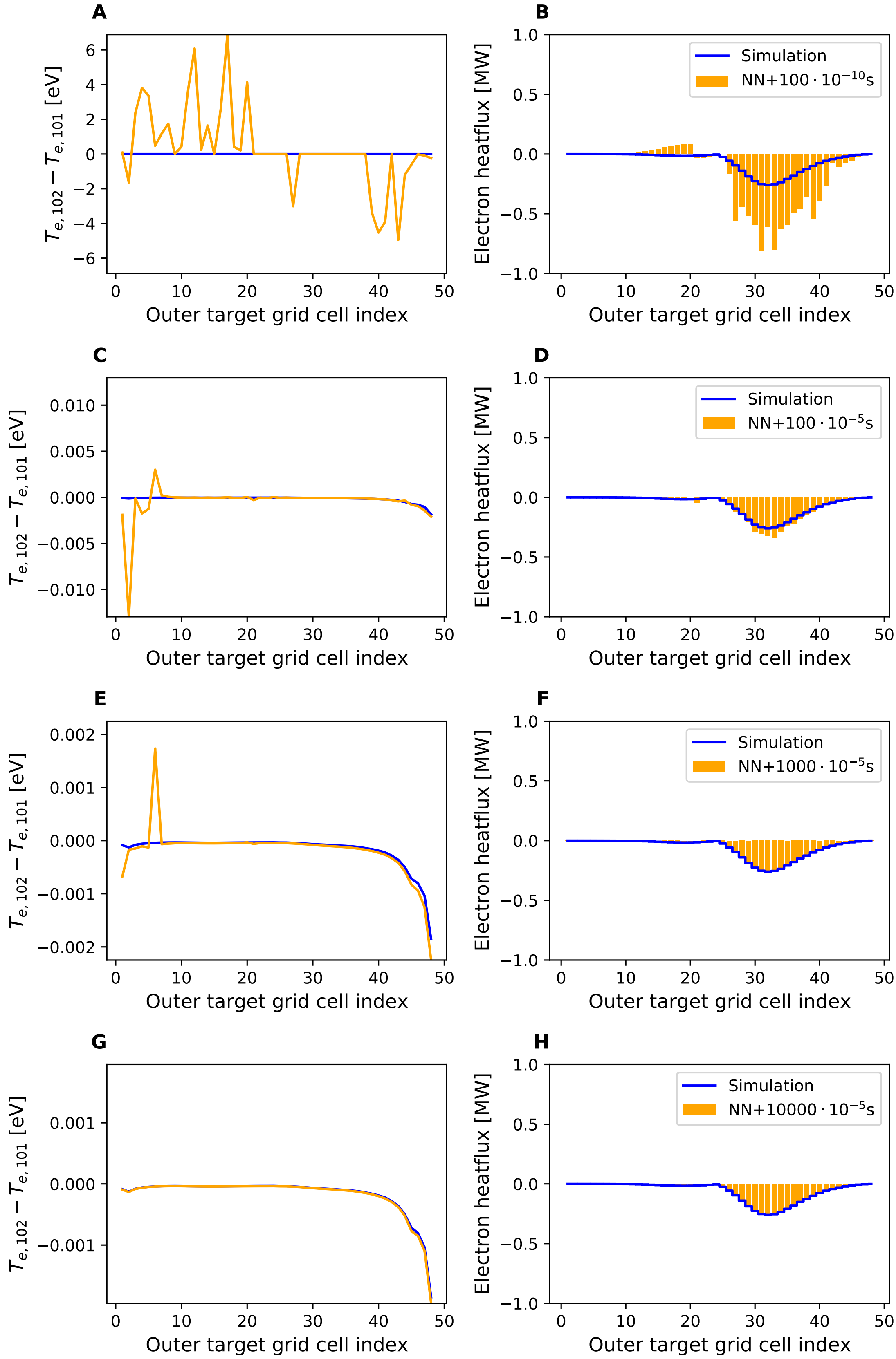


**Figure 8.31:** Each row compares the results between the simulation (blue) from the bottom row in Figure 8.29 and the NN2D predictions that are progressed forward for multiple timesteps with SOLPS-ITER (orange). The left plots show the poloidal temperature differences between the last two rows of grid cells in front of the outer target and the right show the resulting poloidal electron heat flux. The NN2D prediction is progressed by SOLPS-ITER for either $100 \cdot 10^{-10}$ s (A, B), $100 \cdot 10^{-5}$ s (C, D), $1000 \cdot 10^{-5}$ s (E, F) or $10000 \cdot 10^{-5}$ s (G, H).

## 8.7 Towards Physics Informed Neural Networks

Another idea to increase the reliability of the temperature gradients in the neural network predictions is to train a network to explicitly conserve them. This can be achieved by adding additional terms in the loss function during network training. The NN2D model is trained solely with the mean absolute error as loss (Equation 8.1). This loss (but also all other standard loss functions) forces the network to predict temperatures close to the temperatures in the simulations. But for the loss the predictions errors are calculated independently for each location, so it is indifferent to any spatial correlations in the temperatures. The loss function can be easily modified to include additional terms (Equation 8.2). This modified loss contains two extra terms which correspond to the mean absolute error between the predicted and simulated temperature differences between neighbouring grid cells. Two scaling factors are included, which can be chosen freely to modify the strength of these additional loss terms in comparison to the mean absolute error. $\alpha$ scales the influence of the poloidal temperature differences and $\beta$ the strength of the radial temperature differences. Equation 8.2 is meant to highlight the idea behind these extra loss terms. In practice the implementation of this loss is slightly more complicated, because in the complex SOLPS-ITER grid geometry the $i$-th and $i+1$-th row of grid cells are not always adjacent. So the implementation of the loss needs to perform the correct jumps at the correct places, such that the temperature differences are always calculated between neighbouring grid cells.

Another concern arises because the temperatures are scaled with the quantile transformation functions independently in each grid cell prior to training. Since the loss is applied on the scaled temperatures, it will force the network to preserve the temperature differences in this scaled space. Because the backtransformation to eV-scale is non-linear and different for each grid cell, the temperature gradients are not necessarily conserved after this transformation. But inspection of the quantile transformation functions for neighbouring grid cells shows that these are similar because neighbouring grid cells have similar temperature statistics. So even though the backtransformation to eV-scale might distort the temperature gradients a little, the modified loss might still work sufficiently well.

$$\mathrm{MAE} = \frac{1}{N_{\mathrm{samples}}} \sum_{\mathrm{samples}} \sum_{i=1}^{N_{pol}} \sum_{j=1}^{N_{rad}} \left| T_{i,j}^{sim} - T_{i,j}^{pred} \right| \tag{8.1}$$

$$\mathrm{MOD} = \frac{1}{N_{\mathrm{samples}}} \sum_{\mathrm{samples}} \begin{cases} & \sum_{i=1}^{N_{pol}} \sum_{j=1}^{N_{rad}} \left| T_{i,j}^{sim} - T_{i,j}^{pred} \right| \\ + & \alpha \sum_{i=2}^{N_{pol}} \sum_{j=1}^{N_{rad}} \left| (T_{i,j}^{sim} - T_{i-1,j}^{sim}) - (T_{i,j}^{pred} - T_{i-1,j}^{pred}) \right| \\ + & \beta \sum_{i=1}^{N_{pol}} \sum_{j=2}^{N_{rad}} \left| (T_{i,j}^{sim} - T_{i,j-1}^{sim}) - (T_{i,j}^{pred} - T_{i,j-1}^{pred}) \right| \end{cases} \tag{8.2}$$

Therefore, several versions of the NN2D network model were trained with the modified loss function (Equation 8.2) to predict the temperatures. The versions differ in the chosen values for $\alpha$ and $\beta$, the chosen learning rates and the initial network weights. In half of the attempts, the network training is started from scratch and in the other half from a version of the NN2D network pretrained solely with the standard MAE (Equation 8.1) loss. But none of these attempts brought the hoped improvements. Figure 8.32 depicts the mean absolute errors of the poloidal temperature difference and of the temperature at the outer target separatrix. For low

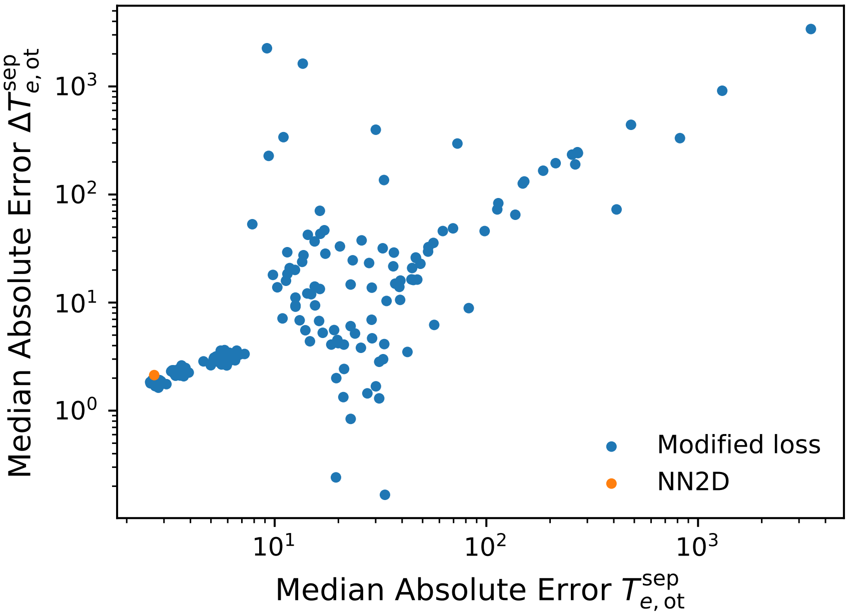


**Figure 8.32:** Median absolute test errors of the poloidal temperature difference at the outer target separatrix against the median absolute test error of the temperature at the outer target separatrix for NN2D models trained with variations of the modified loss (Equation 8.2) or the NN2D model from Section 8.3.

values of $\alpha$ and $\beta$, the models trained with the modified loss obtain similar accuracy on the absolute values of the temperatures as the standard NN2D model, but then no improvement in the temperature differences is obtained. While some attempts with large $\alpha$ and $\beta$ yield drastically smaller errors in the temperature differences, these models come with increased errors for the absolute temperature values, negating the overall usefulness.

More work is required to determine whether other implementations of physics informed loss functions offer improvements beyond the standard NN2D model. One such approach could also be to train models to predict temperature differences instead of the temperatures themselves.

# Chapter 9

# Applications of the surrogate model

The focus in this thesis lies in exploring and developing methods for the creation of surrogate models of the SOL. While the developed surrogate models are primarily proof-of-concepts, compared with the alternative SOL models which are currently available (Section 1.4) already these surrogate models could prove useful. Therefore, the aim of this chapter is to showcase and test some of the applications of this surrogate model. This includes specifically an analysis what physical predictions are made by this model, which will answer the question which effects are included in the fluid neutral dataset.

## 9.1 Inverse application

As introduced in Section 2.5, the strength of cross-field transport is not treated self-consistently in SOL transport simulations but values for effective transport coefficients need to be provided by the modeler. In SOL modeling the most common approach is to take experimental measurements of the plasma at the outer midplane and then vary the transport coefficients until the simulation results match the experimental profiles [84]. Solving this inverse problem can be drastically simplified with a surrogate model as developed here. Because the surrogate model takes the transport coefficients as input and provides the resulting plasma profiles orders of magnitudes faster than simulations, it is possible to try out many combinations of values for the transport coefficients. Especially since the neural network can be executed on a GPU which allows processing of many inputs at once. Alternatively since neural networks are end-to-end differentiable and build on frameworks with automatic differentiation (e.g. PyTorch, TensorFlow, JAX), one can use any gradient based optimization algorithm to optimize the transport coefficients. The optimization algorithm can then be tasked to find the transport coefficients which minimize the difference between the model output and a predefined profile.

To demonstrate these two approaches the developed surrogate model for the temperature (NN2D-$T_e$) was used to make a prediction for the temperature profile at the outer midplane. Then both approaches are applied to try and recover the transport coefficients used in this initial prediction. In both approaches this comes down to finding the transport coefficients that minimize the mean absolute error between

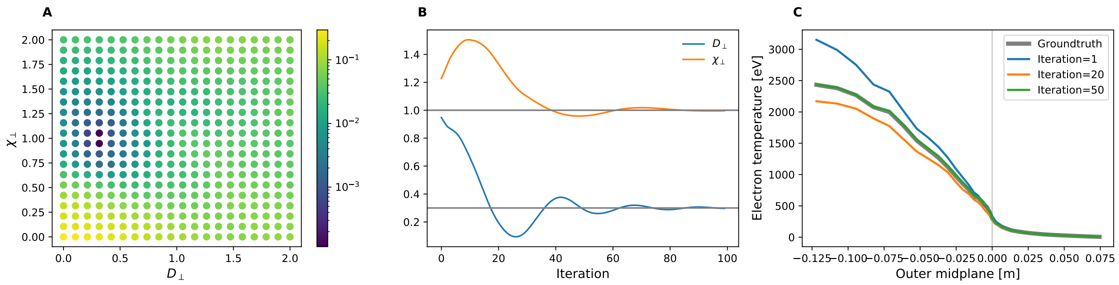


**Figure 9.1:** Applying the surrogate model to find the transport coefficients matching a predefined temperature profile. A: Mean absolute errors between model predicted temperatures at the outer midplane and a predefined profile for varying transport coefficients used in the model input. B: Progression of the transport coefficients over iterations during gradient based optimization minimizing the differences between model prediction and predefined temperature profile. The grey lines denote the transport coefficients used in the predefined profile. C: The predefined temperature profile at the outer midplane (grey) and the model predictions after 1 (blue), 20 (orange) and 50 (green) iterations of the gradient based optimization.

the model prediction and predefined temperature profile. Figure 9.1A depicts these mean absolute errors for a variation of tested $\chi_\perp$ ad $D_\perp$. It is clear that the best fit is obtained around $\chi_\perp = 1.0\,\mathrm{m}^2/\mathrm{s}$ and $D_\perp = 0.3\,\mathrm{m}^2/\mathrm{s}$, which are the values used for the initial reference profile. This approach also has the benefit that besides receiving just the final values one can estimate a range of acceptable transport coefficients. Figure 9.1B shows the progression of the transport coefficients over several iterations during a gradient based optimization. Also here the algorithm is able to find the correct transport coefficients. Figure 9.1C depicts the changes to the temperature profile at the outer midplane during this optimization until the reference profile is obtained.

Of course in real applications, the transport coefficients need to be found which reproduce experimental measurements. Those scenarios are more difficult because it is unlikely that a perfect match can be obtained. But the same problem also applies to any such procedure using simulations instead of a surrogate model. The applicability of surrogate models for such inverse problems is not just restricted to the transport coefficients. Rather the surrogate model can be used for optimization of any machine design or operation parameter based on a desired plasma property anywhere in the domain. As only requirement, the parameters need to be input quantities of the surrogate model and the desired plasma profiles need to be derived from the surrogate output in a differentiable way.

## 9.2 Surrogate initializer

The analysis in Section 8.4 has shown that the accuracy of the surrogate model could be further increased by an inclusion of more simulations. Especially for future surrogate models that include more computationally expensive higher fidelity simulations it would be beneficial if the compute time of each additional simulation could be minimized. Since SOLPS-ITER simulations iterate over a (pseudo-)timestep until a convergence of the plasma properties is reached, the simulation runtime could potentially be reduced if the surrogate model prediction is used as the initial plasma state in the simulations. But given the findings in Section 5 that apparently "closer" initial plasma states lead to a higher likelihood of simulations to diverge, it is not guaranteed that using the surrogate prediction as initial state leads to shorter simulation runtimes. In Section 8.6 SOLPS-ITER was started from surrogate predictions and it appeared that the simulations approach the converged results faster than was initially necessary for these simulations.

To test this procedure further 512 points are randomly distributed in the same 8 dimensional input parameter space as the training data (see Section 6). For each parameter configuration two simulations are started for a maximum of $10^5$ timesteps. One from the default initial state with uniform densities and temperatures in the whole domain and one from the prediction by the NN2D surrogate models that include all state variables (Section 8.6). As expected this leads to mixed results. Figure 9.2 shows the temporal progression of electron density at the outer midplane separatrix in four examplary cases, that depict the possible outcomes. In Figure 9.2A the predicted density by the surrogate model is very close to the final density that is approached by the simulation starting from the uniform state. But the simulation started from the surrogate prediction diverges and yields no result. In Figure 9.2B both simulations converge to the same upstream density. Since the surrogate prediction was already very close to this density, this is reached much faster than in the simulation started from the uniform state. In Figure 9.2C only the simulation started from the surrogate prediction obtains a result, while the simulation started from the uniform state diverges. In Figure 9.2D both simulations seem to approach converged conditions (the actual convergence metrics are not yet fulfilled) but with different upstream densities. It is likely that with more strict convergence metrics only one of the solutions proves valid or that both will converge to a common solution provided more runtime. More testing and analysis is required to determine what properties a surrogate based initial state needs to fulfill such that it leads to consistently faster SOLPS-ITER simulations.

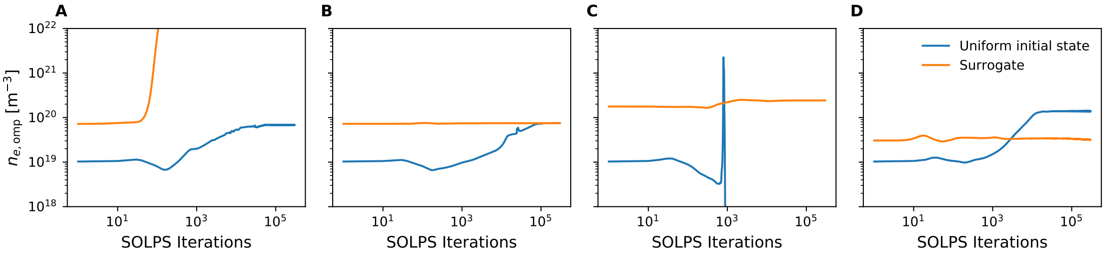


**Figure 9.2:** Time traces of the outer midplane separatrix electron density $n_{e,\mathrm{omp}}$ in four exemplary fluid neutral SOLPS-ITER simulations that were either started from the default initial state with uniform densities and temperatures (blue) or from predictions of the NN2D surrogate as initial state (orange). The examples are chosen to highlight the different situations that can occur and are not converged.

## 9.3 Detachment scaling

The most straight forward area of application for the surrogate model is to make predictions for the conditions in varying tokamaks. Therefore, this sections analyses the predictions made by the surrogate. A special focus is to clarify which aspects of detachment are seen in this model. This is especially interesting because it has shown difficult to analyse such a high dimensional simulation dataset since always multiple parameters are varied at once (Section 7.4). With the trained surrogate model it is now possible to vary all parameters independently. The most pressing question for divertor design is which impurity concentration and plasma density are necessary to reach detachment [64], [78], [109], [187]. This question is so critical because it decides whether a safe divertor operation is compatible with the requirements for the confined plasma. Due to the existence of density limits beyond which the plasma becomes unstable [32], the plasma density cannot be increased unlimited to reach detachment. Yet the amount of impurities that are introduced should be as minimal as possible as higher levels reduce the possible fusion power [187], [200]. Additionally, the impurity atoms contribute greatly to the erosion of divertor and reactor wall components [113]. To estimate the impurity concentrations necessary for detachment across tokamaks, several empirical or semi-analytical scaling relations have been developed [71]–[73]. We can use the surrogate model to answer the same question. For this the models from Section 8.6 are used. The three direct prediction networks for the power crossing the separatrix $P_{\mathrm{SOL}}$, the peak heat flux at the outer divertor target $q_{\mathrm{peak,ot}}$ and the integrated deuterium ion flux at the outer divertor target $\Gamma_{D+,\mathrm{ot}}$ and the NN2D models for plasma densities, velocities and temperatures. With these models predictions are performed for a parameter scan across four tokamaks: ASDEX Upgrade (AUG), JET, ITER and DEMO. The key defining input parameters for the neural network, which represent each tokamak, are given in Table 9.1 (the values are taken from [73]). In addition as input parameter for all four tokamaks, the transport coefficients $\chi_\perp = 1.0\,\mathrm{m}^2/\mathrm{s}$ and $D_\perp = 0.3\,\mathrm{m}^2/\mathrm{s}$ and a core deuterium ion influx of $D_{\mathrm{core}} = 10^{20}$ atoms/s are chosen. For each of the four tokamaks the deuterium gas puff and nitrogen gas puff are then varied between the minimum and maximum values in the simulation dataset (Section 6) in 200 equidistant steps (on a logarithmic scale).

| | AUG | JET | ITER | DEMO |
|---|---|---|---|---|
| $R$ | 1.6 m | 2.9 m | 6.2 m | 9.1 m |
| $B$ | 2.5 T | 2.5 T | 5.3 T | 5.7 T |
| $P_{\text{in}}$ | 10.7 MW | 14 MW | 100 MW | 154.7 MW |
| $n_{GW}$ | $1.44 \cdot 10^{20}\,1/\text{m}^3$ | $9.82 \cdot 10^{19}\,1/\text{m}^3$ | $1.19 \cdot 10^{20}\,1/\text{m}^3$ | $7.39 \cdot 10^{19}\,1/\text{m}^3$ |

**Table 9.1:** Input parameters of the surrogate model $R$, $B$ and $P_{\text{in}}$ used to represent the four tokamaks in the parameter scan. Thes parameter values as well as the Greenwald densities $n_{GW}$ [32] are taken from Table 2 in [73]. $P_{\text{in}}$ describes the expected power entering the domain modeled in SOLPS-ITER and the surrogate model.

Surrogate predictions are performed for all possible combinations of the two gas puffs values, which results to 4000 scenarios for each tokamak.

The results of these parameter scans are summarized in Figures 9.3 (AUG), 9.4 (JET), 9.5 (ITER), 9.6 (DEMO). The key upstream plasma quantities are the electron density $n_{e,\text{omp}}$ and nitrogen concentration $c_{N,\text{omp}}$ at the outer midplane separatrix because these are the limiting condition what can be tolerated for the confined plasma. The nitrogen impurity concentration is calculated by dividing the summed up densities of all nitrogen ions and neutral gas particles by the deuterium ion density $c_N = \frac{n_N + n_{N^+} + \ldots + n_{N^{7+}}}{n_{D^+}}$. To assess which criteria for detachment and safe divertor operation are fulfilled at the outer target, the electron temperature outside the separatrix $T_{e,\text{ot}}$, the peak heat flux $q_{\text{peak,ot}}$ and the integrated deuterium ion flux $\Gamma_{D+,\text{ot}}$ are inspected.

For all tokamaks the parameter scans show similar patterns. In all four tokamaks a large part of the parameter space in the bottom left corner, which corresponds to low gas puff values, is in the sheath-limited regime. There the plasma density is low (Figures 9.3-9.6A) and the temperatures at the outer target very high (Figures 9.3-9.6C). The orange and red lines in Figures 9.3-9.6A-E highlight the transition into the sheath-limited and cold core regimes based on the classification introduced in Section 7.4. As indicated by the lines, only a narrow band of gas puff values lies in the attached or detached conditions. For ITER and DEMO, the criterion for separating sheath-limited from attached cases shows that some gas puff values inside the otherwise sheath-limited domain show strong enough temperature gradients that the classification would rule them as attached although the outer target temperatures are still above 1000 eV (bottom left in Figures 9.5,9.6C). Therefore, it is likely that a stricter criterion (e.g. $T_{e,\text{ot}}/T_{e,\text{omp}} < 0.6$) would provide a more accurate split of these two regimes. For AUG and JET, either a very high deuterium gas puff or a high nitrogen gas puff are enough to bring the SOL into the cold core regime, which is seen as large area above and to the right of the red line in Figures 9.3,9.4A-E. For ITER and DEMO, the same can expected but necessary deuterium gas puffs to achieve the cold core regime are beyond the tested range. Therefore, the cold core regime is only obtained in ITER for a slim region of very high nitrogen gas puffs $> 8 \cdot 10^{22}$ atoms/s and in the top right corner where both gas puffs are large (Figure 9.5A-E). In DEMO the regime is only obtained in this top right corner with large gas puffs (Figure 9.6A-E). The attached and detached scenarios lie in a narrow band that stretches from the bottom right to the top left (Figures 9.3-9.6A-E).

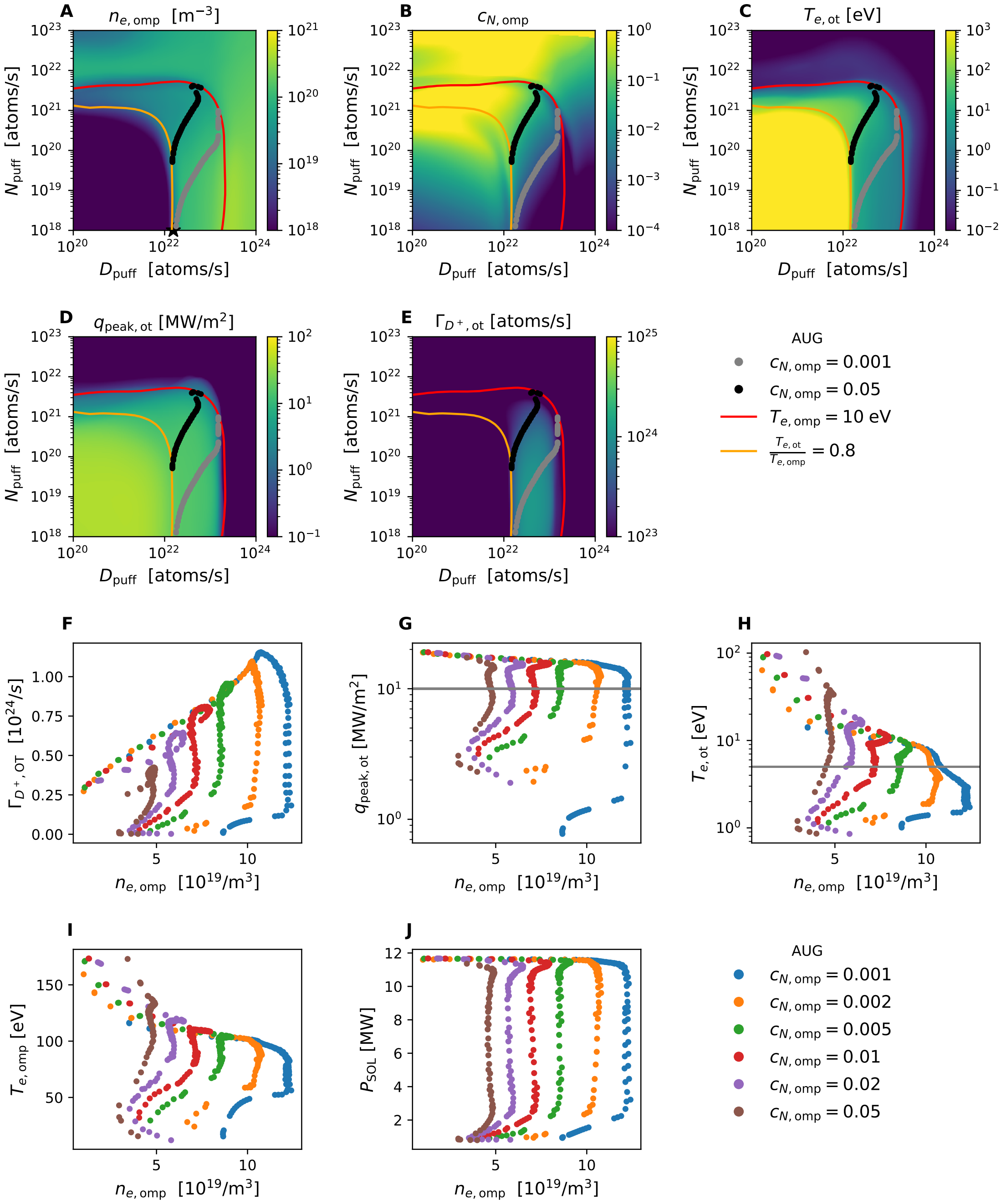


**Figure 9.3:** Predictions of the surrogate model for a gas puff scan with **AUG** parameters. Shown are the resulting electron density $n_{e,\mathrm{omp}}$ (A) and nitrogen concentration $c_{N,\mathrm{omp}}$ (B) at the outer midplane separatrix and the outer target temperature $T_{e,\mathrm{ot}}$ (C) at the separatrix as well as the integrated deuterium ion flux $\Gamma_{D^+,\mathrm{ot}}$ (D) and peak heat flux $q_{\mathrm{peak,ot}}$ (E) towards the outer target. In A-E the the orange and red lines highlight the transition into the sheath-limited and cold core regimes while the grey and black dots show all scenarios with nitrogen concentrations $c_{N,\mathrm{omp}} = 0.1\%$ and $c_{N,\mathrm{omp}} = 5\%$. The black star in A highlights the deuterium gas puff necessary to leave the sheath-limited regime with minimum nitrogen puff. F-J show density scans selected from the gas puff scan above. Scenarios are selected based on the upstream nitrogen concentration (denoted by the color) and when they are not in a sheath-limited or in the cold core regime. Shown are the integrated deuterium ion flux $\Gamma_{D^+,\mathrm{ot}}$ (F) and peak heat flux $q_{\mathrm{peak,ot}}$ (G) towards the outer target, the electron temperature at the outer target $T_{e,\mathrm{ot}}$ (H) and outer midplane $T_{e,\mathrm{omp}}$ (I) separatrix and the power crossing the separatrix $P_{SOL}$ (J) from the main plasma.

But in this band only the lower right part constitute desirable plasma scenarios as in the top left section the plasma contains more nitrogen particles than deuterium ions $c_N > 1$. For ITER the expected tolerable nitrogen concentration at the separatrix to obtain the design goal of $Q > 10$ lies below 1% (assuming a beryllium first wall) [200]. But a complete assessment of allowed impurity concentrations needs to consider the sputtering effects of the impurities at the divertor and wall materials [207]. To highlight the reasonable range of impurity concentrations, in Figures 9.3-9.6A-E the scenarios with upstream nitrogen concentrations of 0.1% and 5% are marked by grey and black points. It shows that scenarios with similar nitrogen concentration lie on almost diagonal lines. Moving from one scenario to another on such a line corresponds to changes in the plasma density or neutral particle content. Another common feature between all tokamaks is that inside the band of attached and detached scenarios the target electron temperature rises at the top right corner where the transition from a primarily deuterium plasma to a nitrogen dominated plasma occurs (Figures 9.3-9.6C).

From these parameter scans, scenarios with constant upstream nitrogen concentration can be extracted. These constitute density scans controlled by the gas puff values. The extracted scans only consider scenarios with $T_{e,\mathrm{ot}}/T_{e,\mathrm{omp}} < 0.6$ and $T_{e,\mathrm{omp}} > 10\,\mathrm{eV}$ to stay solely in the attached and detached regimes. The resulting scans for each tokamak are shown in the Figures 9.3-9.6F-J. For all tokamaks a clear rollover of the ion particle flux to the outer target is seen (Figures 9.3-9.6F). Across all tokamaks the upstream density at which the rollover occurs decreases with increasing impurity concentration. For a given impurity concentration the rollover occurs in each tokamak at a different plasma density. The obtained decrease in deuterium ion flux after the rollover is different for each tokamak. In AUG and JET also for the lowest impurity concentration shown here $c_{N,\mathrm{omp}} = 0.1\%$ the deuterium ion flux to the outer target can be decreased to less than 10% of its maximum value (Figures 9.3-9.4F), while for ITER and DEMO the reduction at the maximum density only reaches $\approx$ 80% of the maximum value (Figures 9.5-9.6F). It is very likely that a higher reduction would be achieved at higher plasma densities. These densities are not seen for ITER and DEMO because the attached/detached regimes at low impurity concentration extend beyond the tested maximum deuterium gas puff of $10^{24}$ atoms/s (Figures 9.5-9.6B). Therefore, the tested density range could be extended for ITER and DEMO, while for AUG and JET the maximum density is given at all impurity concentrations by the transition into the cold core regime. The predictions for all tokamaks have in common that at some point the upstream density stagnates and does not further increase with increasing gas puffs. At some point the upstream density then even starts to decrease. This phenomena is seen also in other SOLPS simulations [131], [200]. For JET, ITER and DEMO this stagnation only occurs in deeper detachment after the ion flux rollover. For ITER and DEMO it can thus only be seen in the scenarios with higher impurity concentration. Interestingly, this stagnation of the upstream density happens in AUG directly after the rollover (Figure 9.3F). For $c_{N,\mathrm{omp}} = 0.1\%$ and $c_{N,\mathrm{omp}} = 0.2\%$ there is thus only a small period of increasing density after the maximum target ion flux before the density stagnates. But for higher impurity concentrations the rollover occurs as a roll backwards. Once the maximum target ion flux is reached, the density first decreases a little and then stagnates until at a later stage it starts to decrease again.

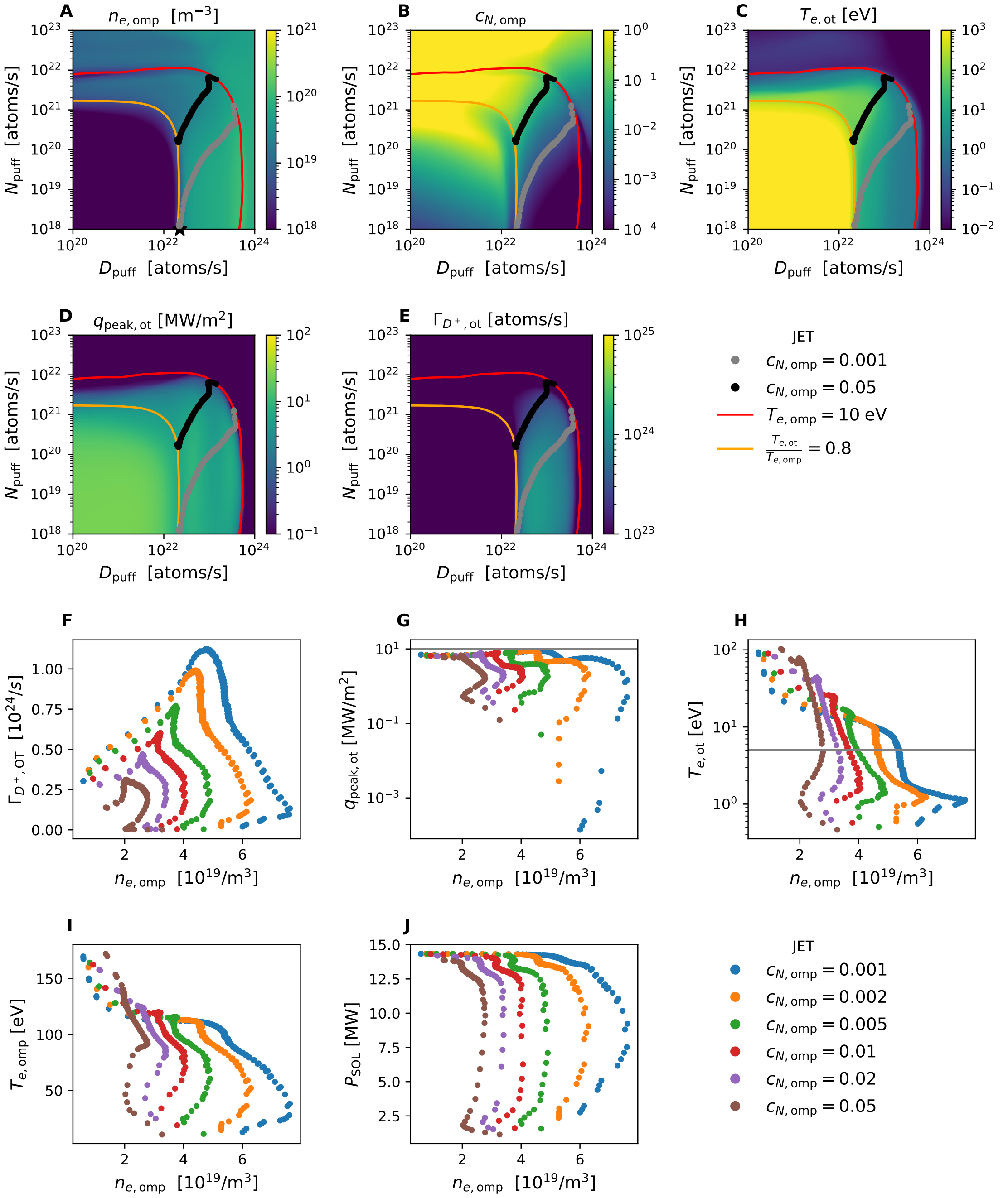


**Figure 9.4:** Predictions of the surrogate model for a gas puff scan with **JET** parameters. Shown are the resulting electron density $n_{e,\mathrm{omp}}$ (A) and nitrogen concentration $c_{N,\mathrm{omp}}$ (B) at the outer midplane separatrix and the outer target temperature $T_{e,\mathrm{ot}}$ (C) at the separatrix as well as the integrated deuterium ion flux $\Gamma_{D^+,\mathrm{ot}}$ (D) and peak heat flux $q_{\mathrm{peak,ot}}$ (E) towards the outer target. In A-E the the orange and red lines highlight the transition into the sheath-limited and cold core regimes while the grey and black dots show all scenarios with nitrogen concentrations $c_{N,\mathrm{omp}} = 0.1\%$ and $c_{N,\mathrm{omp}} = 5\%$. The black star in A highlights the deuterium gas puff necessary to leave the sheath-limited regime with minimum nitrogen puff. F-J show density scans selected from the gas puff scan above. Scenarios are selected based on the upstream nitrogen concentration (denoted by the color) and when they are not in a sheath-limited or in the cold core regime. Shown are the integrated deuterium ion flux $\Gamma_{D^+,\mathrm{ot}}$ (F) and peak heat flux $q_{\mathrm{peak,ot}}$ (G) towards the outer target, the electron temperature at the outer target $T_{e,\mathrm{ot}}$ (H) and outer midplane $T_{e,\mathrm{omp}}$ (I) separatrix and the power crossing the separatrix $P_{SOL}$ (J) from the main plasma.

Such a behaviour is not normally seen and further analysis is necessary to explore the mechanisms behind it.

For all tokamaks, the peak heat flux at the outer target $q_{\mathrm{peak,ot}}$ stays constant at first and then decreases after the rollover of the target ion flux (Figures 9.3-9.4G). For AUG and ITER the safe level of $10\,\mathrm{MW/m^2}$ can be reached for all displayed impurity concentrations (Figures 9.3G, 9.5G). For lower impurity concentrations this requires higher densities. For JET the target heat fluxes are always below the safe level (Figures 9.4G). In all cases except for JET, the heat fluxes decrease rather monotonously after the rollover. For lower nitrogen concentrations in JET, the heat fluxes decrease directly after the rollover, but afterwards remain constant and even slightly increase for a range of densities (e.g blue and orange curves in Figures 9.4G). This occurs because in these cases the location of the peak heat flux changes from slightly outside the separatrix to the lower edge of the target in the PFR (see Section 7.5). It is questionable whether the heat flux values at this boundary are reliable, but since these are the peak values anywhere on the target in these simulations, the simulations remain below the safe limit. For DEMO this limit is only achieved for nitrogen concentrations above 2% (Figure 9.6G). For lower impurity concentrations this can likely be achieved if the deuterium gas puff would be increased beyond the range tested here.

For all tokamaks, the temperatures at the outer target and the outer midplane decrease with increasing the upstream density and later in deep detachment also with decreasing plasma density (Figures 9.3-9.6H, I). In all tokamaks, the functional dependence between the plasma temperature at the target on the upstream density changes strongly at the point of particle flux rollover. After the rollover, the target temperature decreases with a higher gradient than before the rollover. In AUG the functional relation of the temperatures at the outer midplane and the outer target on the density are very similar (Figure 9.3H, I). Therefore, the decrease in target temperature is likely stemming majorly from the decrease in upstream temperatures. Both in AUG as in JET, the power crossing the separatrix stays only constant until the particle flux rollover occurs (Figures 9.3-9.4J). After that $P_{\mathrm{SOL}}$ decreases continuously down to less than 15% of the initial values. For ITER and DEMO (Figures 9.5-9.6J), the power crossing the separatrix shows a similar dependence, but the decrease is a lot less significant and $P_{\mathrm{SOL}}$ stays above 90% of the initial values. Only when the deep detachment is reached, where the upstream densities decrease, also $P_{\mathrm{SOL}}$ decreases strongly.

In contrast to other reduced models, the observations characterizing the plasma states in the previous figures are not just scalar values, but behind each one is a full 2D prediction of the corresponding plasma state in the SOL. Figure 9.7 depicts the SOL plasma profiles of the JET density scan with $c_{N,\mathrm{omp}} = 0.1\%$ and $D_{\mathrm{core}} = 10^{20}$ atoms/s. Each row depicts the plasma profiles corresponding to the marked position in the density scan on the left. At low upstream density, the plasma density in the whole domain is low and consequently the plasma temperature is high. As expected, the inner target detaches earlier. At maximum deuterium ion flux to the outer target, the temperature in the SOL close to the inner target has already decreased and the electron density in front of the inner target is elevated.

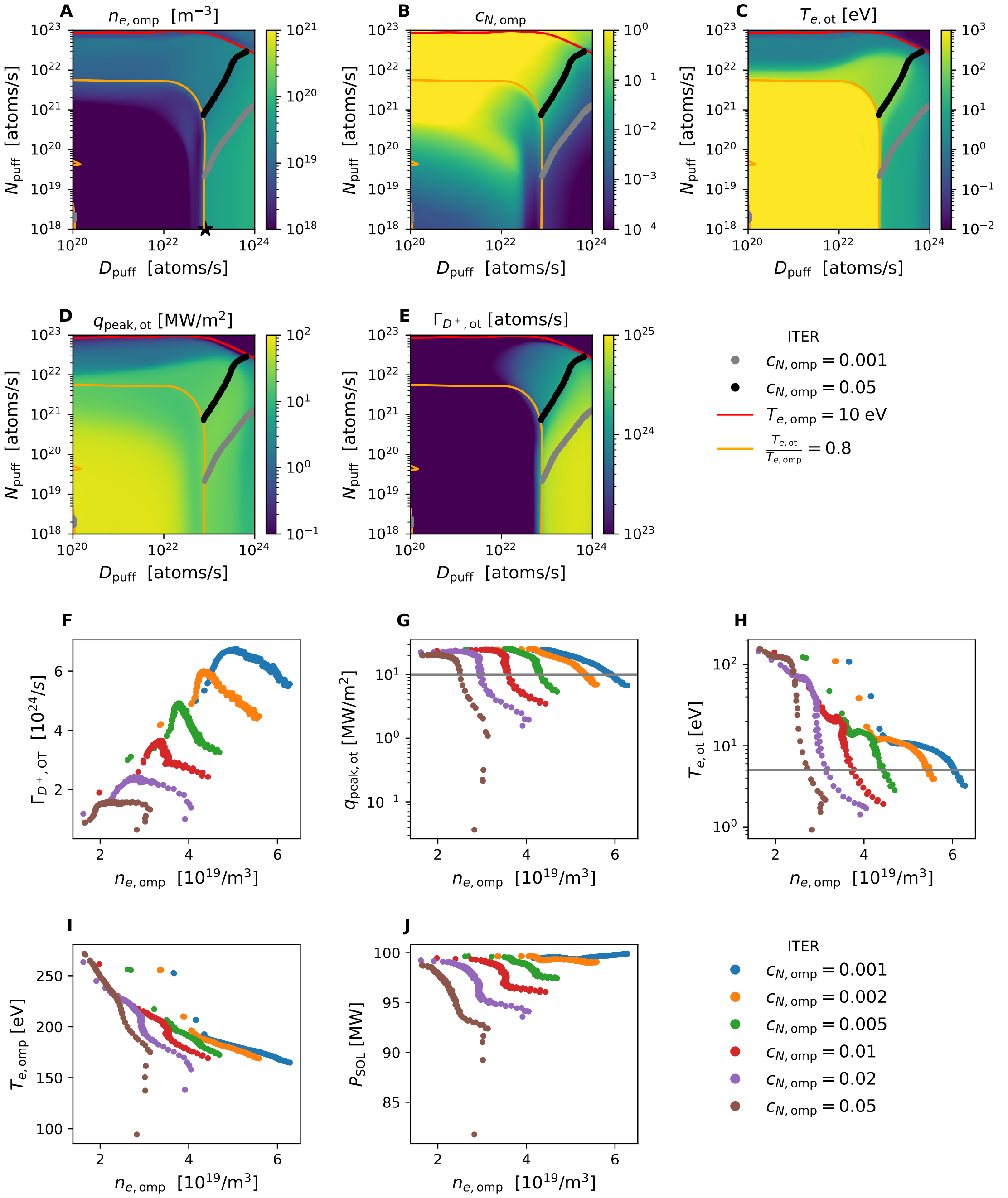


**Figure 9.5:** Predictions of the surrogate model for a gas puff scan with **ITER** parameters. Shown are the resulting electron density $n_{e,\mathrm{omp}}$ (A) and nitrogen concentration $c_{N,\mathrm{omp}}$ (B) at the outer midplane separatrix and the outer target temperature $T_{e,\mathrm{ot}}$ (C) at the separatrix as well as the integrated deuterium ion flux $\Gamma_{D^+,\mathrm{ot}}$ (D) and peak heat flux $q_{\mathrm{peak,ot}}$ (E) towards the outer target. In A-E the the orange and red lines highlight the transition into the sheath-limited and cold core regimes while the grey and black dots show all scenarios with nitrogen concentrations $c_{N,\mathrm{omp}} = 0.1\%$ and $c_{N,\mathrm{omp}} = 5\%$. The black star in A highlights the deuterium gas puff necessary to leave the sheath-limited regime with minimum nitrogen puff. F-J show density scans selected from the gas puff scan above. Scenarios are selected based on the upstream nitrogen concentration (denoted by the color) and when they are not in a sheath-limited or in the cold core regime. Shown are the integrated deuterium ion flux $\Gamma_{D^+,\mathrm{ot}}$ (F) and peak heat flux $q_{\mathrm{peak,ot}}$ (G) towards the outer target, the electron temperature at the outer target $T_{e,\mathrm{ot}}$ (H) and outer midplane $T_{e,\mathrm{omp}}$ (I) separatrix and the power crossing the separatrix $P_{SOL}$ (J) from the main plasma.

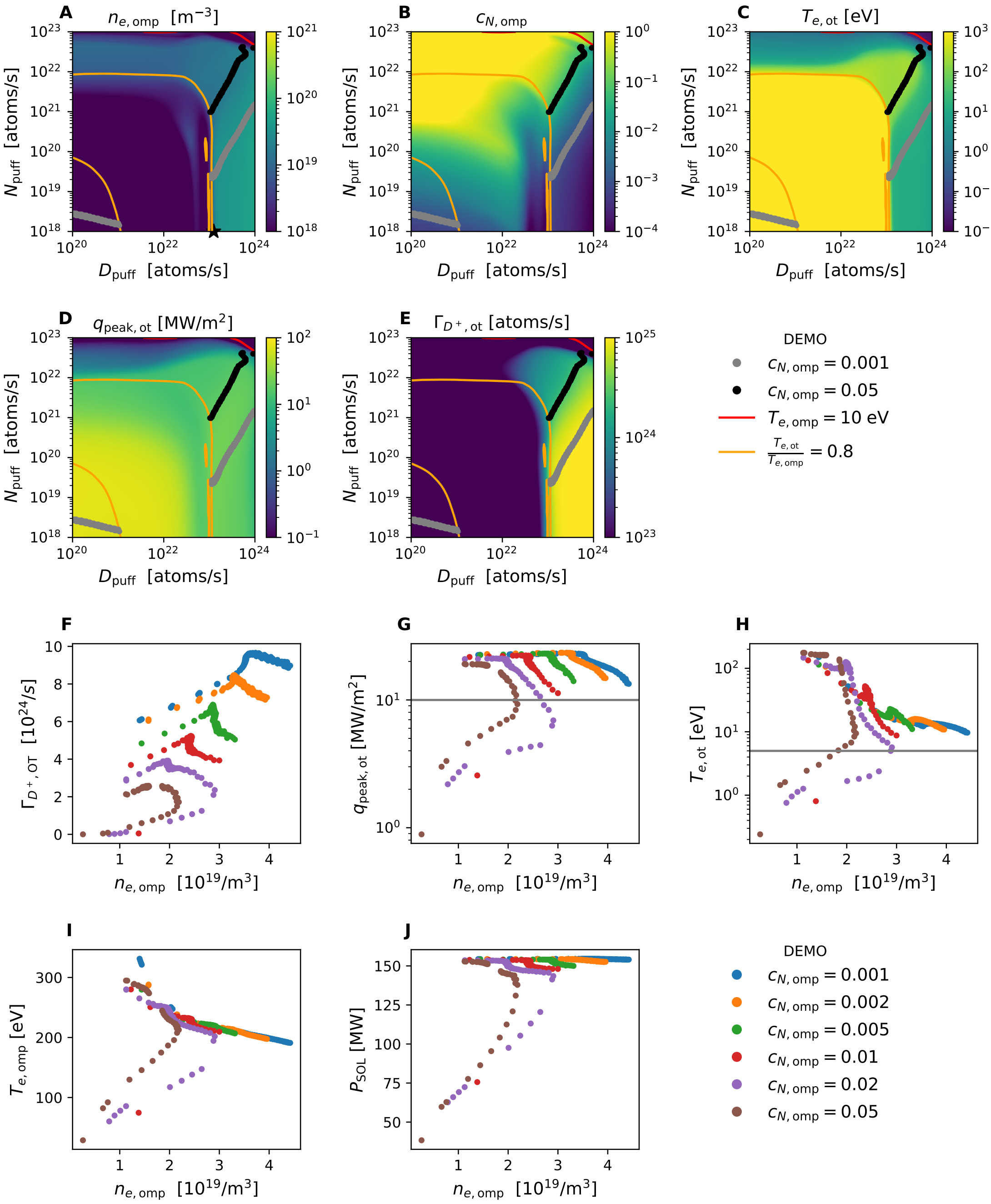


**Figure 9.6:** Predictions of the surrogate model for a gas puff scan with **DEMO** parameters. Shown are the resulting electron density $n_{e,\mathrm{omp}}$ (A) and nitrogen concentration $c_{N,\mathrm{omp}}$ (B) at the outer midplane separatrix and the outer target temperature $T_{e,\mathrm{ot}}$ (C) at the separatrix as well as the integrated deuterium ion flux $\Gamma_{D^+,\mathrm{ot}}$ (D) and peak heat flux $q_{\mathrm{peak,ot}}$ (E) towards the outer target. In A-E the the orange and red lines highlight the transition into the sheath-limited and cold core regimes while the grey and black dots show all scenarios with nitrogen concentrations $c_{N,\mathrm{omp}} = 0.1\%$ and $c_{N,\mathrm{omp}} = 5\%$. The black star in A highlights the deuterium gas puff necessary to leave the sheath-limited regime with minimum nitrogen puff. F-J show density scans selected from the gas puff scan above. Scenarios are selected based on the upstream nitrogen concentration (denoted by the color) and when they are not in a sheath-limited or in the cold core regime. Shown are the integrated deuterium ion flux $\Gamma_{D^+,\mathrm{ot}}$ (F) and peak heat flux $q_{\mathrm{peak,ot}}$ (G) towards the outer target, the electron temperature at the outer target $T_{e,\mathrm{ot}}$ (H) and outer midplane $T_{e,\mathrm{omp}}$ (I) separatrix and the power crossing the separatrix $P_{SOL}$ (J) from the main plasma.

At the far-SOL boundary above the inner target the nitrogen concentration is increased. While the neutral gas densities increase close to the target, the deuterium neutral density inside the confined flux tubes has decreased compared to the low density case. Increasing the plasma density further beyond the rollover leads to further decreases in the plasma temperature in the SOL, most notably close to the targets. Also the neutral density increases and extends further towards the separatrix. Due to the unphysical behaviour of the fluid neutrals to stick to the flux tubes, the detachment of the electron temperatures starts at the outside of the divertor target and grows inwards, while in experiments it is usually the other way around. At the maximum upstream electron density, the neutral densities on the open field lines are high but inside the confined core region still at minimum values. Only in the final steps when the electron density starts to decrease, the neutral gas also enters the confined plasma.

In Figures 9.3-9.6 the minimal value for the deuterium ion input at the core boundary was chosen. The same analysis was repeated with high deuterium core fueling. To select appropriate values for each tokamak, the parameter scans are searched for the lowest deuterium gas puff that by itself ($N_{\text{puff}} = 10^{18}$ atoms/s) reaches the condition $T_{e,\text{ot}}/T_{e,\text{omp}} < 0.6$. These values (marked as stars on the x-axis in Figures 9.3-9.6A) are taken for the high ion influx $D_{\text{core}}$. Figures which summarize the model results with high core fueling are given in Appendix A.5.

The results of the gas puff scans both with low and high core fueling are now used to deduce the minimum upstream impurity concentrations and densities which are necessary to achieve rollover. Here the rollover is defined as the maximum integrated deuterium ion flux towards the outer target. This definition does not take into account whether the target heat flux or temperature fulfill any safety constraints. The results are presented in Figure 9.8. It shows also an identical parameter scan with 50% increased input power compared to the values provided in Table 9.1. In all tokamaks, the difference between low or high core fueling is small. The jump towards higher densities for high impurity concentrations in ITER and DEMO is caused by a flat top phase experienced in these tokamaks at rollover (see Figures A.15F and A.16F). In this phase the ion flux to the outer target stays constant at a high value, before it starts to decrease at higher densities. Because these flat top phases often contain a small positive gradient, the maximum target ion flux is often reached at higher densities at the end of these flat top phases. Thus the rollover detected by the metric used here is seen at higher densities. Using another metric such as the degree of detachment (DOD) [105], which compares the target ion flux to a quadratic scaling, the rollover would be detected already at the start of the flat top phase. For JET a comparison with the experimental fit $c_{N,\text{div}} = 1.245 \left(\frac{n_{e,\text{omp}}}{10^{-19}\,\text{m}^{-3}}\right)^{-2.43}$ from [74] is presented as a reference. The fit is only shown inside the range of densities seen in the experimental shots underlying this fit $(2.2 - 3.5) \cdot 10^{19}\,\text{m}^{-3}$. The empirical impurity concentration needs to be scaled by a factor 0.15 to match the results of the surrogate model. This can be partially because the empirical scaling predicts the impurity concentration in the divertor and not at the outer midplane. Nevertheless, similar discrepancies between existing scaling laws or simulations and the experimental results are common [74].

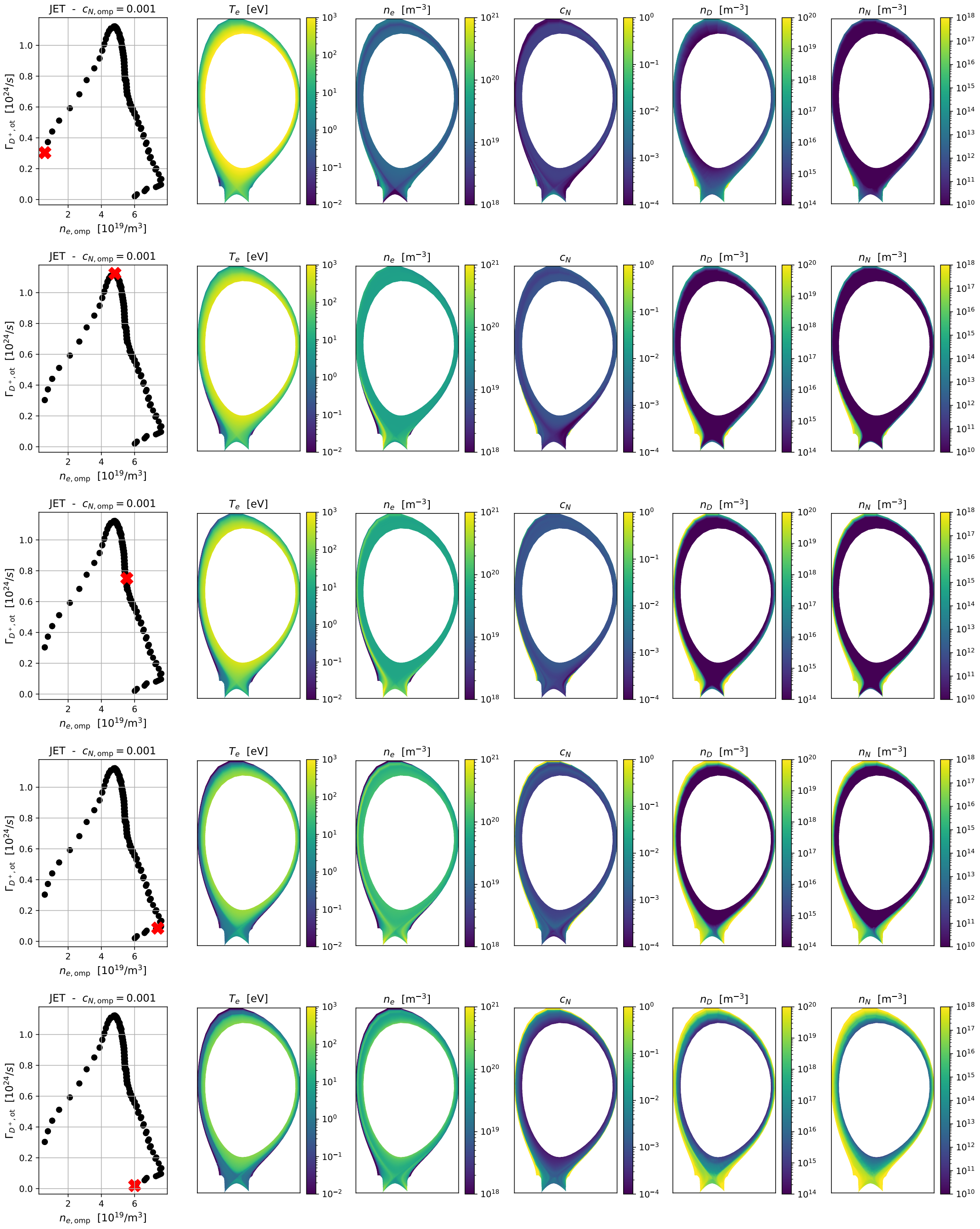


**Figure 9.7:** Predictions of the surrogate model for JET parameters with $c_{N,\mathrm{omp}} = 0.1\%$ and low core fueling. Each row shows on the left the deuterium ion flux towards the outer target against the outer midplane separatrix density $n_{e,\mathrm{omp}}$. In each row the 2D profiles on the right correspond to the position marked by the red cross in the density scan on the left. Shown are from left to right the electron temperature, electron density, nitrogen concentration, neutral deuterium gas density and neutral nitrogen gas density.

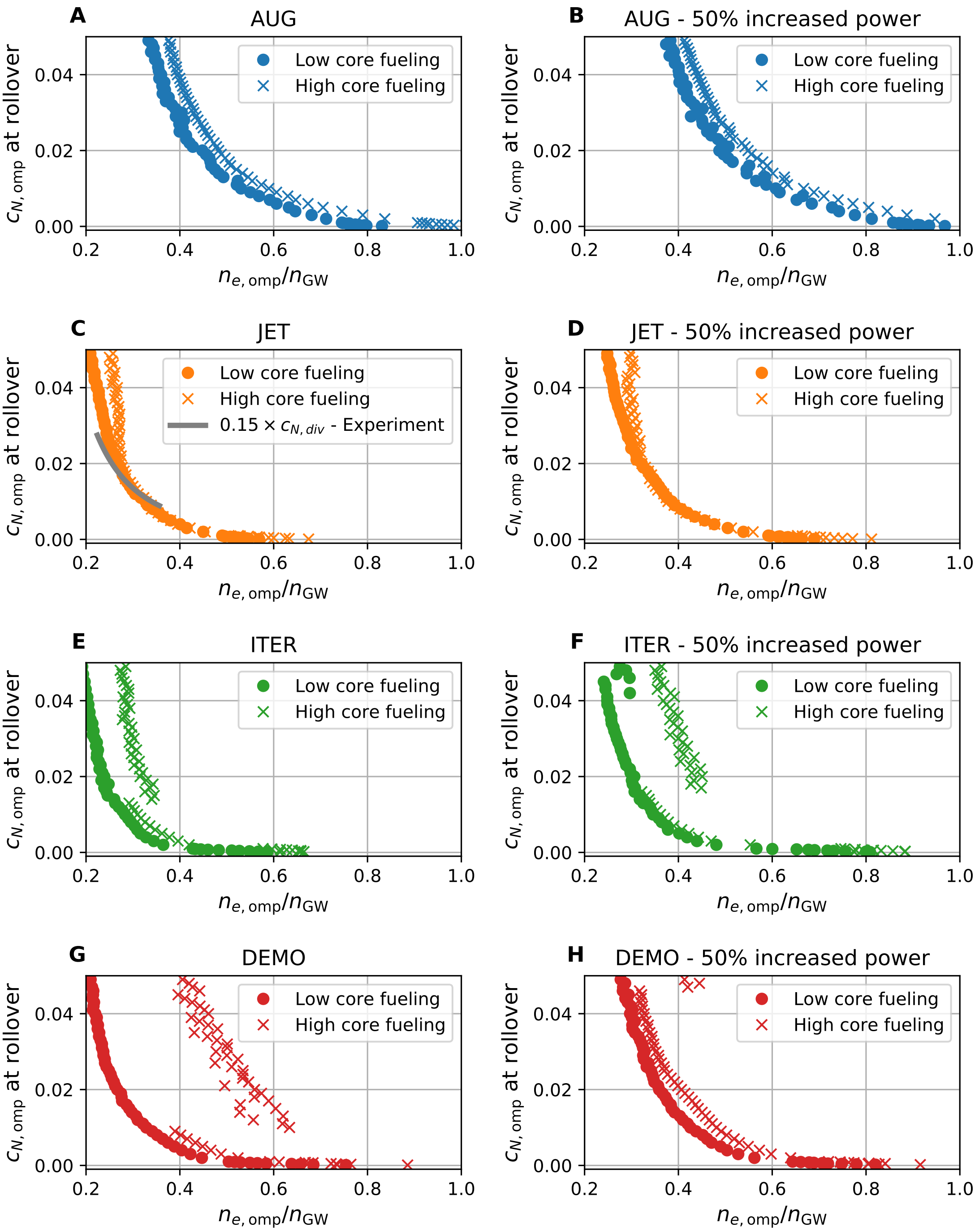


**Figure 9.8:** Impurity concentrations against the electron density at the outer midplane separatrix at particle flux rollover for AUG (A, B), JET (C, D), ITER (E, F) and DEMO (G, H) as predicted by the surrogate. The rollover is defined as the point of maximum deuterium ion flux to the outer target for fixed impurity concentration at the outer midplane separatrix. For comparison across tokamaks the densities are divided by the Greenwald densities given in Table 9.1. The JET results (C) show the experimental fit (grey) given in Equation (6) in [74] multiplied by 0.15. The low core fueling scenarios use $D_{\mathrm{core}} = 10^{20}$ atoms/s while the high fueling scenarios use a $D_{\mathrm{core}}$ equal to the gas puff value marked by stars in Figures 9.3-9.6. The scenarios in B, D, F, H use 50% higher input power than given in Table 9.1.

In conclusion, this section demonstrates how the developed surrogate models can be employed to model the conditions in the scrape-off layer across several tokamaks. With these models it was simple to derive operational constraints for the different tokamaks in the form of the necessary upstream density and nitrogen concentration. Doing the same by simulations would have required extensive simulations studies. Although the models are based on reduced fidelity simulations, the predicted nitrogen concentrations show the same functional trends as seen in experimental measurements. The discrepancy is on a similar level as predictions made by analytical scaling laws. But in comparison to simple analytical derivations, the surrogate models offer the full two dimensional output, which allows for more applications.

# Chapter 10

# Transfer learning

The models constructed in the previous sections relied on an extensive database of reduced fidelity simulations (Section 6). The validity of these models is constrained by the validity of these underlying simulations. In this section we compare the model to higher fidelity simulations, assess its deviations and try methods to mitigate those. For simplicity this relies again only on the model part for the electron temperature because with the modular model setup all other plasma properties can be included later.

A group of existing Q=10 baseline neon-seeded simulations from the ITER IMAS database was selected as higher fidelity simulations. These simulations were already presented in [58], [131], [200] and are called ITER simulations in the following. The shot numbers in the IMAS database of all the ITER simulations used here can be found in the Appendix in Table A.4. These ITER simulations differ in many ways from the simulation setup used in the fluid neutral database here. While the developed simulation database contains cases of ITER-like major radius, these rely on JET wall and magnetic field geometry. The ITER simulations have the correct ITER F57 wall geometry [125] and magnetic equilibrium with a full tungsten divertor and beryllium as first wall material. It was assumed that the divertor is fully covered in eroded beryllium, therefore the targets are treated as beryllium. The wall materials affect only the boundary conditions as no erosion was considered in the simulations. Additionally, the ITER simulations were conducted with a kinetic neutral model and corresponding implementation of gas puffs and particle pumps, follow stricter convergence metrics, use neon instead of nitrogen as impurity seeding and contain helium ions in the plasma. In the ITER simulations only three scalar parameters are independently varied: the input power, the deuterium gas puff and the neon gas puff. While the helium ion inflow is also varied it is perfectly correlated with the input power. As can be seen in Table 1 of [131], these simulations cover a variation of neon concentrations at 100 MW input power and a variation of power at fixed neon concentration. Although the ITER baseline scenario is an H-mode, these simulations contain no H-Mode pedestal but instead assume spatially constant anomalous transport ($D_\perp = 0.3\,\mathrm{m^2/s}$, $\chi_\perp = 1.0\,\mathrm{m^2/s}$).

If we assume the neon puffing rate can be translated one-to-one to a nitrogen puffing rate, then all ITER simulations lie inside the parameter space covered by the models trained on the fluid neutral dataset. Therefore, the previously trained NN2D model can be used to make predictions for the ITER scenario. Figure 10.1A depicts the predicted electron temperatures at the outer target at the separatrix for the parameters of the ITER simulations ($R = 6.2\,\mathrm{m}$, $B = 5.3\,\mathrm{T}$, $D_\perp = 0.3\,\mathrm{m}^2/\mathrm{s}$, $\chi_\perp = 1.0\,\mathrm{m}^2/\mathrm{s}$, $D_\mathrm{core} = 9.1 \cdot 10^{21}\,\mathrm{atoms/s}$, $P_\mathrm{in} = 100\,\mathrm{MW}$) and all possible combinations of deuterium and nitrogen gas puff. For comparison the corresponding temperatures in the ITER simulations at 100 MW input power are depicted as markers on top (nitrogen puffing rate equal to neon puffing rate). It is apparent that in all cases the surrogate model predicts target temperatures higher than in the ITER simulations. Figure 10.1B compares the model predictions to the ITER simulations for varying deuterium gas puff and fixed nitrogen/neon puff of $10^{20}$ atoms/s. For low deuterium gas puff, the surrogate predicts a sheath-limited regime with target temperatures identical to the outer midplane. For high gas puffs, the upstream temperature stays relatively constant between $100 - 200\,\mathrm{eV}$ while the target temperature drops first to 10 eV and then further decreases. In all cases the temperatures at the target in the corresponding ITER simulations are lower, but it appears that these temperature deviations are mostly a shift of the same patterns to lower gas puff values.

The high fidelity simulations have a different neutral model, gas puff implementation and neon instead of nitrogen. To reduce the deviations the gas puff values which are used as inputs in the surrogate model can be determined by an effective scaling. This effective scaling tries to compensate not only the change in impurity species but the combined effect of all the changes. Similar approaches are used when comparing results from different simulation codes [208]. To do this, the ITER simulations are randomly split into a set of 62 training simulations and 16 test simulations. Different scaling factors for the gas puffs are tested and the optimal values found by minimizing the median absolute error of the temperatures at the outer target separatrix on the training set. The optimal scaling is found to be $\mathit{\Gamma}_\mathrm{D,surr} = 10^{0.9}\mathit{\Gamma}_\mathrm{D,ITER}$ $\mathit{\Gamma}_\mathrm{N,surr} = 10^{0.3}\mathit{\Gamma}_\mathrm{Ne,ITER}$. Applying these scaling factors corresponds to shifting the surrogate predictions in the gas puff space as depicted in Figure 10.1C. There the gas puff values given on the axis describe the values of the ITER simulations. Outside of the dashed lines the surrogate model is extrapolating because the input gas puffs to the neural network are larger than the maximum values seen in the fluid neutral dataset. But this does not seem to cause any problems and the agreement between the surrogate prediction and the ITER simulations has drastically improved (Figure 10.1D). But such a simple correction cannot address all differences, and including more observables (e.g. ion densities) besides the temperature would show that one cannot find a consistent scaling which matches all observables.

Even considering just the electron temperature, it is impossible to find a gas puff scaling that achieves agreement everywhere in the domain at once, but rather only key locations can be optimized. Figure 10.2A depicts the electron temperatures in one ITER simulation from the test set. The corresponding temperature prediction by the surrogate with the applied gas puff scaling relies on the JET geometry (Figure 10.2B) but can be reshaped into the correct ITER geometry with simple geometric transformations (Figure 10.2C). Comparing Figure 10.2A to Figure 10.2C demonstrates that although the gas puff rescaling reduced the discrepancies at the divertor targets there are still large discrepancies at other locations in the domain. Due to

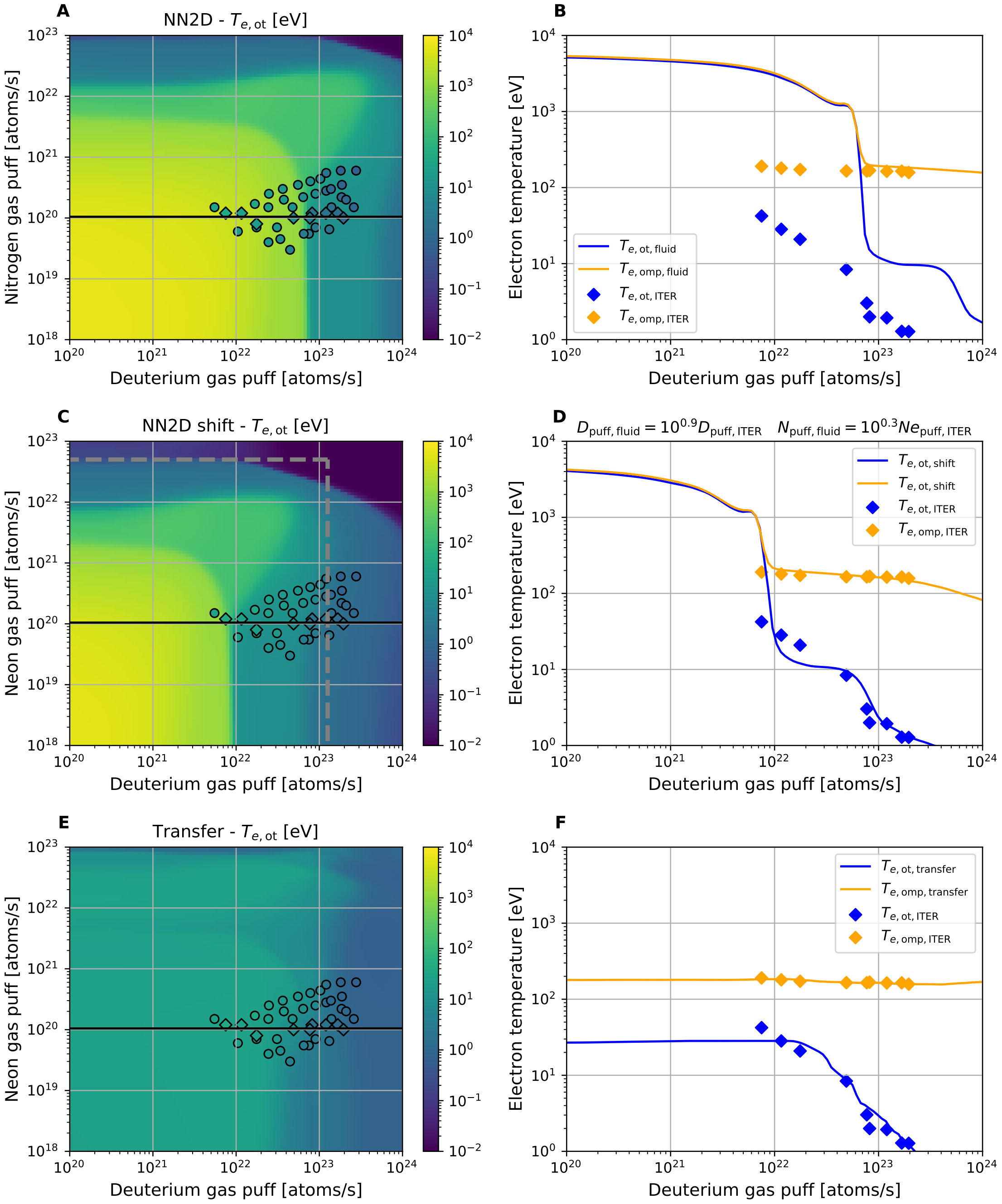


**Figure 10.1:** Temperatures predicted by the NN2D fluid neutral model (A,B), the model with rescaled gas puff (C,D) and the transfer learned model (E,F) in comparison to the temperatures in the ITER simulation (markers on the left) with 100 MW input power. The color inside the markers on the left corresponds to the temperature in the ITER simulations. The dashed grey lines in C mark the area outside which the neural network is extrapolating. The diamond shapes in the left column mark the cases seen also on the right. The ITER simulations depicted here stem from both the training and the test set.

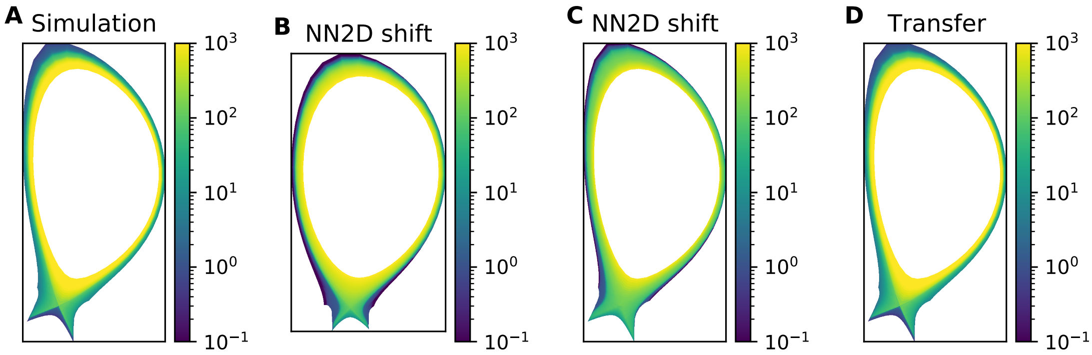


**Figure 10.2:** Electron temperature in one ITER simulation in the test set (A) and the corresponding predictions by the NN2D fluid neutral model with rescaled gas puff (B), the same prediction but geometrically shaped to ITER geometry (C) and the transfer learned model (D).

the different neutral model, the underlying simulations behind the surrogate model are too different from the ITER simulations.

Another method to decrease the discrepancy is to retrain the surrogate model using the high fidelity simulations. Because the dimensionality of the last layer of the neural network is directly linked to the geometry of the simulation grid, it has to be replaced with a new one with the correct number of neurons. Then only the weights of this new last layer are retrained using the ITER training data, while the rest of the network stays the same. The idea behind this approach is that the physical relations already learned on the fluid neutral dataset remain preserved, while the results are calibrated towards the new high fidelity data. In the machine learning community such a procedure is called transfer learning [157]. Figure 10.1E and F depict the results of this transfer learned model in the same parameter scan as before. The agreement with the ITER simulation has improved but simultaneously the entire temperature landscape has changed drastically and bears almost no resemblance to the results of the previous model. Ideally the transfer learned model should retain some of the aspects learned on the fluid neutral dataset, especially in the parts of the gas puff domain not covered by the ITER simulations. This is not the case here as for diminishing deuterium gas puff the target temperature remains at a constant level (Figure 10.1F), while in reality it should increase and approach sheath-limited conditions as seen in the original surrogate model. Therefore, the transfer learning approach employed here might require additional fine tuning such that the ”knowledge” from the fluid neutral dataset is better retained and the model not entirely dominated by the new data. Nevertheless, this transfer learning achieves good agreement with the ITER simulations and not only at the outer target but in the whole 2D domain (Figure 10.2D).

A key question is whether the previous training on lower fidelity data in the transfer learning approach helps the model to learn on the new data. Ideally a smaller number of high fidelity simulations should be required than when training a model from scratch solely on the higher fidelity data. To evaluate this, varying numbers of high fidelity ITER simulations from the training set are used to repeat the transfer learning procedure and train NN2D models from scratch. For the NN2D models trained from scratch the hyperparameters are optimized via random searches similar as in Table 8.4. Using all available training simulations both approaches produce accurate temperature predictions at the outer target for cases from the test set across varying degrees of detachment (Figure 10.3A-D). Using only five training simulations both approaches still manage to capture the changes in target profiles with varying degrees of detachment but the deviations to the simulations have visibly increased (Figure 10.3E-H). Using the same fixed test set, the median errors of the two approaches are computed. Similar as in Section 8.4 the test errors continuously decrease with increasing number of training simulations (Figure 10.4). It appears that for higher number of training simulations the model trained from scratch is slightly more accurate but the differences are small. What is definitely not seen is that the transfer learning approach allows a reduction in the number of training simulations. However, it has to be considered that the test set is rather small so the error estimates come with larger uncertainties. The median absolute test errors obtained here are far lower than that obtained with any number of simulations on the fluid neutral dataset (Section 8.4). This is possible because the ITER simulations cover a much smaller parameter space with only little variations in the resulting temperatures. While the NN2D networks trained on the fluid neutral dataset in Section 8.3 required more than eight hidden layers, the best network trained from scratch on these ITER simulations uses only two hidden layers (see Figure A.17). This is another indication how much "easier" it is to learn the relations in the small ITER dataset.

The model trained from scratch has only three input parameters ($P_{\mathrm{in}}, \Gamma_{\mathrm{D}}, \Gamma_{\mathrm{Ne}}$) because only these three parameters are varied in the ITER simulations. The transfer learned model, however, still has the same eight input parameters which were varied in the fluid neutral dataset. In principle the transfer learned model can still respond to variations in all of these eight parameters. While the model response to variations in $P_{\mathrm{in}}, \Gamma_{\mathrm{D}}, \Gamma_{\mathrm{Ne}}$ is largely overwritten by the new data (as seen in Figure 10.1), it is unclear how this affects the model response to the other five input parameters. Because no high fidelity simulations are available for such variations, there exists not test data to verify the model predictions but they can only be checked for plausibility.

As an example application, this capability of the transfer learned model is used here to determine how an uncertainty in the assumptions of the cross-field transport coefficients affects to the temperatures at the outer target. For this 100 pairs of cross-field transport coefficients are sampled from normal distributions with means $\mu_{D_\perp} = 0.3\,\mathrm{m^2/s}$, $\mu_{\chi_\perp} = 1.0\,\mathrm{m^2/s}$ and 5% standard deviation. These are then used to run the same deuterium gas puff scan as in Figure 10.1F with fixed $\Gamma_{\mathrm{Ne}} = 10^{20}$ atoms/s. Figure 10.5 shows the impacts this has on the temperatures at the outer target and the outer midplane. At the outer midplane this changes mostly the temperatures inside the core plasma while the temperatures at the separatrix are only slightly affected. At the outer target the effects are greatest in the range $\Gamma_{\mathrm{D}} \in [0.5, 1.0] \cdot 10^{23}$ atoms/s where this causes variations in the peak temperature at the target by more than 5 eV. For more detached conditions the effect on the absolute temperature values at the target is a lot smaller. This behaviour seems plausible because in detached conditions the temperature at the target is mainly

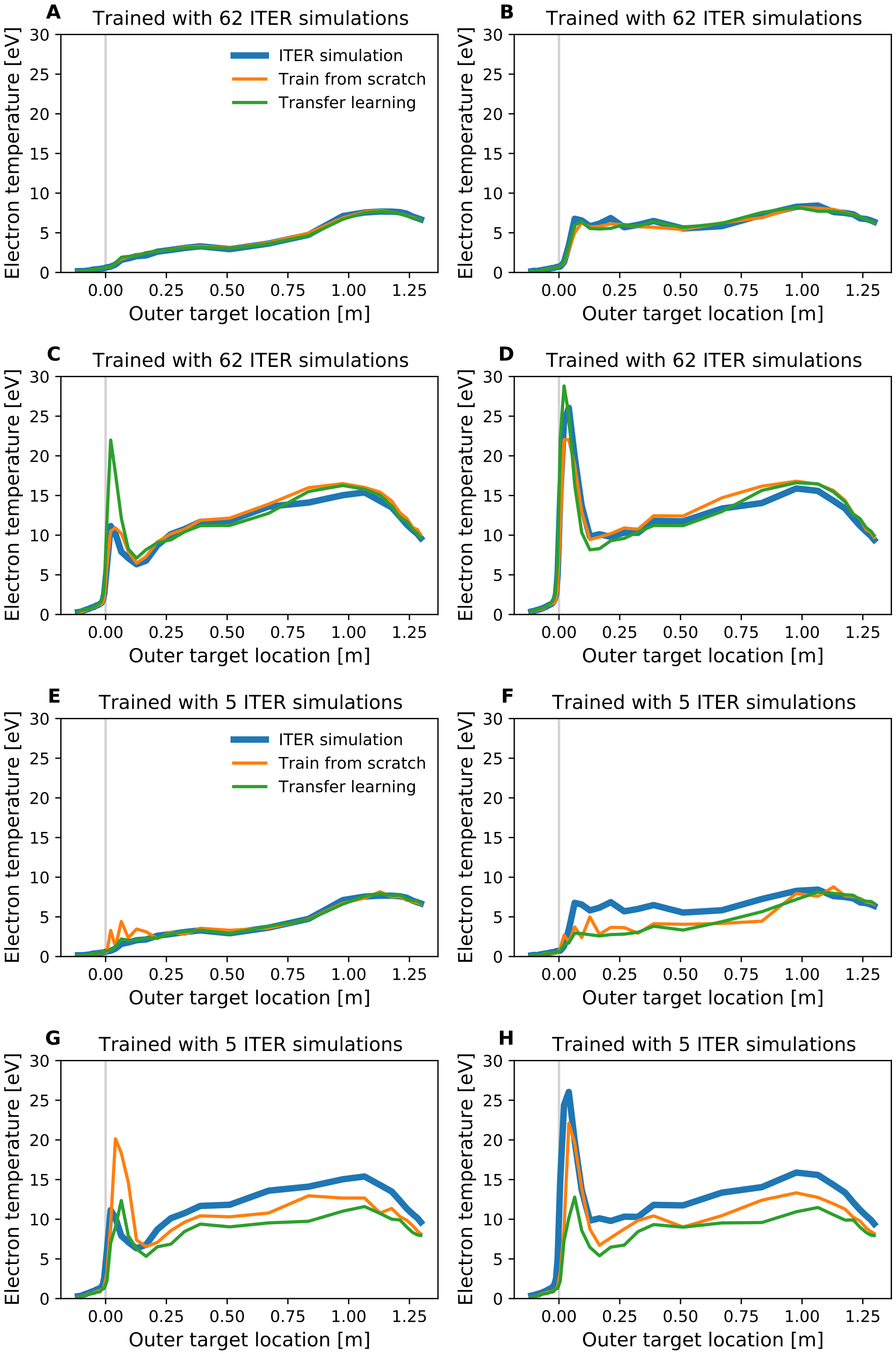


**Figure 10.3:** Comparison between the temperature profiles predicted by the transfer learned model (green) and the model trained from scratch (orange) at the outer target for four ITER simulations (blue) from the test set. The models are either trained with 62 (A-D) or 5 (E-H) ITER training simulations.

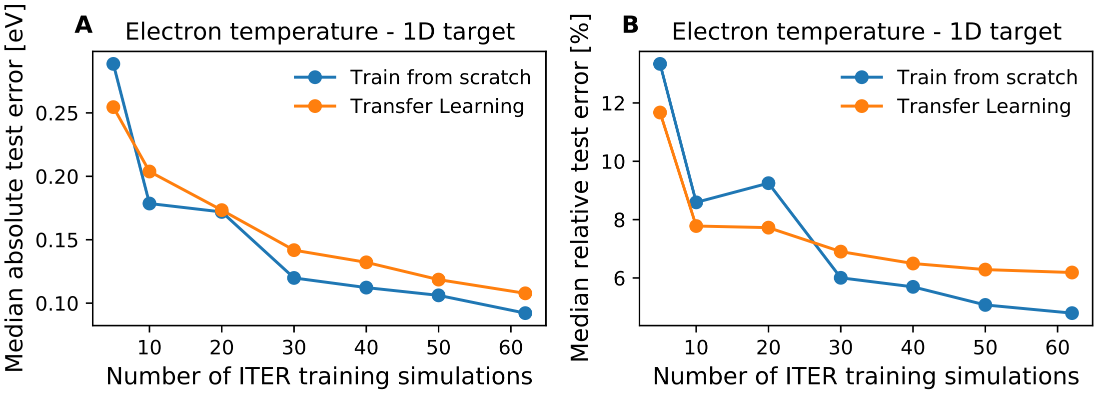


**Figure 10.4:** Median absolute (A) and relative (B) errors of the predicted temperatures at the outer target on the fixed ITER test set for models trained with varying amount of ITER training simulations either from scratch (blue) or as transfer learning based on the previous fluid neutral surrogate (blue).

determined by the volumetric power loss. So changes to the cross-field transport have little extra effect. In attached conditions the heat flux is concentrated close to the separatrix and thus a broadening of the heat flux profile can strongly reduce the peak temperatures at the target.

In conclusion, the surrogate model trained on the fluid neutral simulations shows already very similar trends as present in kinetic neutral simulations. Transfer learning allows adapting the model to higher fidelity simulations but the tested approach does not yield a more accurate model than training solely on few high fidelity simulations. It seems like the transfer learned model still incorporates the effects of parameters not varied in the high fidelity data, but more rigorous studies are required to evaluate this.

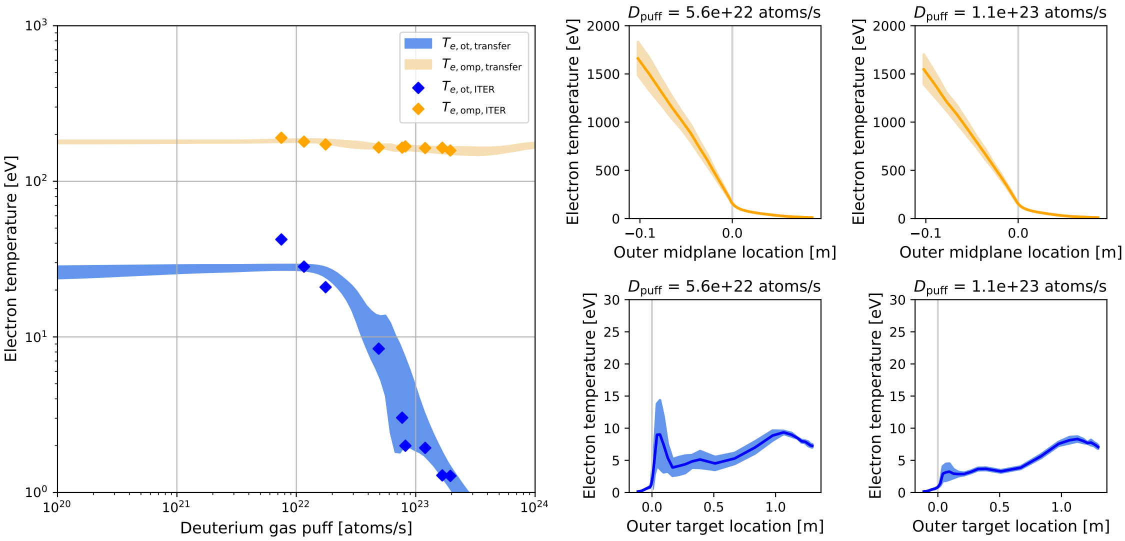


**Figure 10.5:** Predictions of the transfer learned model for the same scenario as in Figure 10.1F but with transport coefficients sampled from normal distributions. The spread of the curves depicts $\pm 1\sigma$ around the mean of the predictions.

# Chapter 11

# Conclusions

This thesis explored the possibility to develop a surrogate model for the tokamak scrape-off layer plasma.

To this end, a database of SOLPS-ITER simulations with a reduced fidelity fluid neutral model was created. The database varied eight tokamak parameters spanning several reactors and physical regimes. Crucially the dataset contains a size scaling to encompass scenarios mimicking existing as well as future larger tokamaks. It was discovered that the key difficulty in creating such a dataset lies in finding a suitable numerical configuration that yields stable simulations across the whole parameter space. For the simulation case in this study, an acceptable level of stability in the simulations was reached and recommendations for the choice of certain numerical settings can be made. But these recommendations are entirely based on trial and error regarding the simulation setup at hand. It remains unknown to which degree these recommendations still hold for SOLPS-ITER simulations with different underlying physics models. It has shown that it is unlikely that one ideal numerical configuration can be found. Future studies should employ adaptive numerical configurations which change according to the variations in the input parameters and the plasma conditions inside the simulations. Although the fluid neutral model introduces some artifacts not seen in simulations with kinetic neutrals, the dataset still reproduces general trends and effects across all scrape-off layer regimes including a rollover of the particle flux at the divertor targets.

Based on this data, Gradient Boosted Regression Tree and neural network based models have been trained and evaluated. While both machine learning methods were able to reproduce the results obtained in the simulations, the neural network models achieved higher accuracies in all tasks and are hence a better candidate for a surrogate model. Two general neural network types based on the fully connected feedforward architecture were evaluated. The usual setup where the output layer contains a separate neuron for each location and neural network architectures which treat the location as additional input and provide scalar outputs. The latter is similar in design to the networks employed in physics-informed neural networks. While predicting only the temperatures at the one dimensional divertor target, both network types are able to obtain similar levels of accuracy. When tasked with providing an output for the whole two dimensional simulation domain, the latter network type showed inferior accuracy. However, this might come from the smaller hyperparameter optimization conducted for these networks as they have drastically higher training times. Training the former neural network type to output the electron temperature either solely at the divertor target or in the whole simulation domain, obtains similar levels of accuracy. A similar comparison was made between a surrogate model constructed of independent networks for different plasma properties and a model that relies on a single network for plasma temperatures and particle densities. Here the model with multiple independent networks obtained slightly higher accuracy. Although the differences are small and more advanced or fine tuned network architectures might make both solutions comparable. Nevertheless, the current recommendation for a surrogate model is to train independent networks which each predict one plasma property in the whole simulation domain.

For each of the independent networks, which are trained to predict only one plasma parameter, very similar network hyperparameters are optimal. The fully connected networks consist of around 10 hidden layers with each around 1000 neurons with SELU activations. The networks use no batch normalization and are trained with small learning rates ($\leq 10^{-4}$) and small L2 regularization ($10^{-5} - 10^{-4}$). Similar hyperparameters were also the optimum for the networks trained to predict the electron temperature only at the divertor target. The optimal network hyperparameters change when smaller numbers of simulations are used as training data. For smaller datasets the networks with higher L2 regularizer and learning rate achieve the best possibly accuracy. When a neural network was trained on a small separate dataset of higher fidelity simulations from the ITER database that span a much smaller parameter space, the highest accuracy could be achieved with a neural network with only 2 hidden layers. This shows that the optimal neural network architecture and hyperparameters depend less on the considered plasma observable or output dimensionality. More important are the dimensionality of the input parameter space, the variability in the results and the number of training simulations.

Tests where conducted in training surrogate models with varying number of training simulations both on the fluid neutral SOLPS-ITER dataset as well as the set of higher fidelity simulations from the ITER database. It was found that the extend of the parameter space and the variability of the simulation results drastically determine the maximum accuracy that is achievable with a surrogate model for a given number of training simulations. On the fluid neutral dataset the accuracy decreased drastically below 1000 training simulations. Above 1000 simulations the median errors estimated on the test set of the predicted temperatures at the outer target are 1.6 eV and 24% which decrease using all available training simulations to 1.0 eV and 13%. Models trained on the smaller ITER simulations are able to obtain median test errors of less than 0.1 eV and 5% at the outer target already with only 60 simulations as training data. This shows that for designated applications it is likely worthwhile to decrease the scope of the developed surrogate models in favor of higher accuracy. Transfer learning from the surrogate model trained on the fluid neutral dataset to the ITER dataset has shown to yield no improvements compared to developing a model on the ITER dataset from scratch. This might change when more complex higher fidelity datasets or more advanced neural network architectures are used.

The developed surrogate models struggle with preserving the spatial gradients of plasma temperatures. This affects only regimes where the spatial temperature differences are small in comparison to the overall scale of values. Deriving heat fluxes from the surrogate outputs in such cases leads to wrong directions and an overestimation of the absolute values of the heat fluxes. This concerns mostly low density simulations in which the temperature values are beyond the physically possible range. But also scenarios with reasonably high temperature attached divertor conditions can be affected. So far no satisfactory solution to this issue has been found. The usage of designated models, which are trained to predict the heat fluxes themselves, avoids these errors. However, in that case the heat fluxes are not consistent with the temperatures and other plasma properties predicted by the surrogate model. The derived particle fluxes in all regimes and the derived heat fluxes in simulations in detached conditions are less affected by this issue.

Nevertheless, it was demonstrated that the developed surrogate models can be employed to make predictions for the conditions in the scrape-off layer across several tokamaks. This was used to predict the plasma density and impurity concentrations necessary for detachment in ASDEX Upgrade, JET, ITER and DEMO. Comparison with experimental results from JET shows that correct functional relations can be recovered but the estimated impurity concentrations are almost an order of magnitude too low. This discrepancy is on a similar level as predictions made by analytical scaling laws. Thus the surrogate models can already be employed in fast scoping studies to determine possible operating spaces for future reactors like DEMO or SPARC.

The key to bring surrogate models of the tokamak scrape-off layer towards state-of-the-art accuracy and validity lies in decreasing the necessary number of training simulations and accelerating each individual simulation.

Specifically suited to reduce the number of necessary training simulations are techniques from the class of active learning [157]. Such approaches iterate between training the surrogate and generating new simulation data, while the parameters of new simulations are chosen based on an estimated improvement of the surrogate error [87], [209]. The work done in this thesis can act as a benchmark for such studies because sophisticated active learning schemes are only worthwhile if they can decrease the necessary number of simulations significantly.

Besides trying to reduce the overall number of training simulations, one can also employ mixed fidelity approaches which train a model on a mixture of simulations with varying physical and computational complexity. The goal is to decrease the number of computationally demanding simulations while still providing a sufficient exploration of the parameter space through the faster simulations. Candidate levels of complexity are variations in the neutral models ("Advanced Fluid Neutral" [143]–[145], fully kinetic EIRENE neutrals [140]) and inclusion of plasma cross-field drifts. But also simpler one dimensional models like the DIV1D code [75] could prove useful. The main challenge is finding, training and optimizing network architectures that are able to leverage and balance the diverse set of training data generated by the different fidelity simulations. These mixed fidelity approaches should ultimately go beyond just simulation data and include experimental data. While this can be seen as highest level of fidelity, it comes with uncertainties in the measurements, hidden biases and sparsity of the available measured signals.

To aid the process of creating a database of scrape-off layer simulations, more research should be conducted to find adaptive algorithms that change the numerical parameters of a simulation automatically. This should also include more research into how the already developed surrogate can be used to provide initial plasma states for consecutive simulations.

Going even further a surrogate model could be developed without running any simulations in the first place by using Physics Informed Neural Networks (PINN) [210]. In these models the governing equations (here the Braginskii equations) are included in the loss function of the network. During training the networks try to find an approximate solution to the equations. But training such models comes with its own technical challenges [203], [211].

In conclusion, this thesis has opened several pathways for following research to bring surrogate models for the scrape-off layer to their full capabilities.

# Chapter 12

# Acknowledgements

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

The author gratefully acknowledges computing time on the supercomputer JURECA [212] at Forschungszentrum Jülich under grant no. solsur.

Although this PhD thesis is the product of a single author, it owes its existence to the enduring support of my colleagues at the Forschungszentrum Jülich, to whom I owe my sincerest appreciation.

Most of all, I want to thank my supervisor Sven Wiesen for the possibility to conduct this project, your constant availability, the motivating spirit and your never ending willingness to put every available lever into motion for me. Of course my gratitude goes also to Yunfeng Liang and Sebastijan Brezinsek, for their good advice and enduring patience, without whom this thesis would not be possible.

I want to express my eternal gratitude to my parents, who have played a pivotal role in shaping the person I am today. I extend my heartfelt appreciation for their unwavering support, understanding and endless patience through my busiest moments. Finally, I am grateful to Maja for her love, encouragement and backing throughout these recent years.

# Chapter 13

# Hardware and software

All simulations are conducted on the batch nodes of JURECA DC [212] with SOLPS-ITER version 3.0.6 [80], [84]. The analysis and surrogate training is done with Python version 3.8.5 and the packages numpy 1.18.5 [213], scipy 1.5.2 [214], pandas 1.1.3 [215], [216], matplotlib 3.3.1 [217], scikit-learn 0.23.2 [218] and tensorflow 2.3.1 [169].

To use SOLPS-ITER to compute auxiliary outputs based on surrogate predictions (as done in Section 8.6) some specific modifications are necessary. While starting SOLPS-ITER with the option *ntim=0* should do so, it was found that for the setup here the heat fluxes are not recomputed but instead the wrong values are read from the inputfile. Starting SOLPS-ITER for one timestep *ntim=1* had the problem that the code read the particle fluxes from the inputfile instead of recomputing them. As workaround therefore with SOLPS-ITER version 3.0.6 the simulation needs to be first started with *ntim=0* and the surrogate provided inputfile and then the resulting *b2fstate* can be used as new inputfile to start with *ntim=1* and a tiny timestep e.g. *dtim*=$10^{-10}$. Users need to be aware which quantities in the SOLPS-ITER initial state are actually used by the code and which can be set to zero and to provide all required quantities via the surrogate. Additionally users should increase the number of digits used to represent the plasma fields in *b2fstati* and *b2fstate.* Only then smooth restarts and proper computation of auxiliary quantities is achievable. This can be done by setting *ifmt=0* in the B2.5 source code file *cfwure.F.*

Generative AI models have been used in writing parts of this thesis. For large parts of the results the first drafts of the text were created by recording an audio dictation by the author, which was then transcribed using Whisper by OpenAI. In most parts these first drafts were largely rewritten manually by the author. Additionally ChatGPT by OpenAI was used for creating text snippets based on keywords and short texts by the author. These pieces of text have been revised, restructured and combined with the authors own text passages. Nevertheless, individual sentences in this work are also unchanged as suggested by ChatGPT. This applies predominantly to chapter 3 but to a lesser degree also to other chapters.

# Appendix A

# Appendix

This appendix contains additional text, Figures and Tables in support of the conclusions drawn in the main text. The material is provided for the sake of completeness and as reference for the interested reader. All major results are drawn in the main text, readers are therefore referred to the corresponding sections in the main text.

## A.1 Oscillations

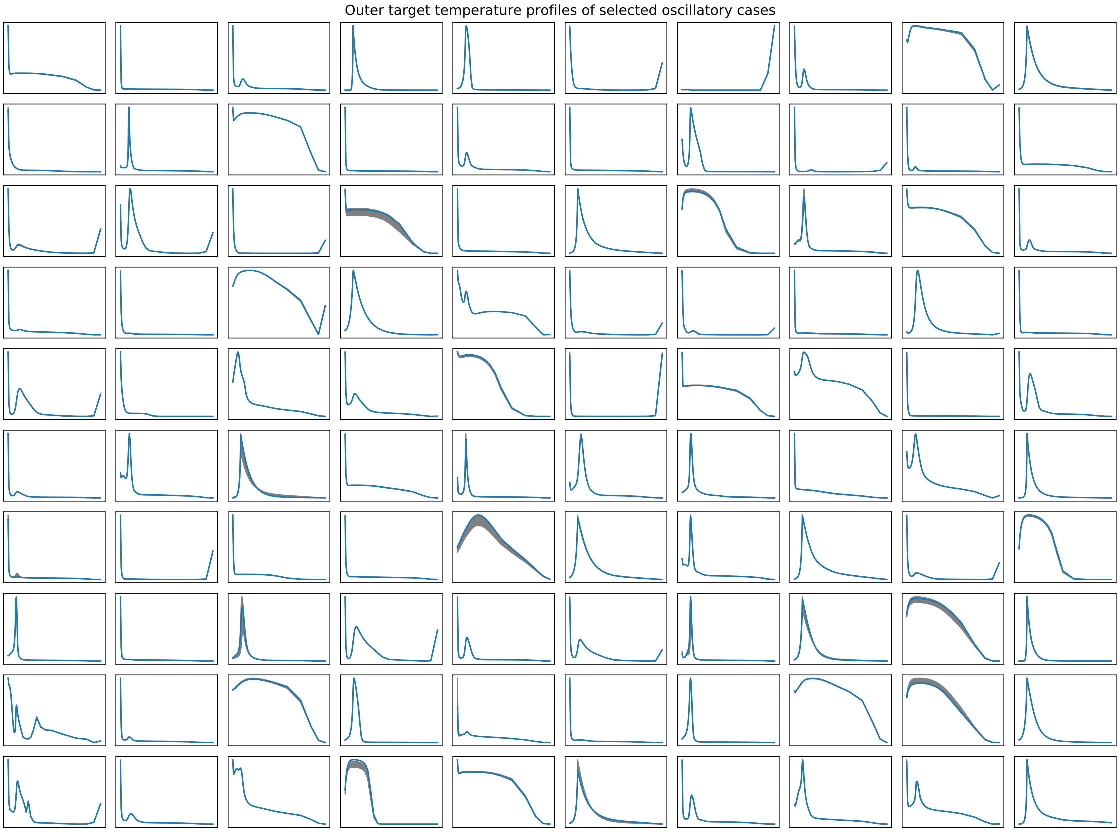


**Figure A.1:** Profiles of the electron temperature across the outer divertor target. Each of the 100 profiles stems from a simulation stopped due to stable oscillations. The x- and y-dimensions of each subplot differ. The final temperature values in the last simulation timestep (blue) are displayed on top of $\mu_{T_{e,\mathrm{ot}}} \pm \sigma_{T_{e,\mathrm{ot}}}$ (grey) with the mean $\mu_{T_{e,\mathrm{ot}}}$ and standard deviation $\sigma_{T_{e,\mathrm{ot}}}$ calculated over the last 0.1 s of the simulations.

## A.2 Influence of the training data size

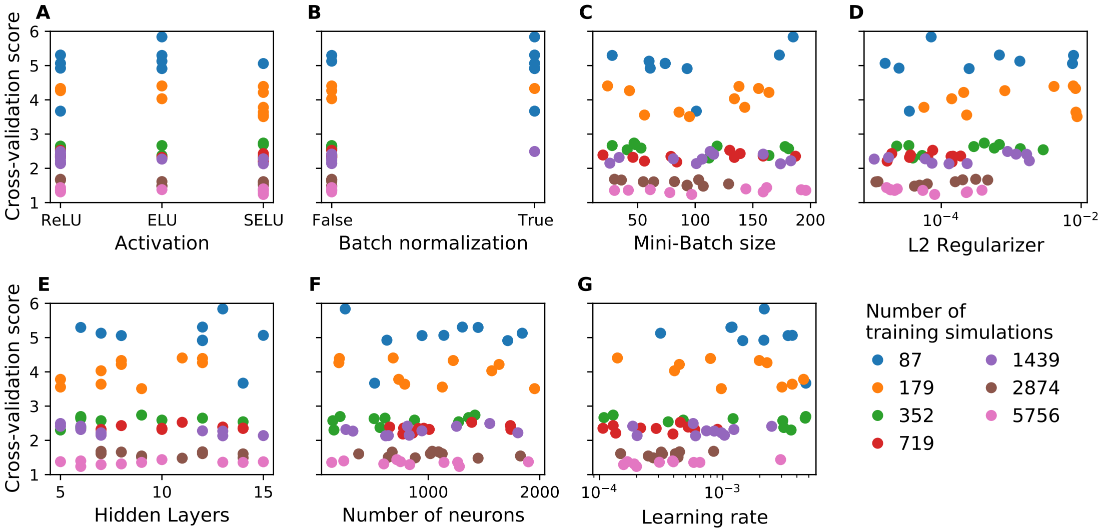


**Figure A.2:** Cross-validation scores of the ten best NNpos1D hyperparameter configurations found for each number of training simulations. Each dot represents one tested hyperparameter configuration in the random search.

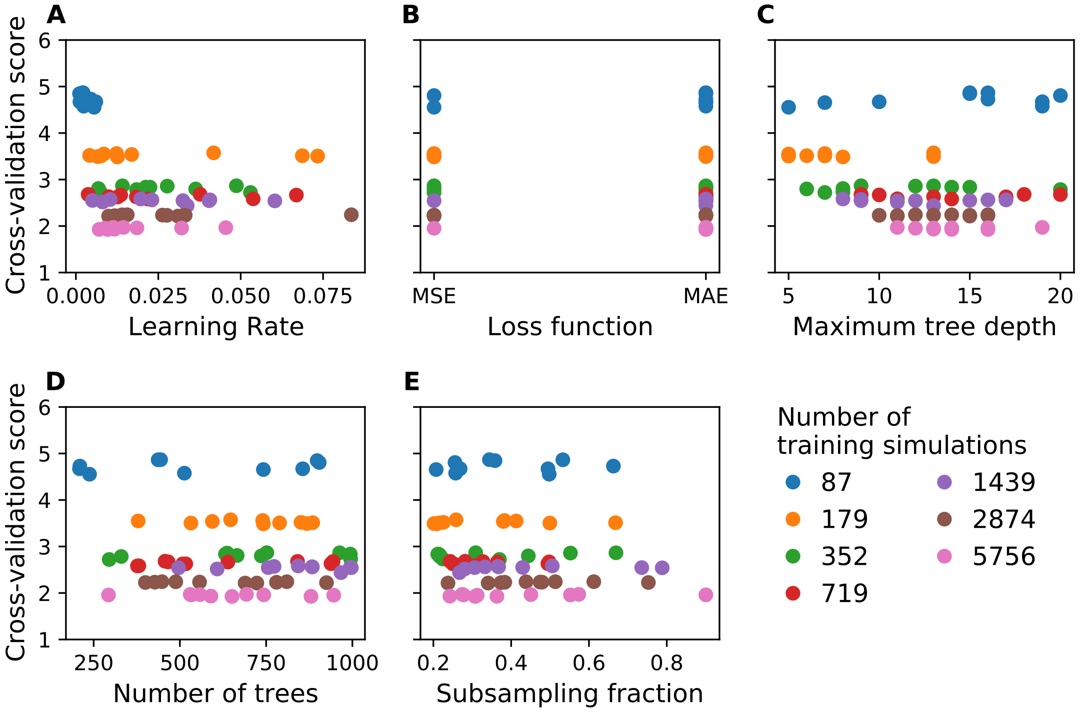


**Figure A.3:** Cross-validation scores of the ten best GBRTpos hyperparameter configurations found for each number of training simulations. Each dot represents one tested hyperparameter configuration in the random search.

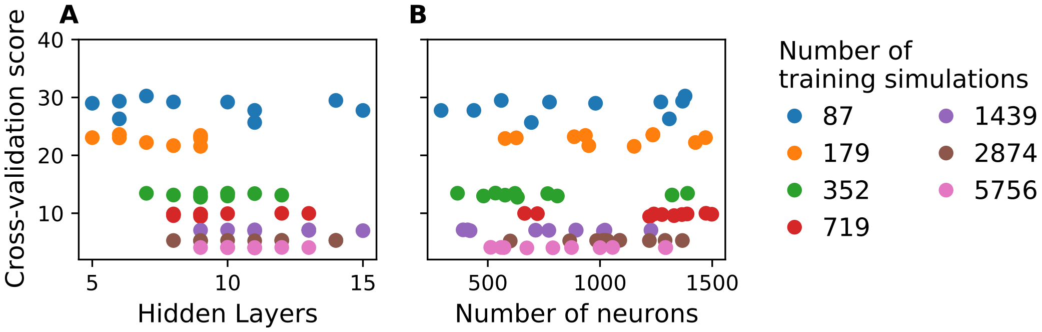


**Figure A.4:** Cross-validation scores of the ten best NN2D hyperparameter configurations found for each number of training simulations. Each dot represents one tested hyperparameter configuration in the random search.

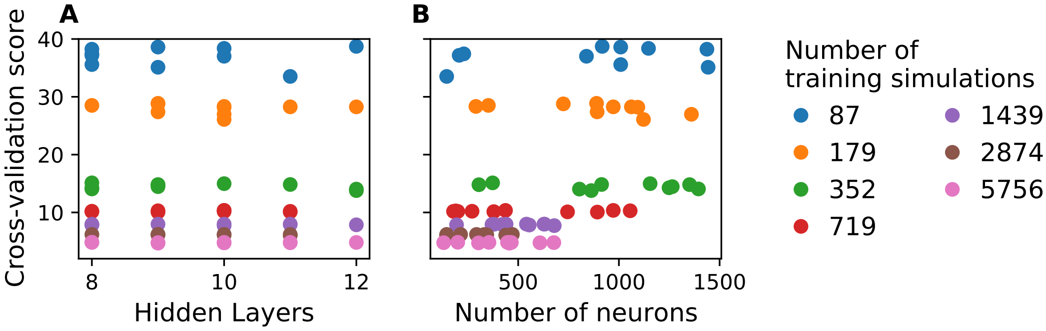


**Figure A.5:** Cross-validation scores of the ten best NNpos2D hyperparameter configurations found for each number of training simulations. Each dot represents one tested hyperparameter configuration in the random search.

## A.3 Incorporation of several plasma properties

| | | 2D | | | | |
|---|---|---|---|---|---|---|
| | | $T_e$ | $n_e$ | $n_D$ | $n_N$ | $p_e$ |
| NN2D-all-in-one | abs | 3.7 | 1.2e+18 | 4.4e+14 | 6.5e+12 | 1.5e+01 |
| | rel | 0.07 | 0.11 | 0.16 | 0.14 | 0.11 |
| NN2D-all-in-one-2 | abs | 4.3 | 1.3e+18 | 5.3e+14 | 7.3e+12 | 1.6e+01 |
| | rel | 0.088 | 0.13 | 0.2 | 0.17 | 0.13 |
| NN2D-separated | abs | 3.3 | 1.1e+18 | 3.8e+14 | 5.7e+12 | 1.4e+01 |
| | rel | 0.065 | 0.11 | 0.15 | 0.12 | 0.12 |
| NNpos2D-all-in-one | abs | 3.7 | 1.1e+18 | 4.9e+14 | 7.6e+12 | 1.3e+01 |
| | rel | 0.069 | 0.11 | 0.16 | 0.15 | 0.11 |

**Table A.1:** Median absolute and relative errors of the model predictions on the test set across the whole 2D domain. The absolute errors are given in the units $[T_e]$ = eV, $[n_e] = [n_D] = [n_N] = \mathrm{m}^{-3}$,$[p_e]$ = Pa.

| | | 1D | | | | |
|---|---|---|---|---|---|---|
| | | $T_e$ | $n_e$ | $n_D$ | $n_N$ | $p_e$ |
| NN2D-all-in-one | abs | 0.96 | 3.4e+18 | 2.1e+18 | 6.7e+15 | 6.5 |
| | rel | 0.13 | 0.2 | 0.18 | 0.28 | 0.2 |
| NN2D-all-in-one-2 | abs | 1.2 | 3.8e+18 | 2.6e+18 | 8.3e+15 | 7.8 |
| | rel | 0.17 | 0.26 | 0.25 | 0.36 | 0.25 |
| NN2D-separated | abs | 0.97 | 4e+18 | 2.1e+18 | 6.3e+15 | 7.9 |
| | rel | 0.12 | 0.25 | 0.19 | 0.27 | 0.24 |
| NNpos2D-all-in-one | abs | 1.2 | 4e+18 | 4.6e+18 | 1.4e+16 | 8.4 |
| | rel | 0.15 | 0.26 | 0.33 | 0.49 | 0.25 |

**Table A.2:** Median absolute and relative errors of the model predictions on the test set at the 1D outer target. The absolute errors are given in the units $[T_e]$ = eV, $[n_e] = [n_D] = [n_N] = \mathrm{m}^{-3}$,$[p_e]$ = Pa.

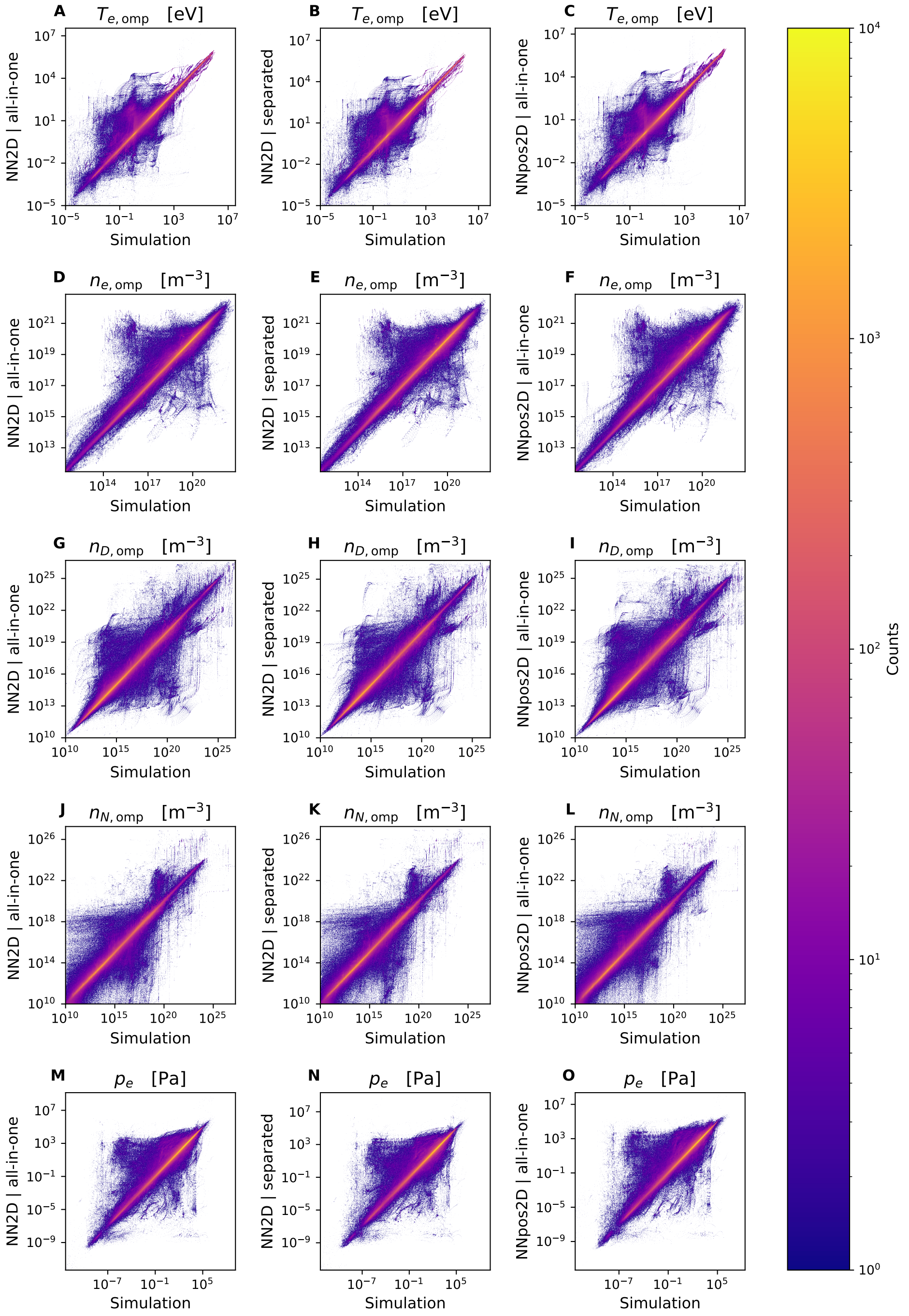


**Figure A.6:** The model predictions in all domain grid cells against the results in the simulations for all simulations in the test set. The color denotes the number of overlapping points at the same plot position. A model is more precise the closer the predictions are to the diagonal.

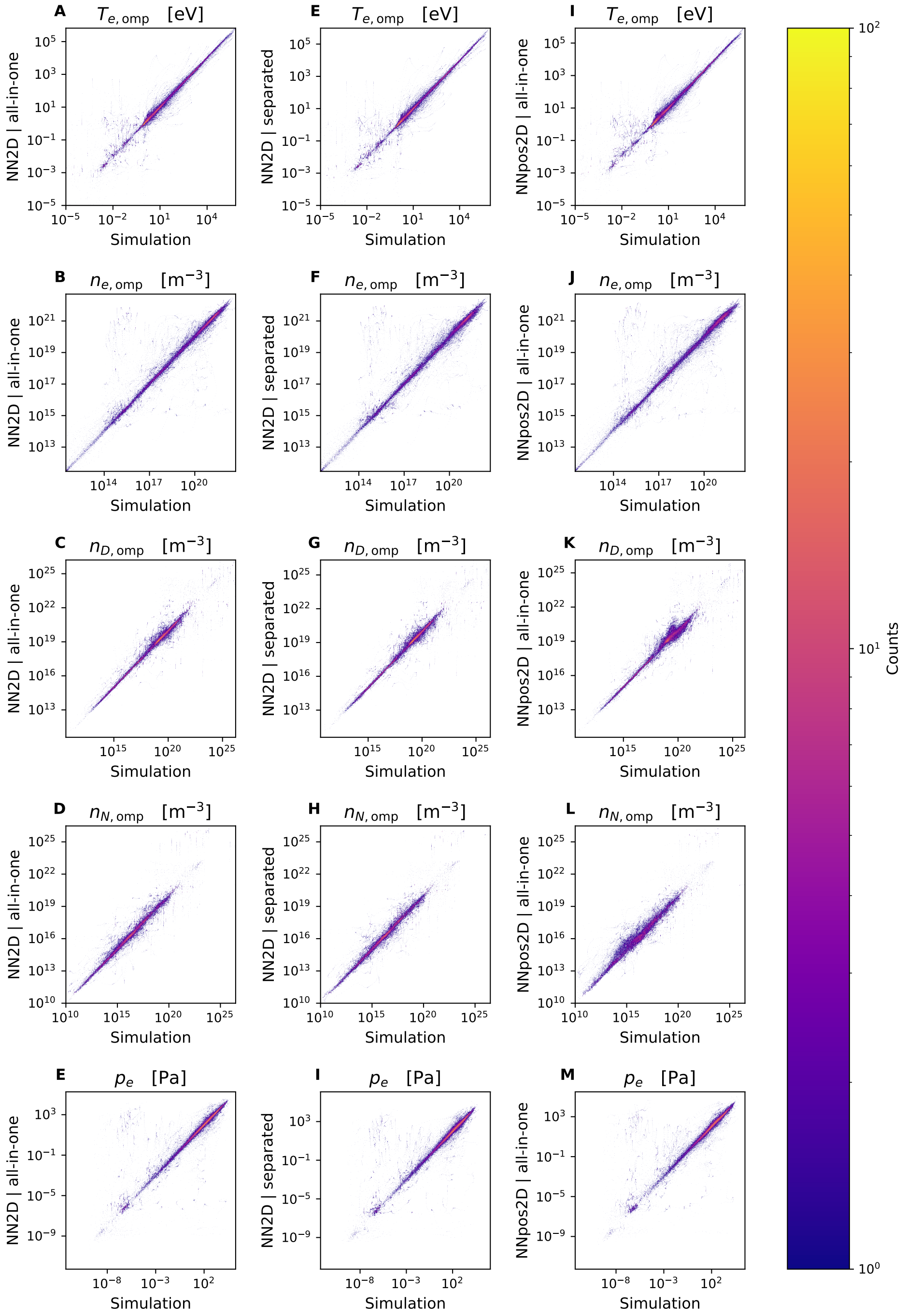


**Figure A.7:** The model predictions in all outer target grid cells against the results in the simulations for all simulations in the test set. The color denotes the number of overlapping points at the same plot position. A model is more precise the closer the predictions are to the diagonal.

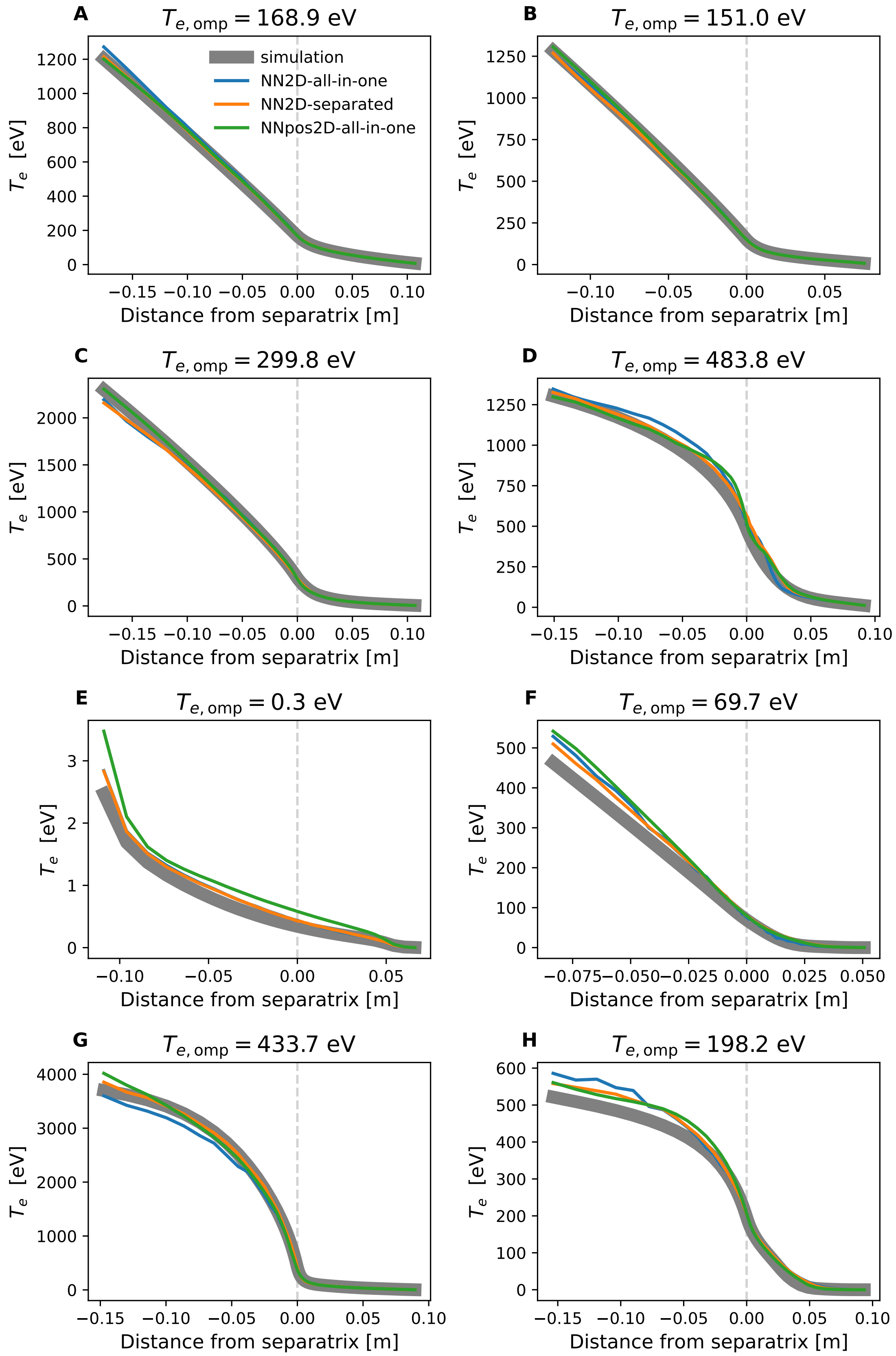


**Figure A.8:** Comparison between the electron temperature profiles predicted by the various models at the outer midplane for eight simulations from the test set. The eight examples are the same as in Figure 8.16. For reference the temperature at the outer midplane separatrix in each simulation is given.

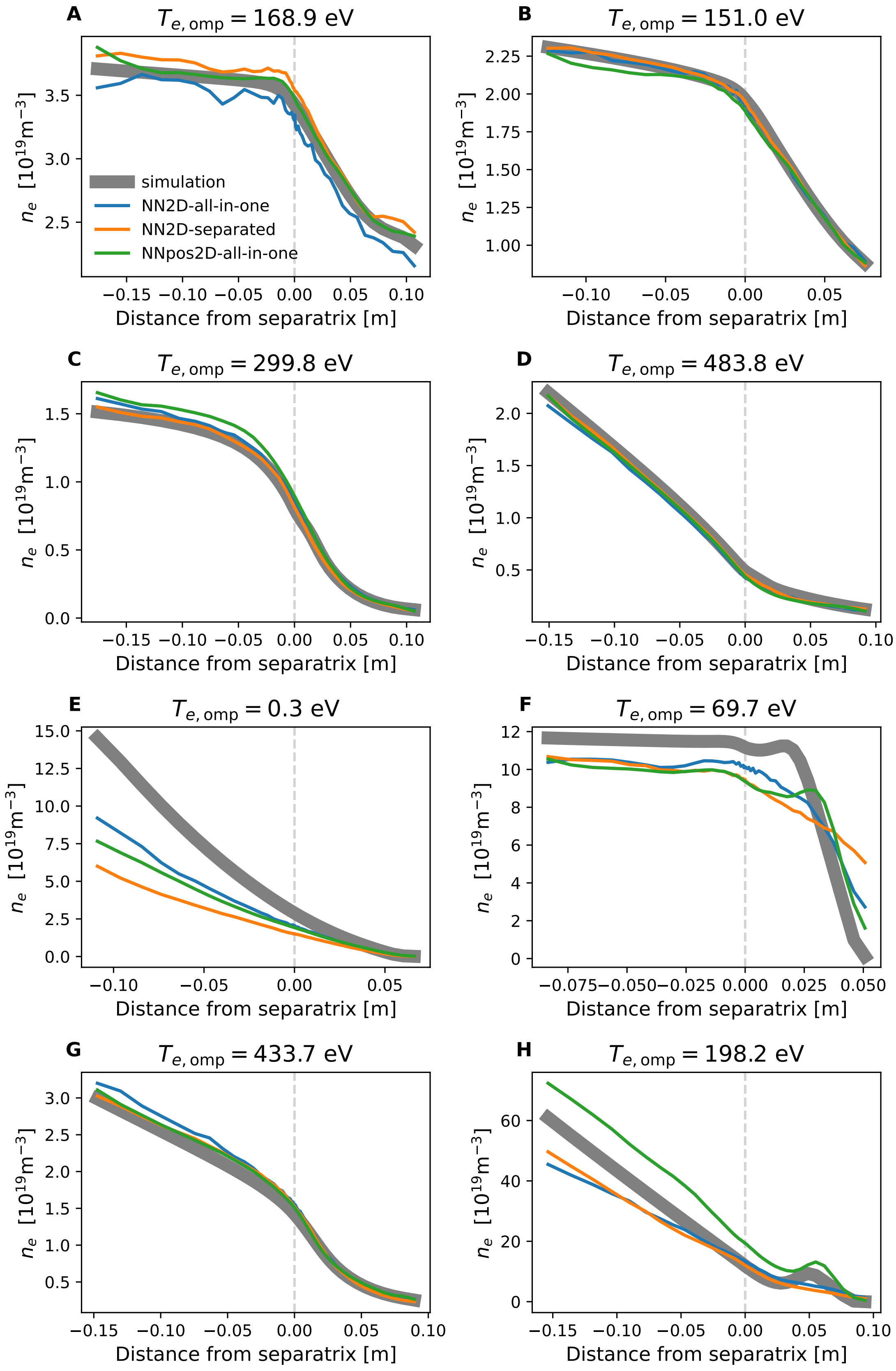


**Figure A.9:** Comparison between the electron density profiles predicted by the various models at the outer midplane for eight simulations from the test set. The eight examples are the same as in Figure 8.16. For reference the temperature at the outer midplane separatrix in each simulation is given.

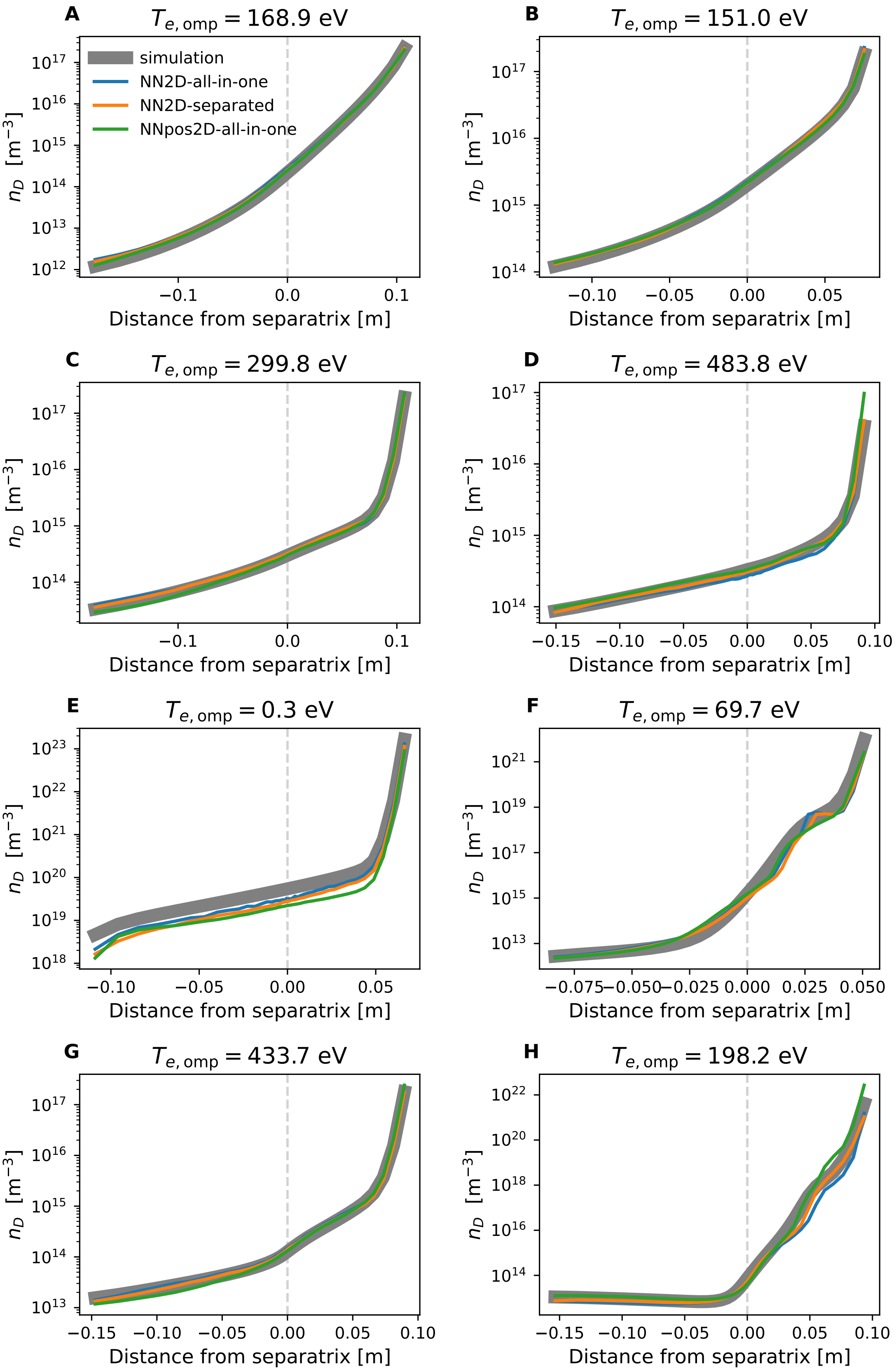


**Figure A.10:** Comparison between the neutral deuterium density profiles predicted by the various models at the outer midplane for eight simulations from the test set. The eight examples are the same as in Figure 8.16. For reference the temperature at the outer midplane separatrix in each simulation is given.

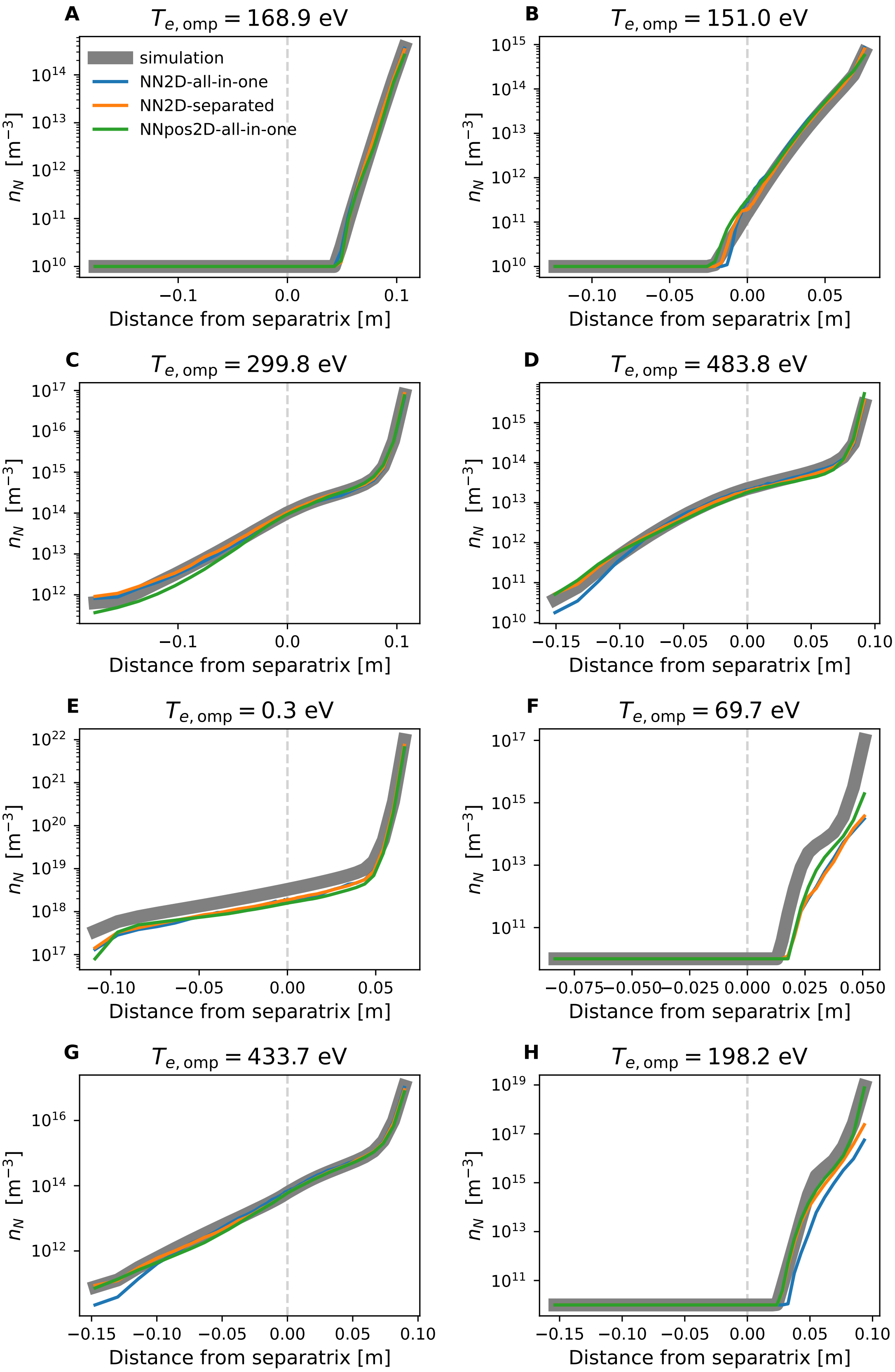


**Figure A.11:** Comparison between the neutral deuterium nitrogenn profiles predicted by the various models at the outer midplane for eight simulations from the test set. The eight examples are the same as in Figure 8.16. For reference the temperature at the outer midplane separatrix in each simulation is given.

## A.4 Predicting the fluid velocities

While the analysis in Section 8.5 showed, that it is beneficial to predict each plasma property by independent networks of similar architecture, the plasma properties this analysis was based on have in common that they are strictly positive scalar numbers. But beside the densities and the temperatures also also the fluid velocities are defininining variables of the plasma in SOLPS. These velocities can assume either positive or negative values depending on whether the fluid flow points in the direction of the magnetic field or in the opposite direction. This allows the network to make errors not only in the absolute value of the velocity but also in its sign. These errors should be inspected independently, as summed up in a median absolute error rate these sign errors might constitute only minor errors ( 200%) compared to the variation across several orders of magnitude present in the simulations. Additionally the quantile preprocessor might be less suited for the velocities. Therefore tests were conducted by training NN2D networks for the fluid velocities with varying preprocessing routines. **None** describes using the velocities as they are without any scaling. **Quantile** uses the standard quantile transformation as explained in Section 3.1.5. **Symlog** applies $u_{i,\text{symlog}} = sign(u_i) \cdot log(1 + |u_i|)$ to each velocity at each location $u_i$ to reduce the scale of the velocity. **Symlog2** applies first the same transformation as the Symlog scaling but then additionally applies the standard scaler (Section 3.1.5) on top $u_{i,\text{symlog2}} = (u_{i,\text{symlog}} - \mu(\mathbf{u}_{\text{symlog}}))/\sigma(\mathbf{u}_{\text{symlog}})$. Here the mean $\mu$ and standard deviation $\sigma$ are calculated independently for each velocity and each location in the simulation grid. In addition to these tests also an entirely different approach was taken were one NN2D network was trained to predict the absolute value of the velocity (of one ion species) while a different network was trained to predict the signs of the velocities. This network also follows the NN2D architecture but the last layer is composed of neurons with sigmoid activation functions and is trained with a binary crossentropy loss. Figure A.12 and Table A.3 compare the predicted velocities for the deuterium ions on the test set between the different models. It is clear that the best model in terms of the rate of incorrect signs and the median absolute errors, is the model that employs to distinct networks for the sign and the absolute value of the velocity. But the standard NN2D model with quantile preprocessor is only slightly worse in all metrics. Using either of the symlog based preprocessors or none at all will yield poorer model accuracy. For simplicity therefore the standard NN2D architecture with the quantile transformation is used for the prediction of velocities in Section 8.6.

| | Sign Errors | | Median Absolute Errors | | |
|---|---|---|---|---|---|
| | False positives | False negatives | All | True positives | True negatives |
| None | 602185 | 533618 | 2.1e+03 | 2.6e+03 | 2e+03 |
| Quantile | 284082 | 322901 | 1.1e+03 | 1.4e+03 | 9.3e+02 |
| Symlog | 335417 | 360361 | 1.7e+03 | 2.2e+03 | 1.5e+03 |
| Symlog2 | 324610 | 350794 | 1.8e+03 | 2.3e+03 | 1.5e+03 |
| Sign+Absolute | 283754 | 296142 | 1.2e+03 | 1.4e+03 | 9e+02 |

**Table A.3:** Prediction errors on the test set for the deuterium ion fluid velocity of NN2D models used with varying pre-processors. The errors are either wrongfully predicted positive or negative signs and the median errors. The median errors are either computed over all predictions or only those for which the correct sign was predicted. All test simulations and all grid cells in the simulation domain are considered in computing the errors.

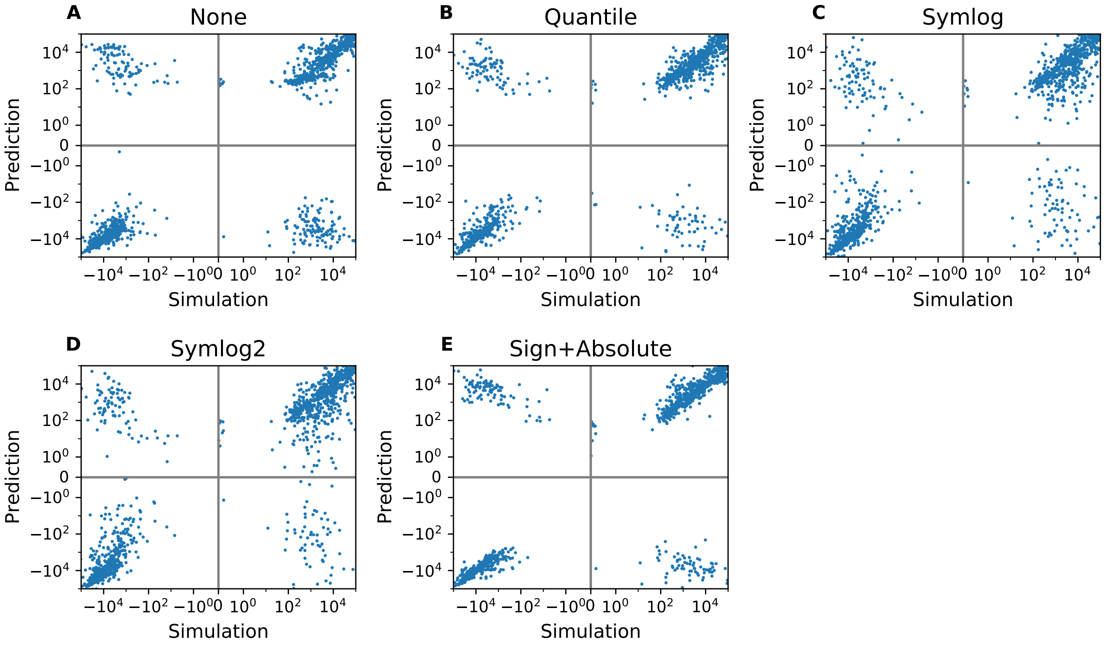


**Figure A.12:** The model predictions for the deuterium ion fluid velocity at the outer midplane separatrix against the results in the simulations in the test set. A model is more precise the closer the predictions are to the diagonal.

## A.5 Detachment scaling

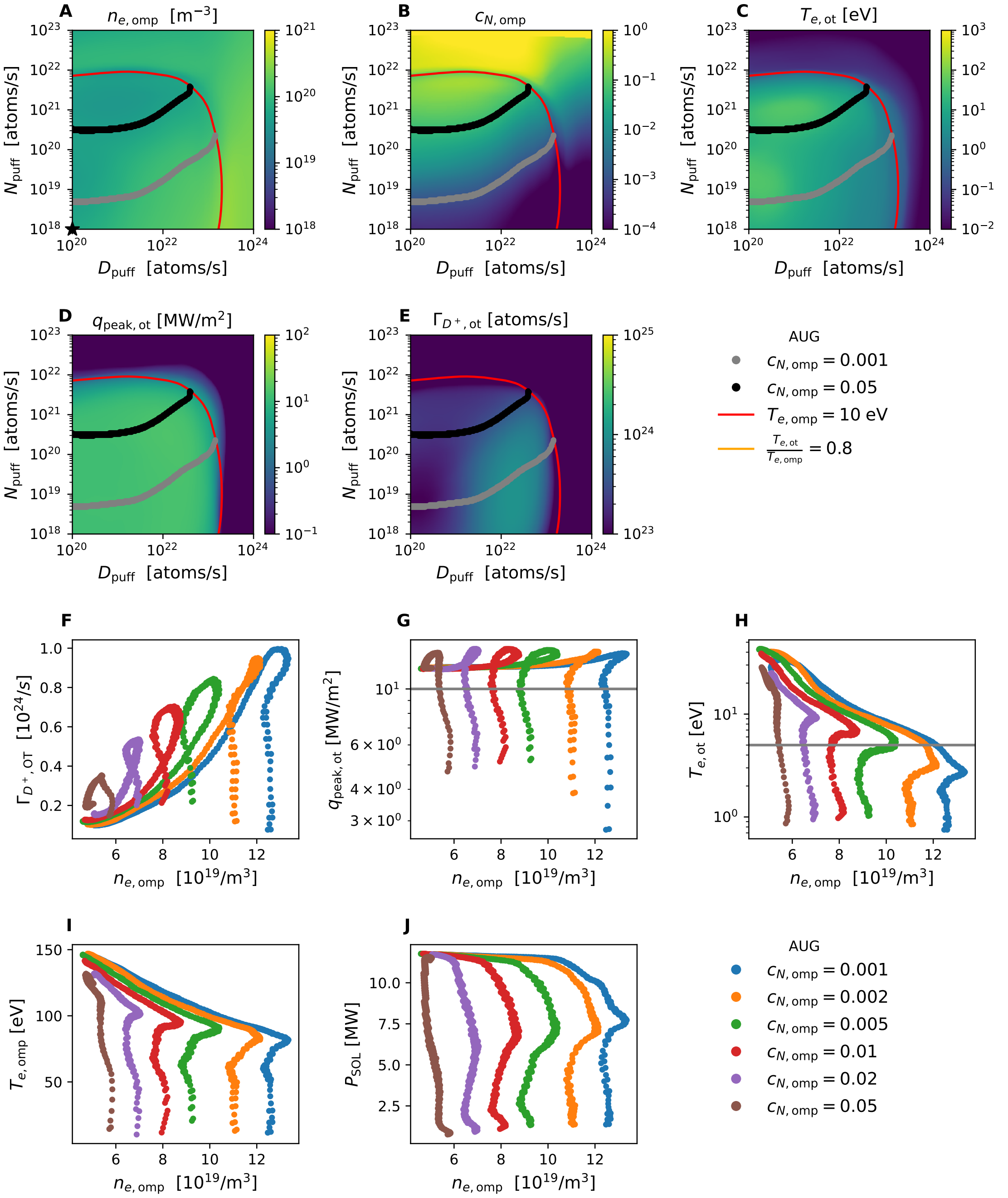


**Figure A.13:** Predictions of the surrogate model for a gas puff scan with **AUG** parameters and high core fueling. Shown are the resulting electron density $n_{e,\mathrm{omp}}$ (A) and nitrogen concentration $c_{N,\mathrm{omp}}$ (B) at the outer midplane separatrix and the outer target temperature $T_{e,\mathrm{ot}}$ (C) at the separatrix as well as the integrated deuterium ion flux $\Gamma_{D^+,\mathrm{ot}}$ (D) and peak heat flux $q_{\mathrm{peak,ot}}$ (E) towards the outer target. In A-E the the orange and red lines highlight the transition into the sheath-limited and cold core regimes while the grey and black dots show all scenarios with nitrogen concentrations $c_{N,\mathrm{omp}} = 0.1\%$ and $c_{N,\mathrm{omp}} = 5\%$. The black star in A highlights the deuterium gas puff necessary to leave the sheath-limited regime with minimum nitrogen puff. F-J show density scans selected from the gas puff scan above. Scenarios are selected based on the upstream nitrogen concentration (denoted by the color) and when they are not in a sheath-limited or in the cold core regime. Shown are the integrated deuterium ion flux $\Gamma_{D^+,\mathrm{ot}}$ (F) and peak heat flux $q_{\mathrm{peak,ot}}$ (G) towards the outer target, the electron temperature at the outer target $T_{e,\mathrm{ot}}$ (H) and outer midplane $T_{e,\mathrm{omp}}$ (I) separatrix and the power crossing the separatrix $P_{SOL}$ (J) from the main plasma.

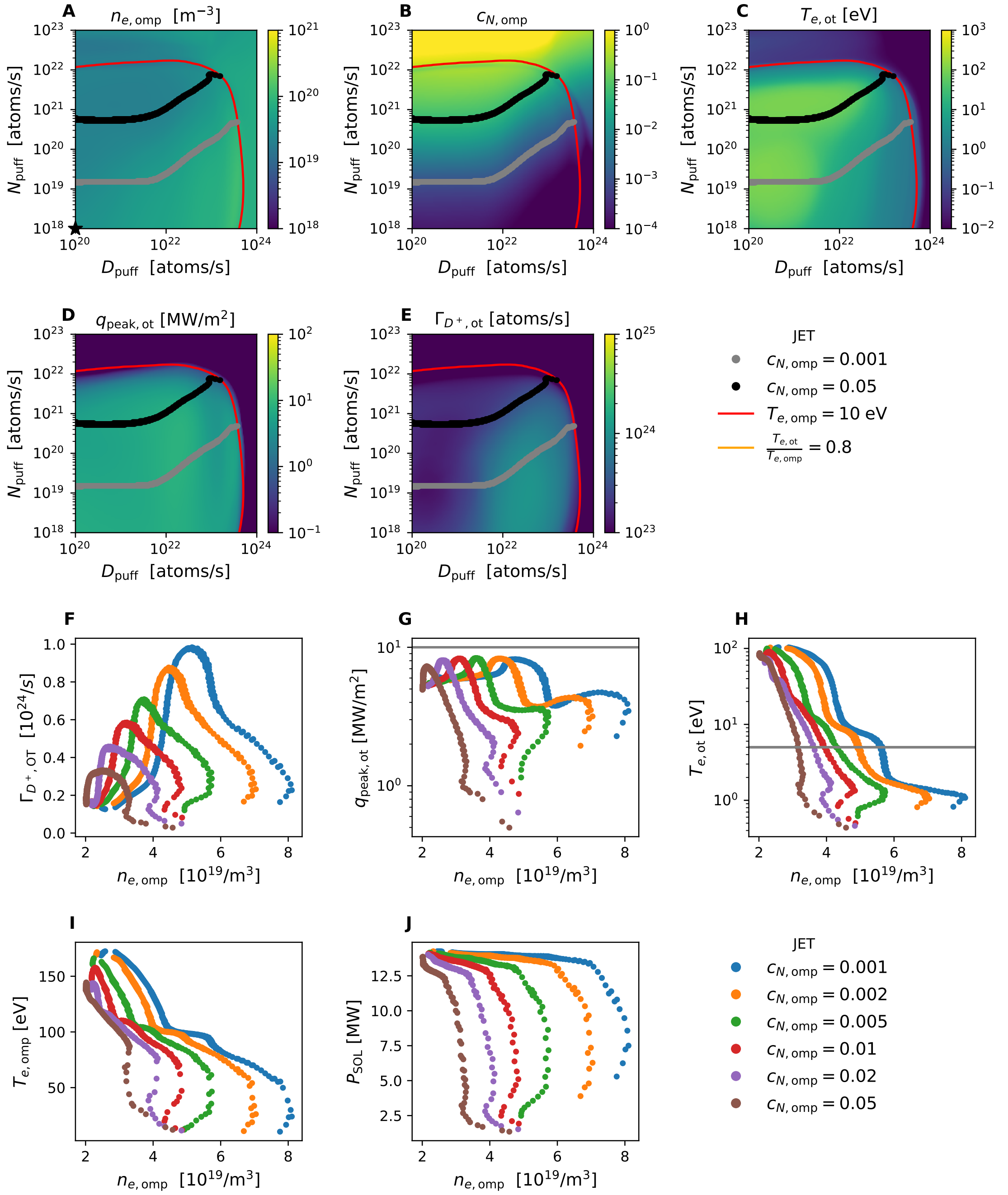


**Figure A.14:** Predictions of the surrogate model for a gas puff scan with **JET** parameters and high core fueling. Shown are the resulting electron density $n_{e,\mathrm{omp}}$ (A) and nitrogen concentration $c_{N,\mathrm{omp}}$ (B) at the outer midplane separatrix and the outer target temperature $T_{e,\mathrm{ot}}$ (C) at the separatrix as well as the integrated deuterium ion flux $\Gamma_{D^+,\mathrm{ot}}$ (D) and peak heat flux $q_{\mathrm{peak,ot}}$ (E) towards the outer target. In A-E the the orange and red lines highlight the transition into the sheath-limited and cold core regimes while the grey and black dots show all scenarios with nitrogen concentrations $c_{N,\mathrm{omp}} = 0.1\%$ and $c_{N,\mathrm{omp}} = 5\%$. The black star in A highlights the deuterium gas puff necessary to leave the sheath-limited regime with minimum nitrogen puff. F-J show density scans selected from the gas puff scan above. Scenarios are selected based on the upstream nitrogen concentration (denoted by the color) and when they are not in a sheath-limited or in the cold core regime. Shown are the integrated deuterium ion flux $\Gamma_{D^+,\mathrm{ot}}$ (F) and peak heat flux $q_{\mathrm{peak,ot}}$ (G) towards the outer target, the electron temperature at the outer target $T_{e,\mathrm{ot}}$ (H) and outer midplane $T_{e,\mathrm{omp}}$ (I) separatrix and the power crossing the separatrix $P_{SOL}$ (J) from the main plasma.

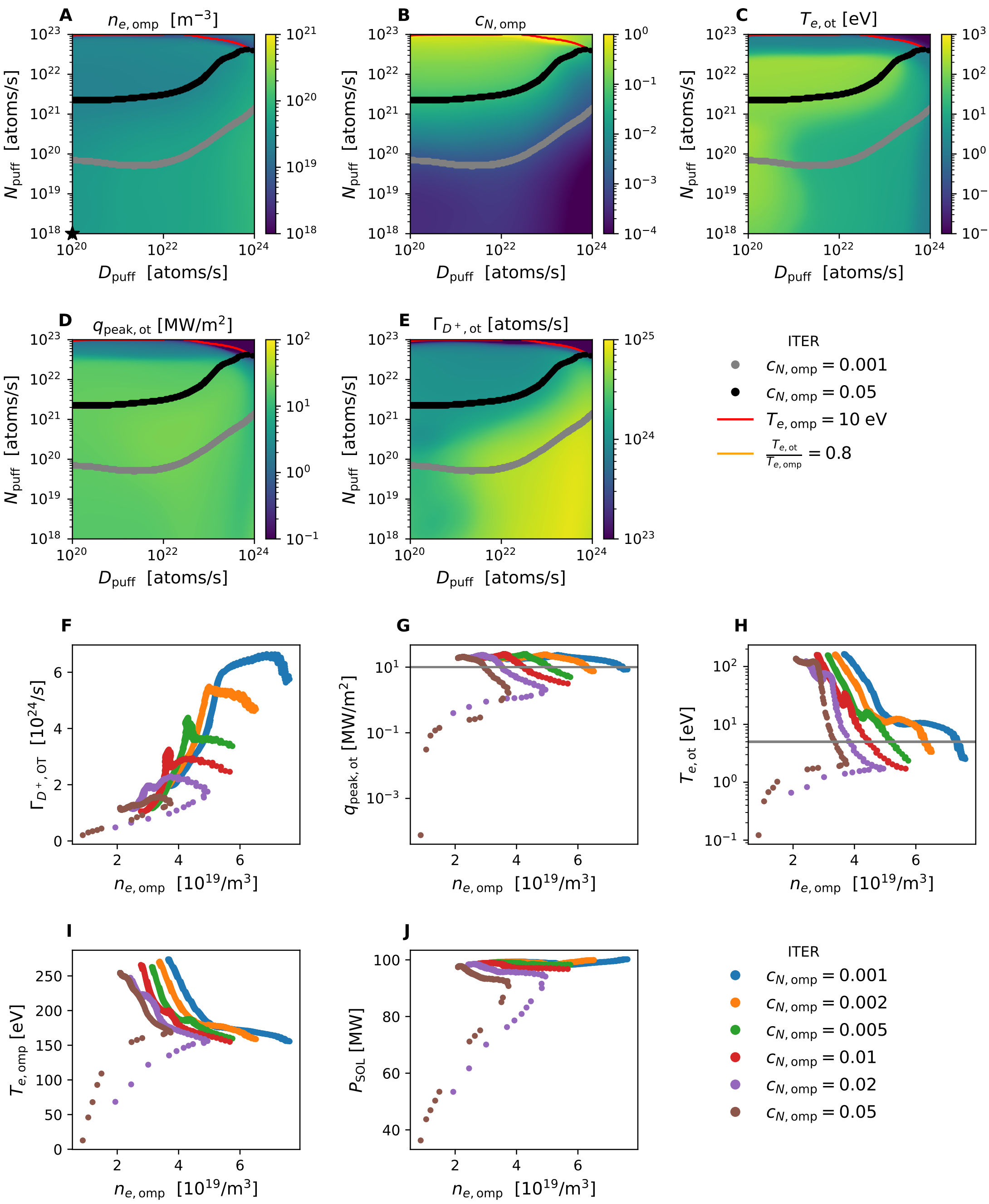


**Figure A.15:** Predictions of the surrogate model for a gas puff scan with **ITER** parameters and high core fueling. Shown are the resulting electron density $n_{e,\mathrm{omp}}$ (A) and nitrogen concentration $c_{N,\mathrm{omp}}$ (B) at the outer midplane separatrix and the outer target temperature $T_{e,\mathrm{ot}}$ (C) at the separatrix as well as the integrated deuterium ion flux $\Gamma_{D^+,\mathrm{ot}}$ (D) and peak heat flux $q_{\mathrm{peak,ot}}$ (E) towards the outer target. In A-E the the orange and red lines highlight the transition into the sheath-limited and cold core regimes while the grey and black dots show all scenarios with nitrogen concentrations $c_{N,\mathrm{omp}} = 0.1\%$ and $c_{N,\mathrm{omp}} = 5\%$. The black star in A highlights the deuterium gas puff necessary to leave the sheath-limited regime with minimum nitrogen puff. F-J show density scans selected from the gas puff scan above. Scenarios are selected based on the upstream nitrogen concentration (denoted by the color) and when they are not in a sheath-limited or in the cold core regime. Shown are the integrated deuterium ion flux $\Gamma_{D^+,\mathrm{ot}}$ (F) and peak heat flux $q_{\mathrm{peak,ot}}$ (G) towards the outer target, the electron temperature at the outer target $T_{e,\mathrm{ot}}$ (H) and outer midplane $T_{e,\mathrm{omp}}$ (I) separatrix and the power crossing the separatrix $P_{SOL}$ (J) from the main plasma.

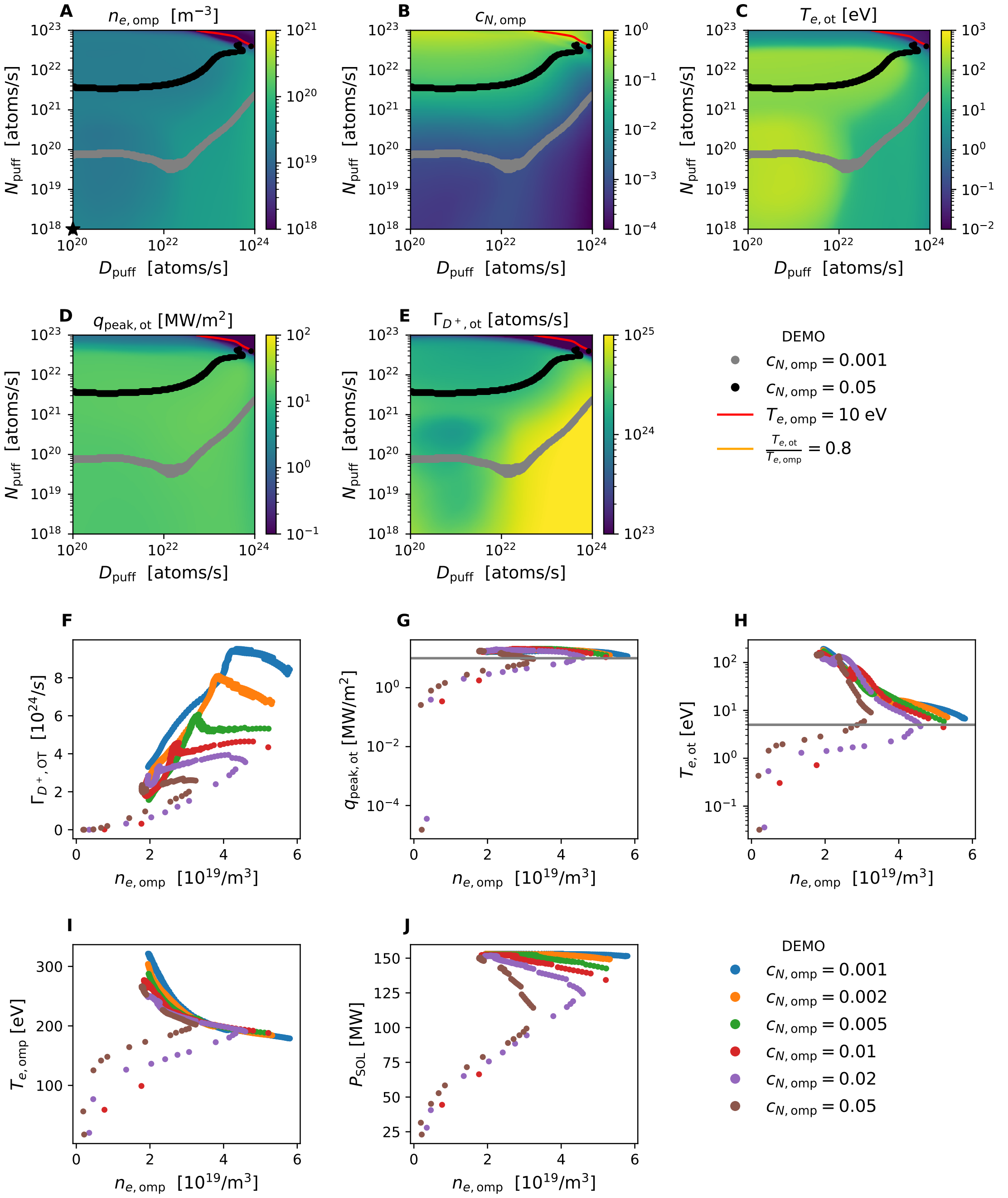
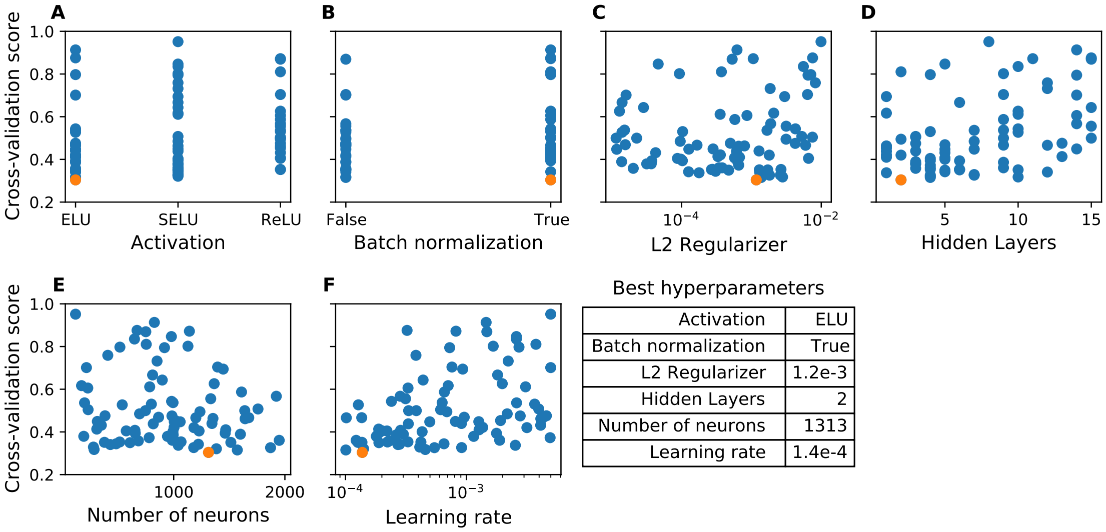


**Figure A.16:** Predictions of the surrogate model for a gas puff scan with **DEMO** parameters and high core fueling. Shown are the resulting electron density $n_{e,\mathrm{omp}}$ (A) and nitrogen concentration $c_{N,\mathrm{omp}}$ (B) at the outer midplane separatrix and the outer target temperature $T_{e,\mathrm{ot}}$ (C) at the separatrix as well as the integrated deuterium ion flux $\Gamma_{D^+,\mathrm{ot}}$ (D) and peak heat flux $q_{\mathrm{peak,ot}}$ (E) towards the outer target. In A-E the the orange and red lines highlight the transition into the sheath-limited and cold core regimes while the grey and black dots show all scenarios with nitrogen concentrations $c_{N,\mathrm{omp}} = 0.1\%$ and $c_{N,\mathrm{omp}} = 5\%$. The black star in A highlights the deuterium gas puff necessary to leave the sheath-limited regime with minimum nitrogen puff. F-J show density scans selected from the gas puff scan above. Scenarios are selected based on the upstream nitrogen concentration (denoted by the color) and when they are not in a sheath-limited or in the cold core regime. Shown are the integrated deuterium ion flux $\Gamma_{D^+,\mathrm{ot}}$ (F) and peak heat flux $q_{\mathrm{peak,ot}}$ (G) towards the outer target, the electron temperature at the outer target $T_{e,\mathrm{ot}}$ (H) and outer midplane $T_{e,\mathrm{omp}}$ (I) separatrix and the power crossing the separatrix $P_{SOL}$ (J) from the main plasma.

## A.6 Transfer learning



Best hyperparameters

| | |
|---|---|
| Activation | ELU |
| Batch normalization | True |
| L2 Regularizer | 1.2e-3 |
| Hidden Layers | 2 |
| Number of neurons | 1313 |
| Learning rate | 1.4e-4 |

**Figure A.17:** Cross-validation score of the NN2D model architecture trained from scratch on the ITER simulations with varying hyperparameters. Each dot represents one tested hyperparameter configuration in the random search. The best hyperparameter configuration is the one with the lowest score. Several configurations exhibit higher cross-validation scores than visible in these plots.

| Pulse | Run | Database | Reference | Workflow | Date |
|---|---|---|---|---|---|
| 122317 | 3 | ITER | F57-100-Ne_0.4%-Be0,D_tpt=1.85e23,Ne_tpt=2.20e20 | SOLPS4.3 | 2022-06-10 |
| 122318 | 1 | ITER | F57-80-Ne_0.4%-Be0,D_tpt=8.65e22,Ne_tpt=6.00e19 | SOLPS4.3 | 2022-06-10 |
| 122319 | 1 | ITER | F57-80-NF-Ne-Be0,D_tpt=1.20e23,Ne_tpt=9.50e19 | SOLPS4.3 | 2022-06-10 |
| 122321 | 3 | ITER | F57-120-Ne_0.4%-Be0,D_tpt=5.47e22,Ne_tpt=3.70e19 | SOLPS4.3 | 2022-06-10 |
| 122332 | 3 | ITER | F57-100-Ne_0.8%-Be0,D_tpt=8.20e22,Ne_tpt=1.20e20 | SOLPS4.3 | 2022-06-10 |
| 122388 | 3 | ITER | F57-60-Ne_0.4%-Be0,D_tpt=1.45e22,Ne_tpt=3.50e19 | SOLPS4.3 | 2022-06-10 |
| 122389 | 1 | ITER | F57-60-Ne_0.4%-Be0,D_tpt=1.91e22,Ne_tpt=3.50e19 | SOLPS4.3 | 2022-06-10 |
| 122390 | 1 | ITER | F57-60-0.4%_Ne-Be0,D_tpt=3.40e22,Ne_tpt=4.00e19 | SOLPS4.3 | 2022-06-10 |
| 122391 | 3 | ITER | F57-60-Ne_0.4%-Be0,D_tpt=6.10e22,Ne_tpt=5.00e19 | SOLPS4.3 | 2022-06-10 |
| 122392 | 1 | ITER | F57-60-Ne_0.4%-Be0,D_tpt=8.45e22,Ne_tpt=6.50e19 | SOLPS4.3 | 2022-06-10 |
| 122393 | 1 | ITER | F57-60-Ne_0.4%-Be0,D_tpt=1.15e23,Ne_tpt=9.00e19 | SOLPS4.3 | 2022-06-10 |
| 122394 | 1 | ITER | F57-60-Ne_0.4%-Be0,D_tpt=1.59e23,Ne_tpt=1.10e20 | SOLPS4.3 | 2022-06-10 |
| 122395 | 1 | ITER | F57-120-Ne_0.4%-Be0,D_tpt=2.30e23,Ne_tpt=1.50e20 | SOLPS4.3 | 2022-06-10 |
| 122396 | 3 | ITER | F57-100-Ne_0.6%-Be0,D_tpt=1.05e22,Ne_tpt=6.00e19 | SOLPS4.3 | 2022-06-10 |
| 122397 | 3 | ITER | F57-100-Ne_0.6%-Be0,D_tpt=1.76e22,Ne_tpt=7.00e19 | SOLPS4.3 | 2022-06-10 |
| 122398 | 3 | ITER | F57-100-Ne_0.6%-Be0,D_tpt=3.10e22,Ne_tpt=7.00e19 | SOLPS4.3 | 2022-06-10 |
| 122399 | 3 | ITER | F57-100-Ne_0.6%-Be0,D_tpt=4.90e22,Ne_tpt=1.00e20 | SOLPS4.3 | 2022-06-10 |
| 122400 | 6 | ITER | F57-100-Ne_0.6%-Be0,D_tpt=7.70e22,Ne_tpt=1.00e20 | SOLPS-ITER | 2022-06-10 |
| 122401 | 6 | ITER | F57-100-Ne_0.6%-Be0,D_tpt=1.20e23,Ne_tpt=1.20e20 | SOLPS-ITER | 2022-06-10 |
| 122402 | 3 | ITER | F57-100-Ne_0.6%-Be0,D_tpt=1.40e23,Ne_tpt=1.50e20 | SOLPS4.3 | 2022-06-10 |
| 122403 | 3 | ITER | F57-100-Ne_0.6%-Be0,D_tpt=1.67e23,Ne_tpt=1.20e20 | SOLPS4.3 | 2022-06-10 |
| 122404 | 6 | ITER | F57-100-Ne_0.6%-Be0,D_tpt=2.10e23,Ne_tpt=2.00e20 | SOLPS-ITER | 2022-06-10 |
| 122405 | 1 | ITER | F57-60-Ne_0.4%-Be0,D_tpt=1.00e23,Ne_tpt=9.00e19 | SOLPS4.3 | 2022-06-10 |
| 122407 | 3 | ITER | F57-100-Ne_0.8%-Be0,D_tpt=4.77e22,Ne_tpt=1.50e20 | SOLPS4.3 | 2022-06-10 |
| 122408 | 3 | ITER | F57-100-Ne_0.8%-Be0,D_tpt=1.76e22,Ne_tpt=8.00e19 | SOLPS4.3 | 2022-06-10 |
| 122409 | 3 | ITER | F57-100-Ne_0.3%-Be0,D_tpt=1.32e23,Ne_tpt=6.50e19 | SOLPS4.3 | 2022-06-10 |
| 122410 | 3 | ITER | F57-100-Ne_0.3%-Be0,D_tpt=7.50e22,Ne_tpt=5.50e19 | SOLPS4.3 | 2022-06-10 |
| 122411 | 3 | ITER | F57-100-Ne_0.3%-Be0,D_tpt=6.50e22,Ne_tpt=5.50e19 | SOLPS4.3 | 2022-06-10 |
| 122412 | 3 | ITER | F57-100-Ne_0.3%-Be0,D_tpt=4.45e22,Ne_tpt=3.00e19 | SOLPS4.3 | 2022-06-10 |
| 122413 | 3 | ITER | F57-100-Ne_0.3%-Be0,D_tpt=3.40e22,Ne_tpt=4.50e19 | SOLPS4.3 | 2022-06-10 |
| 122414 | 3 | ITER | F57-100-Ne_0.3%-Be0,D_tpt=2.45e22,Ne_tpt=4.00e19 | SOLPS4.3 | 2022-06-10 |
| 122415 | 3 | ITER | F57-100-Ne_0.3%-Be0,D_tpt=1.95e23,Ne_tpt=1.00e20 | SOLPS4.3 | 2022-06-10 |
| 122416 | 3 | ITER | F57-100-Ne_0.3%-Be0,D_tpt=2.60e23,Ne_tpt=1.50e20 | SOLPS4.3 | 2022-06-10 |
| 122419 | 1 | ITER | F57-60-Ne_0.4%-Be0,D_tpt=7.48e21,Ne_tpt=3.00e19 | SOLPS4.3 | 2022-06-10 |
| 122437 | 1 | ITER | F57-080-Ne_0.4%-Be0,D_tpt=1.62e23,Ne_tpt=1.20e20 | SOLPS4.3 | 2022-06-10 |
| 122438 | 1 | ITER | F57-080-Ne_0.4%-Be0,D_tpt=2.15e23,Ne_tpt=1.50e20 | SOLPS4.3 | 2022-06-10 |
| 122439 | 3 | ITER | F57-100-Ne_0.4%-Be0,D_tpt=9.00e22,Ne_tpt=7.00e19 | SOLPS4.3 | 2022-06-10 |
| 122463 | 3 | ITER | F57-100-Ne_1.2%-Be0,D_tpt=1.21e23,Ne_tpt=2.80e20 | SOLPS4.3 | 2022-06-10 |
| 122467 | 3 | ITER | F57-100-Ne_1.2%-Be0,D_tpt=2.77e23,Ne_tpt=6.00e20 | SOLPS4.3 | 2022-06-10 |
| 122468 | 3 | ITER | F57-100-Ne_1.2%-Be0,D_tpt=1.87e23,Ne_tpt=3.50e20 | SOLPS4.3 | 2022-06-10 |
| 122469 | 3 | ITER | F57-100-Ne_1.2%-Be0,D_tpt=1.38e23,Ne_tpt=3.00e20 | SOLPS4.3 | 2022-06-10 |
| 122470 | 3 | ITER | F57-100-Ne_1.2%-Be0,D_tpt=9.00e22,Ne_tpt=2.50e20 | SOLPS4.3 | 2022-06-10 |
| 122471 | 3 | ITER | F57-100-Ne_1.2%-Be0,D_tpt=6.40e22,Ne_tpt=2.20e20 | SOLPS4.3 | 2022-06-10 |
| 122472 | 3 | ITER | F57-100-Ne_1.2%-Be0,D_tpt=3.75e22,Ne_tpt=2.00e20 | SOLPS4.3 | 2022-06-10 |
| 122476 | 3 | ITER | F57-100-Ne_1.2%-Be0,D_tpt=2.44e22,Ne_tpt=1.50e20 | SOLPS4.3 | 2022-06-10 |
| 122477 | 3 | ITER | F57-100-Ne_1.2%-Be0,D_tpt=1.16e22,Ne_tpt=1.20e20 | SOLPS4.3 | 2022-06-10 |
| 122478 | 3 | ITER | F57-100-Ne_1.2%-Be0,D_tpt=5.50e21,Ne_tpt=1.50e20 | SOLPS4.3 | 2022-06-10 |
| 122481 | 3 | ITER | F57-100-Ne_1.8%-Be0,D_tpt=1.03e23,Ne_tpt=4.40e20 | SOLPS4.3 | 2022-06-10 |
| 122483 | 3 | ITER | F57-100-Ne_1.8%-Be0,D_tpt=7.80e22,Ne_tpt=4.00e20 | SOLPS4.3 | 2022-06-10 |
| 122484 | 3 | ITER | F57-100-Ne_1.8%-Be0,D_tpt=1.21e23,Ne_tpt=5.50e20 | SOLPS4.3 | 2022-06-10 |
| 122485 | 3 | ITER | F57-100-Ne_1.8%-Be0,D_tpt=1.81e23,Ne_tpt=6.00e20 | SOLPS4.3 | 2022-06-10 |
| 122496 | 3 | ITER | F57-100-Ne_1.8%-Be0,D_tpt=5.54e22,Ne_tpt=3.50e20 | SOLPS4.3 | 2022-06-10 |
| 122497 | 3 | ITER | F57-100-Ne_1.8%-Be0,D_tpt=3.64e22,Ne_tpt=3.00e20 | SOLPS4.3 | 2022-06-10 |
| 122498 | 3 | ITER | F57-100-Ne_1.8%-Be0,D_tpt=2.48e22,Ne_tpt=2.5e20 | SOLPS4.3 | 2022-06-10 |
| 122506 | 3 | ITER | F57-080-Ne_1.2%-Be0,D_tpt=5.90e22,Ne_tpt=2.00e20 | SOLPS4.3 | 2022-06-10 |
| 122507 | 1 | ITER | F57-120-Ne_1.2%-Be0,D_tpt=2.82e22,Ne_tpt=1.80e20 | SOLPS4.3 | 2022-06-10 |
| 122508 | 3 | ITER | F57-100-Ne_1.8%-Be0,D_tpt=1.68e22,Ne_tpt=1.70e20 | SOLPS4.3 | 2022-06-10 |
| 122509 | 3 | ITER | F57-100-Ne_1.8%-Be0,D_tpt=7.50e21,Ne_tpt=1.20e20 | SOLPS4.3 | 2022-06-10 |
| 122512 | 1 | ITER | F57-120-Ne_1.2%-Be0,D_tpt=4.20e22,Ne_tpt=2.30e20 | SOLPS4.3 | 2022-06-10 |
| 122513 | 1 | ITER | F57-120-Ne_1.2%-Be0,D_tpt=6.63e22,Ne_tpt=2.70e20 | SOLPS4.3 | 2022-06-10 |
| 122514 | 1 | ITER | F57-080-Ne_1.2%-Be0,D_tpt=4.00e22,Ne_tpt=1.50e20 | SOLPS4.3 | 2022-06-10 |
| 122515 | 1 | ITER | F57-080-Ne_1.2%-Be0,D_tpt=2.50e22,Ne_tpt=1.30e20 | SOLPS4.3 | 2022-06-10 |
| 122518 | 1 | ITER | F57-080-Ne_1.2%-Be0,D_tpt=1.50e22,Ne_tpt=8.00e19 | SOLPS4.3 | 2022-06-10 |
| 122519 | 1 | ITER | F57-120-Ne_1.2%-Be0,D_tpt=9.50e22,Ne_tpt=3.00e20 | SOLPS4.3 | 2022-06-10 |
| 122520 | 1 | ITER | F57-120-Ne_1.2%-Be0,D_tpt=1.25e23,Ne_tpt=3.80e20 | SOLPS4.3 | 2022-06-10 |
| 122521 | 1 | ITER | F57-120-Ne_1.2%-Be0,D_tpt=1.65e23,Ne_tpt=4.00e20 | SOLPS4.3 | 2022-06-10 |
| 122522 | 1 | ITER | F57-080-Ne_1.2%-Be0,D_tpt=7.50e21,Ne_tpt=1.00e20 | SOLPS4.3 | 2022-06-10 |
| 122523 | 1 | ITER | F57-080-Ne_1.2%-Be0,D_tpt=3.50e21,Ne_tpt=1.00e20 | SOLPS4.3 | 2022-06-10 |
| 122524 | 1 | ITER | F57-060-Ne_1.2%-Be0,D_tpt=3.25e22,Ne_tpt=1.10e20 | SOLPS4.3 | 2022-06-10 |
| 122535 | 1 | ITER | F57-060-Ne_1.2%-Be0,D_tpt=1.90e22,Ne_tpt=1.00e20 | SOLPS4.3 | 2022-06-10 |
| 122546 | 1 | ITER | F57-120-Ne_1.2%-Be0,D_tpt=3.50e22,Ne_tpt=0.00 | SOLPS4.3 | 2022-06-10 |
| 122547 | 1 | ITER | F57-120-Ne_1.2%-Be0,D_tpt=8.50e21,Ne_tpt=1.20e20 | SOLPS4.3 | 2022-06-10 |
| 122548 | 1 | ITER | F57-080-Ne_1.2%-Be0,D_tpt=8.50e22,Ne_tpt=2.20e20 | SOLPS4.3 | 2022-06-10 |
| 122549 | 1 | ITER | F57-080-Ne_1.2%-Be0,D_tpt=1.12e23,Ne_tpt=2.50e20 | SOLPS4.3 | 2022-06-10 |
| 122560 | 1 | ITER | F57-060-Ne_1.2%-Be0,D_tpt=1.20e22,Ne_tpt=7.00e19 | SOLPS4.3 | 2022-06-10 |
| 122576 | 1 | ITER | F57-120-Ne_1.2%-Be0,D_tpt=2.00e23,Ne_tpt=5.60e20 | SOLPS4.3 | 2022-06-10 |
| 122584 | 1 | ITER | F57-080-Ne_1.2%-Be0,D_tpt=1.47e23,Ne_tpt=2.5e20 | SOLPS4.3 | 2022-06-10 |
| 122585 | 1 | ITER | F57-080-Ne_1.2%-Be0,D_tpt=1.72e23,Ne_tpt=3.0e20 | SOLPS4.3 | 2022-06-10 |

**Table A.4:** Overview of all ITER simulations from the IMAS database used in Chapter 10. In cases where multiple SOLPS runs for a given Pulse number were present on the ITER file system at the time of retrieval, it cannot be guaranteed that always the run number given in this table is used but instead potentially older run versions might have been used.

# Bibliography

[1] S. Dasbach and S. Wiesen, “Towards fast surrogate models for interpolation of tokamak edge plasmas”, *Nuclear Materials and Energy*, vol. 34, p. 101 396, Mar. 2023, ISSN: 2352-1791. DOI: `10.1016/j.nme.2023.101396`.

[2] S. Wiesen *et al.*, “Data-driven models in fusion exhaust: AI methods and perspectives”, *Nuclear Fusion*, vol. 64, no. 8, p. 086 046, Jul. 2024, ISSN: 0029-5515. DOI: `10.1088/1741-4326/ad5a1d`.

[3] K. Calvin *et al.*, “IPCC, 2023: Climate Change 2023: Synthesis Report. Contribution of Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change [Core Writing Team, H. Lee and J. Romero (eds.)]. IPCC, Geneva, Switzerland.”, Intergovernmental Panel on Climate Change (IPCC), Tech. Rep., Jul. 2023. DOI: `10.59327/IPCC/AR6-9789291691647`.

[4] M. Ge, J. Friedrich, and L. Vigna, “4 Charts Explain Greenhouse Gas Emissions by Countries and Sectors”, Feb. 2020.

[5] C. Wolfram, O. Shelef, and P. Gertler, “How Will Energy Demand Develop in the Developing World?”, *Journal of Economic Perspectives*, vol. 26, no. 1, pp. 119–138, Feb. 2012, ISSN: 0895-3309. DOI: `10.1257/jep.26.1.119`.

[6] T. Ahmad and D. Zhang, “A critical review of comparative global historical energy consumption and future demand: The story told so far”, *Energy Reports*, vol. 6, pp. 1973–1991, Nov. 2020, ISSN: 2352-4847. DOI: `10.1016/j.egyr.2020.07.020`.

[7] M. Moynihan and A. B. Bortz, *Fusion’s Promise: How Technological Breakthroughs in Nuclear Fusion Can Conquer Climate Change on Earth (And Carry Humans To Mars, Too).* Cham: Springer International Publishing, 2023, ISBN: 978-3-031-22906-0. DOI: `10.1007/978-3-031-22906-0`.

[8] J. P. Freidberg, *Plasma Physics and Fusion Energy.* Cambridge: Cambridge University Press, 2007, ISBN: 978-0-521-73317-5. DOI: `10.1017/CBO9780511755705`.

[9] M. J. Gorley, “Critical Assessment 12: Prospects for reduced activation steel for fusion plant”, *Materials Science and Technology*, vol. 31, no. 8, pp. 975–980, Jun. 2015, ISSN: 0267-0836. DOI: `10.1179/1743284714Y.0000000732`.

[10] M. R. Gilbert *et al.*, “Waste implications from minor impurities in European DEMO materials”, *Nuclear Fusion*, vol. 59, no. 7, p. 076 015, May 2019, ISSN: 0029-5515. DOI: `10.1088/1741-4326/ab154e`.

[11] G. Federici, W. Biel, M. R. Gilbert, R. Kemp, N. Taylor, and R. Wenninger, "European DEMO design strategy and consequences for materials", *Nuclear Fusion*, vol. 57, no. 9, p. 092 002, Jun. 2017, ISSN: 0029-5515. DOI: `10.1088/1741-4326/57/9/092002`.

[12] T. E. G. Nicholas *et al.*, "Re-examining the role of nuclear fusion in a renewables-based energy mix", *Energy Policy*, vol. 149, p. 112 043, Feb. 2021, ISSN: 0301-4215. DOI: `10.1016/j.enpol.2020.112043`.

[13] J. Wesson and D. J. Campbell, *Tokamaks* (Oxford science publications 118), 3rd ed. Oxford : New York: Clarendon Press ; Oxford University Press, 2004, ISBN: 978-0-19-850922-6.

[14] F. F. Chen and F. F. Chen, *Introduction to plasma physics and controlled fusion*, 2nd ed. New York: Plenum Press, 1984, ISBN: 978-0-306-41332-2.

[15] R. J. Goldston and P. H. Rutherford, *Introduction to plasma physics*. Bristol, UK ; Philadelphia: Institute of Physics Pub, 1995, ISBN: 978-0-7503-0183-1.

[16] P. C. Stangeby, *The plasma boundary of magnetic fusion devices* (Plasma physics series). Bristol: Institute of Physics Pub, 2000, ISBN: 978-0-7503-0559-4.

[17] W. J. Huang, M. Wang, F. G. Kondev, G. Audi, and S. Naimi, "The AME 2020 atomic mass evaluation (I). Evaluation of input data, and adjustment procedures*", *Chinese Physics C*, vol. 45, no. 3, p. 030 002, Mar. 2021, ISSN: 1674-1137. DOI: `10.1088/1674-1137/abddb0`.

[18] M. Wang, W. J. Huang, F. G. Kondev, G. Audi, and S. Naimi, "The AME 2020 atomic mass evaluation (II). Tables, graphs and references*", *Chinese Physics C*, vol. 45, no. 3, p. 030 003, Mar. 2021, ISSN: 1674-1137. DOI: `10.1088/1674-1137/abddaf`.

[19] S. Meschini *et al.*, "Review of commercial nuclear fusion projects", *Frontiers in Energy Research*, vol. 11, 2023, ISSN: 2296-598X.

[20] D. A. Brown *et al.*, "ENDF/B-VIII.0: The 8th Major Release of the Nuclear Reaction Data Library with CIELO-project Cross Sections, New Standards and Thermal Scattering Data", *Nuclear Data Sheets*, Special Issue on Nuclear Reaction Data, vol. 148, pp. 1–142, Feb. 2018, ISSN: 0090-3752. DOI: `10.1016/j.nds.2018.02.001`.

[21] N. Otuka *et al.*, "Towards a More Complete and Accurate Experimental Nuclear Reaction Data Library (EXFOR): International Collaboration Between Nuclear Reaction Data Centres (NRDC)", *Nuclear Data Sheets*, vol. 120, pp. 272–276, Jun. 2014, ISSN: 0090-3752. DOI: `10.1016/j.nds.2014.07.065`.

[22] F. Mehmood, T. Kamal, and U. Ashraf, *Generation and Applications of Plasma (An Academic Review)*, Oct. 2018. DOI: `10.20944/preprints201810.0061.v1`. [Online]. Available: `https://www.preprints.org/manuscript/201810.0061/v1`.

[23] ITER Physics Expert Groups, ITER Physics Basis Editors, and ITER EDA, "Chapter 2: Plasma confinement and transport", *Nuclear Fusion*, vol. 39, no. 12, p. 2175, Dec. 1999, ISSN: 0029-5515. DOI: `10.1088/0029-5515/39/12/302`.

[24] H. Zohm *et al.*, "On the physics guidelines for a tokamak DEMO", *Nuclear Fusion*, vol. 53, no. 7, p. 073 019, Jun. 2013, ISSN: 0029-5515. DOI: `10.1088/0029-5515/53/7/073019`.

[25] S. Li, H. Jiang, Z. Ren, and C. Xu, "Optimal Tracking for a Divergent-Type Parabolic PDE System in Current Profile Control", *Abstract and Applied Analysis*, vol. 2014, e940965, Jun. 2014, ISSN: 1085-3375. DOI: `10.1155/2014/940965`.

[26] M. Barbarino, "A brief history of nuclear fusion", *Nature Physics*, vol. 16, no. 9, pp. 890–893, Sep. 2020, ISSN: 1745-2481. DOI: `10.1038/s41567-020-0940-7`.

[27] R. C. Wolf *et al.*, "Major results from the first plasma campaign of the Wendelstein 7-X stellarator", *Nuclear Fusion*, vol. 57, no. 10, p. 102 020, Jul. 2017, ISSN: 0029-5515. DOI: `10.1088/1741-4326/aa770d`.

[28] H. Yamada, "Overview of results from the Large Helical Device", *Nuclear Fusion*, vol. 51, no. 9, p. 094 021, Aug. 2011, ISSN: 0029-5515. DOI: `10.1088/0029-5515/51/9/094021`.

[29] B. N. Wan *et al.*, "Recent advances in EAST physics experiments in support of steady-state operation for ITER and CFETR", *Nuclear Fusion*, vol. 59, no. 11, p. 112 003, Jun. 2019, ISSN: 0029-5515. DOI: `10.1088/1741-4326/ab0396`.

[30] ITER Organization, "ITER Research Plan within the Staged Approach (Level III – Provisional Version)", Tech. Rep., Sep. 2018.

[31] F. Wagner, "A quarter-century of H-mode studies", *Plasma Physics and Controlled Fusion*, vol. 49, no. 12B, B1, Nov. 2007, ISSN: 0741-3335. DOI: `10.1088/0741-3335/49/12B/S01`.

[32] M. Greenwald *et al.*, "A new look at density limits in tokamaks", *Nuclear Fusion*, vol. 28, no. 12, p. 2199, Dec. 1988, ISSN: 0029-5515. DOI: `10.1088/0029-5515/28/12/009`.

[33] T. C. Hender *et al.*, "Chapter 3: MHD stability, operational limits and disruptions", *Nuclear Fusion*, vol. 47, no. 6, S128, Jun. 2007, ISSN: 0029-5515. DOI: `10.1088/0029-5515/47/6/S03`.

[34] P. C. d. Vries *et al.*, "Survey of disruption causes at JET", *Nuclear Fusion*, vol. 51, no. 5, p. 053 018, Apr. 2011, ISSN: 0029-5515. DOI: `10.1088/0029-5515/51/5/053018`.

[35] R. Wenninger *et al.*, "Advances in the physics basis for the European DEMO design", *Nuclear Fusion*, vol. 55, no. 6, p. 063 003, Apr. 2015, ISSN: 0029-5515. DOI: `10.1088/0029-5515/55/6/063003`.

[36] Y. Sarazin *et al.*, "Impact of scaling laws on tokamak reactor dimensioning", *Nuclear Fusion*, vol. 60, no. 1, p. 016 010, Oct. 2019, ISSN: 0029-5515. DOI: `10.1088/1741-4326/ab48a5`.

[37] H. Zohm, "On the size of tokamak fusion power plants", *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 377, no. 2141, p. 20 170 437, Mar. 2019. DOI: `10.1098/rsta.2017.0437`.

[38] A. J. Creely *et al.*, “Overview of the SPARC tokamak”, *Journal of Plasma Physics*, vol. 86, no. 5, p. 865 860 502, Oct. 2020, ISSN: 0022-3778, 1469-7807. DOI: `10.1017/S0022377820001257`.

[39] P. Bruzzone *et al.*, “High temperature superconductors for fusion magnets”, *Nuclear Fusion*, vol. 58, no. 10, p. 103 001, Aug. 2018, ISSN: 0029-5515. DOI: `10.1088/1741-4326/aad835`.

[40] Q. Wang *et al.*, “Progress of ultra-high-field superconducting magnets in China”, *Superconductor Science and Technology*, vol. 35, no. 2, p. 023 001, Dec. 2021, ISSN: 0953-2048. DOI: `10.1088/1361-6668/ac3f9b`.

[41] Y. Zhai, D. van der Laan, P. Connolly, and C. Kessel, “Conceptual design of HTS magnets for fusion nuclear science facility”, *Fusion Engineering and Design*, vol. 168, p. 112 611, Jul. 2021, ISSN: 0920-3796. DOI: `10.1016/j.fusengdes.2021.112611`.

[42] G. Sannazzaro *et al.*, “Structural load specification for ITER tokamak components”, in *2009 23rd IEEE/NPSS Symposium on Fusion Engineering*, Jun. 2009, pp. 1–4. DOI: `10.1109/FUSION.2009.5226521`.

[43] A. Portone *et al.*, “The Role of Advanced Modeling in the Assembly and Operation of Tokamak Magnets”, *IEEE Transactions on Applied Superconductivity*, vol. 34, no. 3, pp. 1–7, May 2024, ISSN: 1558-2515. DOI: `10.1109/TASC.2024.3356476`.

[44] D. Uglietti, “A review of commercial high temperature superconducting materials for large magnets: From wires and tapes to cables and conductors”, *Superconductor Science and Technology*, vol. 32, no. 5, p. 053 001, Apr. 2019, ISSN: 0953-2048. DOI: `10.1088/1361-6668/ab06a2`.

[45] W. Iliffe *et al.*, “In-situ measurements of the effect of radiation damage on the superconducting properties of coated conductors”, *Superconductor Science and Technology*, vol. 34, no. 9, 09LT01, Aug. 2021, ISSN: 0953-2048. DOI: `10.1088/1361-6668/ac1523`.

[46] D. Torsello, D. Gambino, L. Gozzelino, A. Trotta, and F. Laviano, “Expected radiation environment and damage for YBCO tapes in compact fusion reactors”, *Superconductor Science and Technology*, vol. 36, no. 1, p. 014 003, Dec. 2022, ISSN: 0953-2048. DOI: `10.1088/1361-6668/aca369`.

[47] C. Linsmeier *et al.*, “Development of advanced high heat flux and plasma-facing materials”, *Nuclear Fusion*, vol. 57, no. 9, p. 092 007, Jun. 2017, ISSN: 0029-5515. DOI: `10.1088/1741-4326/aa6f71`.

[48] D. Guo *et al.*, “Preliminary studies on the emulation of 14 MeV neutron irradiation in SiC with heavy ions”, *Fusion Engineering and Design*, vol. 100, pp. 274–279, Nov. 2015, ISSN: 0920-3796. DOI: `10.1016/j.fusengdes.2015.06.085`.

[49] A. F. Rowcliffe, L. M. Garrison, Y. Yamamoto, L. Tan, and Y. Katoh, “Materials challenges for the fusion nuclear science facility”, *Fusion Engineering and Design*, Special Issue: FESS-FNSF Study, vol. 135, pp. 290–301, Oct. 2018, ISSN: 0920-3796. DOI: `10.1016/j.fusengdes.2017.07.012`.

[50] D. Torsello *et al.*, "Triple ion beam irradiation of glass-ceramic materials for nuclear fusion technology", *Journal of Nuclear Materials*, vol. 567, p. 153 783, Aug. 2022, ISSN: 0022-3115. DOI: `10.1016/j.jnucmat.2022.153783`.

[51] M. Kovari, M. Coleman, I. Cristescu, and R. Smith, "Tritium resources available for fusion reactors", *Nuclear Fusion*, vol. 58, no. 2, p. 026 010, Dec. 2017, ISSN: 0029-5515. DOI: `10.1088/1741-4326/aa9d25`.

[52] R. J. Pearson, A. B. Antoniazzi, and W. J. Nuttall, "Tritium supply and use: A key issue for the development of nuclear fusion energy", *Fusion Engineering and Design*, Special Issue: Proceedings of the 13th International Symposium on Fusion Nuclear Technology (ISFNT-13), vol. 136, pp. 1140–1148, Nov. 2018, ISSN: 0920-3796. DOI: `10.1016/j.fusengdes.2018.04.090`.

[53] EUROfusion, *European Research Roadmap to the Realisation of Fusion Energy*, Nov. 2018. [Online]. Available: `https://www.euro-fusion.org/fileadmin/user_upload/EUROfusion/Documents/2018_Research_roadmap_long_version_01.pdf`.

[54] G. Federici, L. Boccaccini, F. Cismondi, M. Gasparotto, Y. Poitevin, and I. Ricapito, "An overview of the EU breeding blanket design strategy as an integral part of the DEMO design effort", *Fusion Engineering and Design*, vol. 141, pp. 30–42, Apr. 2019, ISSN: 0920-3796. DOI: `10.1016/j.fusengdes.2019.01.141`.

[55] B. N. Sorbom *et al.*, "ARC: A compact, high-field, fusion nuclear science facility and demonstration power plant with demountable magnets", *Fusion Engineering and Design*, vol. 100, pp. 378–405, Nov. 2015, ISSN: 0920-3796. DOI: `10.1016/j.fusengdes.2015.07.008`.

[56] E. Martelli *et al.*, "Advancements in DEMO WCLL breeding blanket design and integration", *International Journal of Energy Research*, vol. 42, no. 1, pp. 27–52, 2018, ISSN: 1099-114X. DOI: `10.1002/er.3750`.

[57] T. Eich *et al.*, "Scaling of the tokamak near the scrape-off layer H-mode power width and implications for ITER", vol. 53, no. 9, p. 093 031, Aug. 2013, ISSN: 0029-5515. DOI: `10.1088/0029-5515/53/9/093031`.

[58] R. A. Pitts *et al.*, "Physics basis for the first ITER tungsten divertor", *Nuclear Materials and Energy*, vol. 20, p. 100 696, Aug. 2019, ISSN: 2352-1791. DOI: `10.1016/j.nme.2019.100696`.

[59] R. P. Wenninger *et al.*, "DEMO divertor limitations during and in between ELMs", *Nuclear Fusion*, vol. 54, no. 11, p. 114 003, Nov. 2014, ISSN: 0029-5515. DOI: `10.1088/0029-5515/54/11/114003`.

[60] T. Loarer *et al.*, "Gas balance and fuel retention in fusion devices", *Nuclear Fusion*, vol. 47, no. 9, p. 1112, Aug. 2007, ISSN: 0029-5515. DOI: `10.1088/0029-5515/47/9/007`.

[61] M. Shimada *et al.*, "ITER research plan of plasma–wall interaction", *Journal of Nuclear Materials*, Proceedings of the 18th International Conference on Plasma-Surface Interactions in Controlled Fusion Device, vol. 390-391, pp. 282–285, Jun. 2009, ISSN: 0022-3115. DOI: `10.1016/j.jnucmat.2009.01.113`.

[62] T. Pütterich *et al.*, “Calculation and experimental test of the cooling factor of tungsten”, *Nuclear Fusion*, vol. 50, no. 2, p. 025 012, Jan. 2010, ISSN: 0029-5515. DOI: `10.1088/0029-5515/50/2/025012`.

[63] R. Dux, A. Loarte, C. Angioni, D. Coster, E. Fable, and A. Kallenbach, “The interplay of controlling the power exhaust and the tungsten content in ITER”, *Nuclear Materials and Energy*, Proceedings of the 22nd International Conference on Plasma Surface Interactions 2016, 22nd PSI, vol. 12, pp. 28–35, Aug. 2017, ISSN: 2352-1791. DOI: `10.1016/j.nme.2016.10.013`.

[64] T. Putterich, E. Fable, R. Dux, R. Neu, M. G. O’Mullane, and R. Wenninger, “Impurity Limits in a Reactor Grade Fusion Device”, in *42nd EPS Conference on Plasma Physics*, 2015.

[65] C. S. Pitcher and P. C. Stangeby, “Experimental divertor physics”, *Plasma Physics and Controlled Fusion*, vol. 39, no. 6, pp. 779–930, Jun. 1997, ISSN: 0741-3335. DOI: `10.1088/0741-3335/39/6/001`.

[66] Y. R. Martin, T. Takizuka, and ITPA CDBM H-mode Threshold Database Working Group, “Power requirement for accessing the H-mode in ITER”, *Journal of Physics: Conference Series*, vol. 123, no. 1, p. 012 033, Jul. 2008, ISSN: 1742-6596. DOI: `10.1088/1742-6596/123/1/012033`.

[67] M. Siccinio *et al.*, “DEMO physics challenges beyond ITER”, *Fusion Engineering and Design*, vol. 156, p. 111 603, Jul. 2020, ISSN: 0920-3796. DOI: `10.1016/j.fusengdes.2020.111603`.

[68] H. Zohm, F. Militello, T. W. Morgan, W. Morris, H. Reimerdes, and M. Siccinio, “The EU strategy for solving the DEMO exhaust problem”, *Fusion Engineering and Design*, vol. 166, p. 112 307, May 2021, ISSN: 0920-3796. DOI: `10.1016/j.fusengdes.2021.112307`.

[69] G. F. Harrer *et al.*, “Quasicontinuous Exhaust Scenario for a Fusion Reactor: The Renaissance of Small Edge Localized Modes”, *Physical Review Letters*, vol. 129, no. 16, p. 165 001, Oct. 2022. DOI: `10.1103/PhysRevLett.129.165001`.

[70] M. A. Makowski *et al.*, “Analysis of a multi-machine database on divertor heat fluxes”, *Physics of Plasmas*, vol. 19, no. 5, p. 056 122, May 2012, ISSN: 1070-664X. DOI: `10.1063/1.4710517`.

[71] A. Kallenbach, M. Bernert, R. Dux, F. Reimold, M. Wischmeier, and A. U. Team, “Analytical calculations for impurity seeded partially detached divertor conditions”, *Plasma Physics and Controlled Fusion*, vol. 58, no. 4, p. 045 013, Feb. 2016, ISSN: 0741-3335. DOI: `10.1088/0741-3335/58/4/045013`.

[72] M. L. Reinke, “Heat flux mitigation by impurity seeding in high-field tokamaks”, *Nuclear Fusion*, vol. 57, no. 3, p. 034 004, Jan. 2017, ISSN: 0029-5515. DOI: `10.1088/1741-4326/aa5145`.

[73] R. J. Goldston, M. L. Reinke, and J. A. Schwartz, “A new scaling for divertor detachment”, *Plasma Physics and Controlled Fusion*, vol. 59, no. 5, p. 055 015, Mar. 2017, ISSN: 0741-3335. DOI: `10.1088/1361-6587/aa5e6e`.

[74] S. S. Henderson *et al.*, “Parameter dependencies of the experimental nitrogen concentration required for detachment on ASDEX Upgrade and JET”, *Nuclear Materials and Energy*, vol. 28, p. 101 000, Sep. 2021, ISSN: 2352-1791. DOI: `10.1016/j.nme.2021.101000`.

[75] G. L. Derks *et al.*, “Benchmark of a self-consistent dynamic 1D divertor model DIV1D using the 2D SOLPS-ITER code”, *Plasma Physics and Controlled Fusion*, vol. 64, no. 12, p. 125 013, Nov. 2022, ISSN: 0741-3335. DOI: `10.1088/1361-6587/ac9dbd`.

[76] A. S. Kukushkin, H. D. Pacher, V. Kotov, G. W. Pacher, and D. Reiter, “Finalizing the ITER divertor design: The key role of SOLPS modeling”, *Fusion Engineering and Design*, vol. 86, no. 12, pp. 2865–2873, Dec. 2011, ISSN: 0920-3796. DOI: `10.1016/j.fusengdes.2011.06.009`.

[77] M. Wischmeier, “High density operation for reactor-relevant power exhaust”, *Journal of Nuclear Materials*, vol. 463, pp. 22–29, Aug. 2015, ISSN: 0022-3115. DOI: `10.1016/j.jnucmat.2014.12.078`.

[78] S. Wiesen *et al.*, “Plasma edge and plasma-wall interaction modelling: Lessons learned from metallic devices”, *Nuclear Materials and Energy*, Proceedings of the 22nd International Conference on Plasma Surface Interactions 2016, 22nd PSI, vol. 12, pp. 3–17, Aug. 2017, ISSN: 2352-1791. DOI: `10.1016/j.nme.2017.03.033`.

[79] E. Kaveeva *et al.*, “Speed-up of SOLPS-ITER code for tokamak edge modeling”, *Nuclear Fusion*, vol. 58, no. 12, p. 126 018, Oct. 2018, ISSN: 0029-5515. DOI: `10.1088/1741-4326/aae162`.

[80] S. Wiesen *et al.*, “The new SOLPS-ITER code package”, *Journal of Nuclear Materials*, vol. 463, pp. 480–484, Aug. 2015, ISSN: 0022-3115. DOI: `10.1016/j.jnucmat.2014.10.012`.

[81] E. Kaveeva *et al.*, “SOLPS-ITER modelling of ITER edge plasma with drifts and currents”, vol. 60, no. 4, p. 046 019, Mar. 2020, ISSN: 0029-5515. DOI: `10.1088/1741-4326/ab73c1`.

[82] D. Boeyaert, S. Carli, K. Ghoos, W. Dekeyser, S. Wiesen, and M. Baelmans, “Numerical error analysis of SOLPS-ITER simulations of EAST”, *Nuclear Fusion*, vol. 63, no. 1, p. 016 005, Nov. 2022, ISSN: 0029-5515. DOI: `10.1088/1741-4326/aca0ab`.

[83] C. M. Bishop, *Pattern recognition and machine learning* (Information science and statistics). New York: Springer, 2006, ISBN: 978-0-387-31073-2.

[84] R. Schneider *et al.*, “Plasma Edge Physics with B2-Eirene”, *Contributions to Plasma Physics*, vol. 46, no. 1-2, pp. 3–191, 2006, ISSN: 1521-3986. DOI: `https://doi.org/10.1002/ctpp.200610001`.

[85] K. Schmid, K. Krieger, A. Kukushkin, and A. Loarte, “DIVIMP modeling of tungsten impurity transport in ITER”, *Journal of Nuclear Materials*, Plasma-Surface Interactions-17, vol. 363-365, pp. 674–679, Jun. 2007, ISSN: 0022-3115. DOI: `10.1016/j.jnucmat.2007.01.045`.

[86] J. Romazanov *et al.*, “First ERO2.0 modeling of Be erosion and non-local transport in JET ITER-like wall”, *Physica Scripta*, vol. 2017, no. T170, p. 014 018, Sep. 2017, ISSN: 1402-4896. DOI: `10.1088/1402-4896/aa89ca`.

[87] A. Bhosekar and M. Ierapetritou, "Advances in surrogate based modeling, feasibility analysis, and optimization: A review", *Computers & Chemical Engineering*, vol. 108, pp. 250–267, Jan. 2018, ISSN: 0098-1354. DOI: `10.1016/j.compchemeng.2017.09.017`.

[88] G. Sun and S. Wang, "A review of the artificial neural network surrogate modeling in aerodynamic design", *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 233, no. 16, pp. 5863–5872, Dec. 2019, ISSN: 0954-4100. DOI: `10.1177/0954410019864485`.

[89] W. A. Pruett and R. L. Hester, "The Creation of Surrogate Models for Fast Estimation of Complex Model Outcomes", *PLOS ONE*, vol. 11, no. 6, e0156574, Jun. 2016, ISSN: 1932-6203. DOI: `10.1371/journal.pone.0156574`.

[90] Y. Zhang and L.-S. Young, "DNN-assisted statistical analysis of a model of local cortical circuits", *Scientific Reports*, vol. 10, no. 1, p. 20 139, Nov. 2020, ISSN: 2045-2322. DOI: `10.1038/s41598-020-76770-3`.

[91] K. L. van de Plassche *et al.*, "Fast modeling of turbulent transport in fusion plasmas using neural networks", *Physics of Plasmas*, vol. 27, no. 2, p. 022 310, Feb. 2020, ISSN: 1070-664X. DOI: `10.1063/1.5134126`.

[92] R. Preuss and U. v. Toussaint, "Gaussian Processes for SOLPS Data Emulation", *Fusion Science and Technology*, Mar. 2017. DOI: `10.13182/FST15-178`.

[93] V. Gopakumar and D. Samaddar, "Image mapping the temporal evolution of edge characteristics in tokamaks using neural networks", vol. 1, no. 1, p. 015 006, Feb. 2020, ISSN: 2632-2153. DOI: `10.1088/2632-2153/ab5639`.

[94] B. Zhu *et al.*, "Data-driven model for divertor plasma detachment prediction", *Journal of Plasma Physics*, vol. 88, no. 5, p. 895 880 504, Oct. 2022, ISSN: 0022-3778, 1469-7807. DOI: `10.1017/S002237782200085X`.

[95] M. Brenzke *et al.*, "Divertor power load predictions based on machine learning", *Nuclear Fusion*, vol. 61, no. 4, p. 046 023, Mar. 2021, ISSN: 0029-5515. DOI: `10.1088/1741-4326/abdb94`.

[96] T. A. Brody, *The Philosophy Behind Physics*, L. De La Peña and P. E. Hodgson, Eds. Berlin, Heidelberg: Springer, 1993, ISBN: 978-3-540-57952-6. DOI: `10.1007/978-3-642-78978-6`.

[97] N. Holtkamp, "An overview of the ITER project", *Fusion Engineering and Design*, Proceedings of the 24th Symposium on Fusion Technology, vol. 82, no. 5, pp. 427–434, Oct. 2007, ISSN: 0920-3796. DOI: `10.1016/j.fusengdes.2007.03.029`.

[98] B. Thacker, S. Doebling, F. Hemez, M. Anderson, J. Pepin, and E. Rodriguez, "Concepts of Model Verification and Validation", Los Alamos National Laboratory, Tech. Rep., Oct. 2004.

[99] S. I. Braginskii, "Transport Processes in a Plasma", *Reviews of Plasma Physics*, vol. 1, p. 205, Jan. 1965.

[100] V. M. Zhdanov, "Transport Processes in Multicomponent Plasma", *Plasma Physics and Controlled Fusion*, vol. 44, no. 10, p. 2283, Oct. 2002, ISSN: 0741-3335. DOI: `10.1088/0741-3335/44/10/701`.

[101] K.-U. Riemann, "The Bohm criterion and sheath formation", *Journal of Physics D: Applied Physics*, vol. 24, no. 4, p. 493, Apr. 1991, ISSN: 0022-3727. DOI: `10.1088/0022-3727/24/4/001`.

[102] R. Chodura, "Plasma Flow in the Sheath and the Presheath of a Scrape-Off Layer", in *Physics of Plasma-Wall Interactions in Controlled Fusion*, D. E. Post and R. Behrisch, Eds., Boston, MA: Springer US, 1986, pp. 99–134, ISBN: 978-1-4757-0067-1. DOI: `10.1007/978-1-4757-0067-1_4`.

[103] P. C. Stangeby, "Basic physical processes and reduced models for plasma detachment", *Plasma Physics and Controlled Fusion*, vol. 60, no. 4, p. 044 022, Mar. 2018, ISSN: 0741-3335. DOI: `10.1088/1361-6587/aaacf6`.

[104] C. Silva, H. Figueiredo, P. Duarte, and H. Fernandes, "Characterization of the poloidal asymmetries in the ISTTOK edge plasma", *Plasma Physics and Controlled Fusion*, vol. 53, no. 8, p. 085 021, Jun. 2011, ISSN: 0741-3335. DOI: `10.1088/0741-3335/53/8/085021`.

[105] A. Loarte *et al.*, "Plasma detachment in JET Mark I divertor experiments", *Nuclear Fusion*, vol. 38, no. 3, p. 331, Mar. 1998, ISSN: 0029-5515. DOI: `10.1088/0029-5515/38/3/303`.

[106] S. I. Krasheninnikov and A. S. Kukushkin, "Physics of ultimate detachment of a tokamak divertor plasma", *Journal of Plasma Physics*, vol. 83, no. 5, p. 155 830 501, Oct. 2017, ISSN: 0022-3778, 1469-7807. DOI: `10.1017/S0022377817000654`.

[107] A. Kallenbach *et al.*, "Partial detachment of high power discharges in ASDEX Upgrade", *Nuclear Fusion*, vol. 55, no. 5, p. 053 026, Apr. 2015, ISSN: 0029-5515. DOI: `10.1088/0029-5515/55/5/053026`.

[108] J. Horacek *et al.*, "Multi-machine scaling of the main SOL parallel heat flux width in tokamak limiter plasmas", *Plasma Physics and Controlled Fusion*, vol. 58, no. 7, p. 074 005, May 2016, ISSN: 0741-3335. DOI: `10.1088/0741-3335/58/7/074005`.

[109] A. Kallenbach *et al.*, "Impurity seeding for tokamak power exhaust: From present devices via ITER to DEMO", *Plasma Physics and Controlled Fusion*, vol. 55, no. 12, p. 124 041, Nov. 2013, ISSN: 0741-3335. DOI: `10.1088/0741-3335/55/12/124041`.

[110] D. Neuwirth, V. Rohde, T. Schwarz-Selinger, and A. U. Team, "Formation of ammonia during nitrogen-seeded discharges at ASDEX Upgrade", *Plasma Physics and Controlled Fusion*, vol. 54, no. 8, p. 085 008, Jun. 2012, ISSN: 0741-3335. DOI: `10.1088/0741-3335/54/8/085008`.

[111] V. Rohde and M. Oberkofler, "Ammonia production in nitrogen seeded plasma discharges in ASDEX Upgrade", *Journal of Nuclear Materials*, vol. 463, pp. 672–675, Aug. 2015, ISSN: 0022-3115. DOI: `10.1016/j.jnucmat.2015.01.010`.

[112] A. Drenik *et al.*, "Evolution of nitrogen concentration and ammonia production in N2-seeded H-mode discharges at ASDEX Upgrade", *Nuclear Fusion*, vol. 59, no. 4, p. 046 010, Feb. 2019, ISSN: 0029-5515. DOI: `10.1088/1741-4326/aafe23`.

[113] A. Eksaeva *et al.*, “Predictive 3D modelling of erosion and deposition in ITER with ERO2.0: From beryllium main wall, tungsten divertor to full-tungsten device”, *Physica Scripta*, vol. 97, no. 1, p. 014 001, Jan. 2022, ISSN: 1402-4896. DOI: `10.1088/1402-4896/ac454f`.

[114] V. Rozhansky *et al.*, “Multi-machine SOLPS-ITER comparison of impurity seeded H-mode radiative divertor regimes with metal walls”, *Nuclear Fusion*, vol. 61, no. 12, p. 126 073, Dec. 2021, ISSN: 0029-5515. DOI: `10.1088/1741-4326/ac3699`.

[115] H. D. Pacher *et al.*, “Operating window for high divertor radiation in ITER”, *Journal of Nuclear Materials*, vol. 266-269, pp. 1172–1179, Mar. 1999, ISSN: 0022-3115. DOI: `10.1016/S0022-3115(98)00567-4`.

[116] A. Loarte, “Multi-machine modelling of divertor geometry effects”, *Journal of Nuclear Materials*, 14th Int. Conf. on Plasma-Surface Interactions in Controlled Fusion Devices, vol. 290-293, pp. 805–811, Mar. 2001, ISSN: 0022-3115. DOI: `10.1016/S0022-3115(00)00456-6`.

[117] A. Loarte, “Effects of divertor geometry on tokamak plasmas”, *Plasma Physics and Controlled Fusion*, vol. 43, no. 6, R183, Jun. 2001, ISSN: 0741-3335. DOI: `10.1088/0741-3335/43/6/201`.

[118] M. Groth *et al.*, “Divertor plasma conditions and neutral dynamics in horizontal and vertical divertor configurations in JET-ILW low confinement mode plasmas”, *Journal of Nuclear Materials*, PLASMA-SURFACE INTERACTIONS 21, vol. 463, pp. 471–476, Aug. 2015, ISSN: 0022-3115. DOI: `10.1016/j.jnucmat.2014.12.030`.

[119] C. Sang, P. C. Stangeby, H. Y. Guo, and D. Wang, “The role of divertor pumping in plasma detachment and particle exhaust in a closed divertor”, vol. 61, no. 1, p. 016 022, Dec. 2020, ISSN: 0029-5515. DOI: `10.1088/1741-4326/abc356`.

[120] H. Q. Wang *et al.*, “Dependence of particle and power dissipation on divertor geometry and plasma shaping in DIII-D small-angle-slot divertor”, *Nuclear Materials and Energy*, vol. 33, p. 101 301, Oct. 2022, ISSN: 2352-1791. DOI: `10.1016/j.nme.2022.101301`.

[121] C. Meineri, P. Muscente, C. Theiler, and D. Galassi, “Numerical study of fully baffled Super-X L-mode discharges on TCV”, *Nuclear Materials and Energy*, vol. 34, p. 101 383, Mar. 2023, ISSN: 2352-1791. DOI: `10.1016/j.nme.2023.101383`.

[122] R. Kembleton, M. Siccinio, F. Maviglia, and F. Militello, “Benefits and Challenges of Advanced Divertor Configurations in DEMO”, *Fusion Engineering and Design*, vol. 179, p. 113 120, Jun. 2022, ISSN: 0920-3796. DOI: `10.1016/j.fusengdes.2022.113120`.

[123] V. A. Soukhanovskii, “A review of radiative detachment studies in tokamak advanced magnetic divertor configurations”, *Plasma Physics and Controlled Fusion*, vol. 59, no. 6, p. 064 005, Apr. 2017, ISSN: 0741-3335. DOI: `10.1088/1361-6587/aa6959`.

[124] V. A. Soukhanovskii, G. Cunningham, J. R. Harrison, F. Federici, and P. Ryan, “First snowflake divertor experiments in MAST-U tokamak”, *Nuclear Materials and Energy*, vol. 33, p. 101 278, Oct. 2022, ISSN: 2352-1791. DOI: `10.1016/j.nme.2022.101278`.

[125] A. S. Kukushkin *et al.*, “Analysis of performance of the optimized divertor in ITER”, vol. 49, no. 7, p. 075 008, May 2009, ISSN: 0029-5515. DOI: `10.1088/0029-5515/49/7/075008`.

[126] F. Subba, D. P. Coster, M. Moscheni, and M. Siccinio, “SOLPS-ITER modeling of divertor scenarios for EU-DEMO”, vol. 61, no. 10, p. 106 013, Sep. 2021, ISSN: 0029-5515. DOI: `10.1088/1741-4326/ac1c85`.

[127] L. Aho-Mantila *et al.*, “Scoping the characteristics and benefits of a connected double-null configuration for power exhaust in EU-DEMO”, *Nuclear Materials and Energy*, vol. 26, p. 100 886, Mar. 2021, ISSN: 2352-1791. DOI: `10.1016/j.nme.2020.100886`.

[128] J. Romazanov *et al.*, “First Monte-Carlo modelling of global beryllium migration in ITER using ERO2.0”, *Contributions to Plasma Physics*, vol. 60, no. 5-6, e201900149, 2020, ISSN: 1521-3986. DOI: `10.1002/ctpp.201900149`.

[129] A. Kirschner *et al.*, “Erosion estimates for the divertor and main wall components from STEP”, *Nuclear Fusion*, vol. 63, no. 12, p. 126 055, Nov. 2023, ISSN: 0029-5515. DOI: `10.1088/1741-4326/ad067d`.

[130] P. C. Stangeby and C. Sang, “Strong correlation between D2 density and electron temperature at the target of divertors found in SOLPS analysis”, *Nuclear Fusion*, vol. 57, no. 5, p. 056 007, Mar. 2017, ISSN: 0029-5515. DOI: `10.1088/1741-4326/aa5e27`.

[131] D. Moulton, P. C. Stangeby, X. Bonnin, and R. A. Pitts, “Comparison between SOLPS-4.3 and the Lengyel Model for ITER baseline neon-seeded plasmas”, vol. 61, no. 4, p. 046 029, Mar. 2021, ISSN: 0029-5515. DOI: `10.1088/1741-4326/abe4b2`.

[132] A. R. Polevoi, A. Loarte, A. S. Kukushkin, H. D. Pacher, G. W. Pacher, and F. Köchl, “Analysis of fuelling requirements in ITER H-modes with SOLPS-EPED1 derived scalings”, *Nuclear Fusion*, vol. 57, no. 2, p. 022 014, Sep. 2016, ISSN: 0029-5515. DOI: `10.1088/0029-5515/57/2/022014`.

[133] M. Wensing *et al.*, “SOLPS-ITER validation with TCV L-mode discharges”, *Physics of Plasmas*, vol. 28, no. 8, p. 082 508, Aug. 2021, ISSN: 1070-664X. DOI: `10.1063/5.0056216`.

[134] T. D. Rognlien, J. L. Milovich, M. E. Rensink, and G. D. Porter, “A fully implicit, time dependent 2-D fluid code for modeling tokamak edge plasmas”, *Journal of Nuclear Materials*, Plasma-Surface Interactions in Controlled Fusion Devices, vol. 196-198, pp. 347–351, Dec. 1992, ISSN: 0022-3115. DOI: `10.1016/S0022-3115(06)80058-9`.

[135] H. Bufferand *et al.*, “Implementation of drift velocities and currents in SOLEDGE2D-EIRENE”, *Nuclear Materials and Energy*, Proceedings of the 22nd International Conference on Plasma Surface Interactions 2016, 22nd PSI, vol. 12, pp. 852–857, Aug. 2017, ISSN: 2352-1791. DOI: `10.1016/j.nme.2016.11.031`.

[136] S. Wiesen, *EDGE2D/EIRENE code interface report*, Jun. 2006.

[137] Y. Feng, F. Sardei, J. Kisslinger, P. Grigull, K. McCormick, and D. Reiter, "3D Edge Modeling and Island Divertor Physics", *Contributions to Plasma Physics*, vol. 44, no. 1-3, pp. 57–69, 2004, ISSN: 1521-3986. DOI: `10.1002/ctpp.200410009`.

[138] B. J. Braams, "Computational studies in tokamak equilibrium and transport", Ph.D. dissertation, Rijksuniversiteit Utrecht, Netherlands, Jun. 1986.

[139] V. A. Rozhansky, S. P. Voskoboynikov, E. G. Kaveeva, D. P. Coster, and R. Schneider, "Simulation of tokamak edge plasma including self-consistent electric fields", *Nuclear Fusion*, vol. 41, no. 4, pp. 387–401, Apr. 2001, ISSN: 0029-5515. DOI: `10.1088/0029-5515/41/4/305`.

[140] D. Reiter, M. Baelmans, and P. Börner, "The EIRENE and B2-EIRENE Codes", *Fusion Science and Technology*, Apr. 2005. DOI: `10.13182/FST47-172`.

[141] D. P. Coster, "Exploring the edge operating space of fusion reactors using reduced physics models", *Nuclear Materials and Energy*, Proceedings of the 22nd International Conference on Plasma Surface Interactions 2016, 22nd PSI, vol. 12, pp. 1055–1060, Aug. 2017, ISSN: 2352-1791. DOI: `10.1016/j.nme.2016.12.033`.

[142] D. P. Coster, X. Bonnin, B. Braams, D. Reiter, R. Schneider, and the ASDEX Upgrade Team, "Simulation of the Edge Plasma in Tokamaks", *Physica Scripta*, vol. 2004, no. T108, p. 7, Jan. 2004, ISSN: 1402-4896. DOI: `10.1238/Physica.Topical.108a00007`.

[143] N. Horsten, G. Samaey, and M. Baelmans, "Development and assessment of 2D fluid neutral models that include atomic databases and a microscopic reflection model", vol. 57, no. 11, p. 116 043, Aug. 2017, ISSN: 0029-5515. DOI: `10.1088/1741-4326/aa8009`.

[144] M. Blommaert, W. Dekeyser, N. Horsten, P. Börner, and M. Baelmans, "Implementation of a consistent fluid-neutral model in SOLPS-ITER and benchmark with EIRENE", *Contributions to Plasma Physics*, vol. 58, no. 6-8, pp. 718–724, 2018, ISSN: 1521-3986. DOI: `10.1002/ctpp.201700175`.

[145] W. Van Uytven, M. Blommaert, W. Dekeyser, N. Horsten, and M. Baelmans, "Implementation of a separate fluid-neutral energy equation in SOLPS-ITER and its impact on the validity range of advanced fluid-neutral models", *Contributions to Plasma Physics*, vol. 60, no. 5-6, e201900147, 2020, ISSN: 1521-3986. DOI: `10.1002/ctpp.201900147`.

[146] N. Horsten *et al.*, "Validation of SOLPS-ITER simulations with kinetic, fluid, and hybrid neutral models for JET-ILW low-confinement mode plasmas", *Nuclear Materials and Energy*, vol. 33, p. 101 247, Oct. 2022, ISSN: 2352-1791. DOI: `10.1016/j.nme.2022.101247`.

[147] W. V. Uytven, W. Dekeyser, M. Blommaert, S. Carli, and M. Baelmans, "Assessment of advanced fluid neutral models for the neutral atoms in the plasma edge and application in ITER geometry", *Nuclear Fusion*, vol. 62, no. 8, p. 086 023, Jun. 2022, ISSN: 0029-5515. DOI: `10.1088/1741-4326/ac72b4`.

[148] K. Verhaegh *et al.*, "The role of plasma-molecule interactions on power and particle balance during detachment on the TCV tokamak", *Nuclear Fusion*, vol. 61, no. 10, p. 106 014, Sep. 2021, ISSN: 0029-5515. DOI: `10.1088/1741-4326/ac1dc5`.

[149] K. Verhaegh *et al.*, "Investigating the impact of the molecular charge-exchange rate on detached SOLPS-ITER simulations", *Nuclear Fusion*, vol. 63, no. 7, p. 076 015, May 2023, ISSN: 0029-5515. DOI: `10.1088/1741-4326/acd394`.

[150] D. Tskhakaya *et al.*, "On Kinetic Effects during Parallel Transport in the SOL", *Contributions to Plasma Physics*, vol. 48, no. 1-3, pp. 89–93, 2008, ISSN: 1521-3986. DOI: `10.1002/ctpp.200810015`.

[151] S. Patankar, *Numerical Heat Transfer and Fluid Flow*. Taylor & Francis, Oct. 2018, ISBN: 978-1-4822-3421-3.

[152] J. H. Ferziger, M. Perić, and R. L. Street, *Computational Methods for Fluid Dynamics*. Cham: Springer International Publishing, 2020, ISBN: 978-3-319-99693-6. DOI: `10.1007/978-3-319-99693-6`.

[153] J. D. Lore *et al.*, "Time-dependent SOLPS-ITER simulations of the tokamak plasma boundary for model predictive control using SINDy*", *Nuclear Fusion*, vol. 63, no. 4, p. 046 015, Mar. 2023, ISSN: 0029-5515. DOI: `10.1088/1741-4326/acbe0e`.

[154] J. Lore *et al.*, "Time-Dependent Boundary Modeling to Inform Design of SPARC Diagnostic and Actuators", Tech. Rep. ORNL/TM–2023/3063, NFE-21-108605, 2000338, Sep. 2023. DOI: `10.2172/2000338`.

[155] A. E. Jaervinen *et al.*, "ExB Flux Driven Detachment Bifurcation in the DIII-D Tokamak", *Physical Review Letters*, vol. 121, no. 7, p. 075 001, Aug. 2018. DOI: `10.1103/PhysRevLett.121.075001`.

[156] W. Dekeyser *et al.*, "Plasma edge simulations including realistic wall geometry with SOLPS-ITER", *Nuclear Materials and Energy*, vol. 27, p. 100 999, Jun. 2021, ISSN: 2352-1791. DOI: `10.1016/j.nme.2021.100999`.

[157] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems*, Second edition. Beijing [China] ; Sebastopol, CA: O'Reilly Media, Inc, 2019, ISBN: 978-1-4920-3264-9.

[158] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning* (Adaptive computation and machine learning). Cambridge, Massachusetts: The MIT Press, 2016, ISBN: 978-0-262-03561-3.

[159] G. James, D. Witten, T. Hastie, R. Tibshirani, and J. E. Taylor, *An introduction to statistical learning: with applications in Python* (Springer texts in statistics). Cham, Switzerland: Springer, 2023, ISBN: 978-3-031-39189-7.

[160] J. Huang, G. Galal, M. Etemadi, and M. Vaidyanathan, "Evaluation and Mitigation of Racial Bias in Clinical Machine Learning Models: Scoping Review", *JMIR Medical Informatics*, vol. 10, no. 5, e36388, May 2022. DOI: `10.2196/36388`.

[161] J. Willard, X. Jia, S. Xu, M. Steinbach, and V. Kumar, “Integrating Scientific Knowledge with Machine Learning for Engineering and Environmental Systems”, *ACM Computing Surveys*, vol. 55, no. 4, 66:1–66:37, Nov. 2022, ISSN: 0360-0300. DOI: `10.1145/3514228`.

[162] S. P. Adam, S.-A. N. Alexandropoulos, P. M. Pardalos, and M. N. Vrahatis, “No Free Lunch Theorem: A Review”, in *Approximation and Optimization : Algorithms, Complexity and Applications*, I. C. Demetriou and P. M. Pardalos, Eds., Cham: Springer International Publishing, 2019, pp. 57–82, ISBN: 978-3-030-12767-1. DOI: `10.1007/978-3-030-12767-1_5`.

[163] J. Degrave *et al.*, “Magnetic control of tokamak plasmas through deep reinforcement learning”, *Nature*, vol. 602, no. 7897, pp. 414–419, Feb. 2022, ISSN: 1476-4687. DOI: `10.1038/s41586-021-04301-9`.

[164] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization”, *arXiv:1412.6980 [cs]*, Jan. 2017.

[165] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, “Self-Normalizing Neural Networks”, in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017.

[166] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”, in *Proceedings of the 32nd International Conference on Machine Learning*, PMLR, Jun. 2015, pp. 448–456.

[167] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”, *The journal of machine learning research*, vol. 15, pp. 1929–1958, Jun. 2014.

[168] A. Paszke *et al.*, “PyTorch: An Imperative Style, High-Performance Deep Learning Library”, *arXiv:1912.01703 [cs-LG]*, Dec. 2019. DOI: `10.48550/arXiv.1912.01703`.

[169] M. Abadi *et al.*, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*, Nov. 2015. [Online]. Available: `https://www.tensorflow.org/`.

[170] Y. Freund and R. E. Schapire, “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting”, *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, Aug. 1997, ISSN: 0022-0000. DOI: `10.1006/jcss.1997.1504`.

[171] J. H. Friedman, “Greedy function approximation: A gradient boosting machine.”, *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001, ISSN: 0090-5364, 2168-8966. DOI: `10.1214/aos/1013203451`.

[172] L. Iosipoi and A. Vakhrushev, “SketchBoost: Fast Gradient Boosted Decision Tree for Multioutput Problems”, *Advances in Neural Information Processing Systems*, vol. 35, pp. 25 422–25 435, Dec. 2022.

[173] J. W. Haverkort, *Axisymmetric Ideal MHD Tokamak Equilibria*, May 2009. [Online]. Available: `https://jwhaverkort.weblog.tudelft.nl/files/2024/03/Equilibria.pdf`.

[174] S. Wiesen *et al.*, “Modelling radiative power exhaust towards future fusion devices”, in *46th EPS Conference on Plasma Physics*, 2019.

[175] L. L. Lao, H. S. John, R. D. Stambaugh, A. G. Kellman, and W. Pfeiffer, “Reconstruction of current profile parameters and plasma shapes in tokamaks”, *Nuclear Fusion*, vol. 25, no. 11, p. 1611, Nov. 1985, ISSN: 0029-5515. DOI: `10.1088/0029-5515/25/11/007`.

[176] E. Sytova *et al.*, “Comparing N versus Ne as divertor radiators in ASDEX-upgrade and ITER”, *Nuclear Materials and Energy*, vol. 19, pp. 72–78, May 2019, ISSN: 2352-1791. DOI: `10.1016/j.nme.2019.02.019`.

[177] A. Smolders *et al.*, “Comparison of high density and nitrogen seeded detachment using SOLPS-ITER simulations of the tokamak á configuration variable”, *Plasma Physics and Controlled Fusion*, vol. 62, no. 12, p. 125 006, Dec. 2020, ISSN: 0741-3335, 1361-6587. DOI: `10.1088/1361-6587/abbcc5`.

[178] E. Kaveeva *et al.*, “SOLPS-ITER drift modelling of JET Ne and N-seeded H-modes”, *Nuclear Materials and Energy*, vol. 28, p. 101 030, Sep. 2021, ISSN: 2352-1791. DOI: `10.1016/j.nme.2021.101030`.

[179] I. Y. Senichenkov *et al.*, “SOLPS-ITER modeling of Ar and N seeded discharges in ASDEX upgrade and ITER”, *Nuclear Materials and Energy*, vol. 34, p. 101 361, Mar. 2023, ISSN: 2352-1791. DOI: `10.1016/j.nme.2022.101361`.

[180] D. P. Coster, “Reduced Physics Models in SOLPS for Reactor Scoping Studies”, *Contributions to Plasma Physics*, vol. 56, no. 6-8, pp. 790–795, 2016, ISSN: 1521-3986. DOI: `https://doi.org/10.1002/ctpp.201610035`.

[181] D. Reiter, “Progress in two-dimensional plasma edge modelling”, *Journal of Nuclear Materials*, Plasma-Surface Interactions in Controlled Fusion Devices, vol. 196-198, pp. 80–89, Dec. 1992, ISSN: 0022-3115. DOI: `10.1016/S0022-3115(06)80014-0`.

[182] L. Aho-Mantila *et al.*, “Predictions of radiation pattern and in–out asymmetries in the DEMO scrape-off layer using fluid neutrals”, *Nuclear Fusion*, vol. 62, no. 5, p. 056 015, Mar. 2022, ISSN: 0029-5515. DOI: `10.1088/1741-4326/ac4d62`.

[183] F. Subba *et al.*, “Modelling of mitigation of the power divertor loading for the EU DEMO through Ar injection”, *Plasma Physics and Controlled Fusion*, vol. 60, no. 3, p. 035 013, Feb. 2018, ISSN: 0741-3335. DOI: `10.1088/1361-6587/aaa508`.

[184] W. Dekeyser, X. Bonnin, S. W. Lisgo, R. A. Pitts, and B. LaBombard, “Implementation of a 9-point stencil in SOLPS-ITER and implications for Alcator C-Mod divertor plasma simulations”, *Nuclear Materials and Energy*, vol. 18, pp. 125–130, Jan. 2019, ISSN: 2352-1791. DOI: `10.1016/j.nme.2018.12.016`.

[185] D. P. Coster, “Detachment physics in SOLPS simulations”, *Journal of Nuclear Materials*, Proceedings of the 19th International Conference on Plasma-Surface Interactions in Controlled Fusion, vol. 415, no. 1, Supplement, S545–S548, Aug. 2011, ISSN: 0022-3115. DOI: `10.1016/j.jnucmat.2010.12.223`.

[186] V. Kotov and D. Reiter, "B2-B2.5 Code Benchmarking, Part III: Convergence issues of the B2-EIRENE code", Forschungszentrum Jülich GmbH, Tech. Rep. 4371.

[187] H. D. Pacher *et al.*, "Impurity seeding and scaling of edge parameters in ITER", *Journal of Nuclear Materials*, Proceedings of the 18th International Conference on Plasma-Surface Interactions in Controlled Fusion Device, vol. 390-391, pp. 259–262, Jun. 2009, ISSN: 0022-3115. DOI: `10.1016/j.jnucmat.2009.01.089`.

[188] T. Pütterich, E. Fable, R. Dux, M. O'Mullane, R. Neu, and M. Siccinio, "Determination of the tolerable impurity concentrations in a fusion reactor using a consistent set of cooling factors", *Nuclear Fusion*, vol. 59, no. 5, p. 056 013, Mar. 2019, ISSN: 0029-5515. DOI: `10.1088/1741-4326/ab0384`.

[189] I. M. Sobol', "On the distribution of points in a cube and the approximate evaluation of integrals", *USSR Computational Mathematics and Mathematical Physics*, vol. 7, no. 4, pp. 86–112, Jan. 1967, ISSN: 0041-5553. DOI: `10.1016/0041-5553(67)90144-9`.

[190] S. E. Davis, S. Cremaschi, and M. R. Eden, "Efficient Surrogate Model Development: Impact of Sample Size and Underlying Model Dimensions", in *Computer Aided Chemical Engineering*, ser. 13 International Symposium on Process Systems Engineering (PSE 2018), M. R. Eden, M. G. Ierapetritou, and G. P. Towler, Eds., vol. 44, Elsevier, Jan. 2018, pp. 979–984. DOI: `10.1016/B978-0-444-64241-7.50158-0`.

[191] D. Coster, "Characterization of oscillations observed in reduced physics SOLPS simulations", *Contributions to Plasma Physics*, vol. 58, no. 6-8, pp. 666–674, 2018, ISSN: 1521-3986. DOI: `https://doi.org/10.1002/ctpp.201700159`.

[192] A. S. Kukushkin and S. I. Krasheninnikov, "Bifurcations and oscillations in divertor plasma", *Plasma Physics and Controlled Fusion*, vol. 61, no. 7, p. 074 001, May 2019, ISSN: 0741-3335. DOI: `10.1088/1361-6587/ab1bba`.

[193] R. D. Smirnov, A. S. Kukushkin, S. I. Krasheninnikov, A. Y. Pigarov, and T. D. Rognlien, "Impurity-induced divertor plasma oscillations", *Physics of Plasmas*, vol. 23, no. 1, p. 012 503, Jan. 2016, ISSN: 1070-664X. DOI: `10.1063/1.4939539`.

[194] A. Loarte *et al.*, "Self-Sustained Divertor Plasma Oscillations in the JET Tokamak", *Physical Review Letters*, vol. 83, no. 18, pp. 3657–3660, Nov. 1999. DOI: `10.1103/PhysRevLett.83.3657`.

[195] S. I. Krasheninnikov, A. S. Kukushkin, V. I. Pistunovich, and V. A. Pozharov, "Self-sustained oscillations in the divertor plasma", *Nuclear Fusion*, vol. 27, no. 11, pp. 1805–1816, Nov. 1987, ISSN: 0029-5515. DOI: `10.1088/0029-5515/27/11/006`.

[196] F. Reimold *et al.*, "Divertor studies in nitrogen induced completely detached H-modes in full tungsten ASDEX Upgrade", *Nuclear Fusion*, vol. 55, no. 3, p. 033 004, Feb. 2015, ISSN: 0029-5515. DOI: `10.1088/0029-5515/55/3/033004`.

[197] U. Stroth *et al.*, “Model for access and stability of the X-point radiator and the threshold for marfes in tokamak plasmas”, *Nuclear Fusion*, vol. 62, no. 7, p. 076 008, Apr. 2022, ISSN: 0029-5515. DOI: `10.1088/1741-4326/ac613a`.

[198] O. Pan *et al.*, “SOLPS-ITER simulations of an X-point radiator in the ASDEX Upgrade tokamak”, *Nuclear Fusion*, vol. 63, no. 1, p. 016 001, Nov. 2022, ISSN: 0029-5515. DOI: `10.1088/1741-4326/ac9742`.

[199] S. S. Henderson *et al.*, “Divertor detachment and reattachment with mixed impurity seeding on ASDEX Upgrade”, *Nuclear Fusion*, vol. 63, no. 8, p. 086 024, Jul. 2023, ISSN: 0029-5515. DOI: `10.1088/1741-4326/ace2d6`.

[200] H. D. Pacher, A. S. Kukushkin, G. W. Pacher, V. Kotov, R. A. Pitts, and D. Reiter, “Impurity seeding in ITER DT plasmas in a carbon-free environment”, *Journal of Nuclear Materials*, vol. 463, pp. 591–595, Aug. 2015, ISSN: 0022-3115. DOI: `10.1016/j.jnucmat.2014.11.104`.

[201] K. Hoshino *et al.*, “Benchmarking Kinetic and Fluid Neutral Models with Drift Effects”, *Contributions to Plasma Physics*, vol. 48, no. 1-3, pp. 136–140, 2008, ISSN: 1521-3986. DOI: `10.1002/ctpp.200810024`.

[202] W. Dekeyser *et al.*, “SOLPS-ITER Modeling of the Alcator C-Mod Divertor Plasma”, *Plasma and Fusion Research*, vol. 11, pp. 1 403 103–1 403 103, 2016. DOI: `10.1585/pfr.11.1403103`.

[203] S. Wang, S. Sankaran, H. Wang, and P. Perdikaris, “An Expert’s Guide to Training Physics-informed Neural Networks”, *arXiv:2308.08468 [cs.LG]*, Aug. 2023. DOI: `10.48550/arXiv.2308.08468`.

[204] A. V. Chankin *et al.*, “Comparison between measured divertor parameters in ASDEX Upgrade and SOLPS code solutions”, *Journal of Nuclear Materials*, Plasma-Surface Interactions-17, vol. 363-365, pp. 335–340, Jun. 2007, ISSN: 0022-3115. DOI: `10.1016/j.jnucmat.2007.01.088`.

[205] Y. P. Chen *et al.*, “SOLPS-ITER modeling of SOL-divertor plasmas with different configurations in EAST”, *Nuclear Fusion*, vol. 60, no. 3, p. 036 019, Feb. 2020, ISSN: 0029-5515. DOI: `10.1088/1741-4326/ab69e3`.

[206] A. Perek *et al.*, “A spectroscopic inference and SOLPS-ITER comparison of flux-resolved edge plasma parameters in detachment experiments on TCV”, *Nuclear Fusion*, vol. 62, no. 9, p. 096 012, Jul. 2022, ISSN: 0029-5515. DOI: `10.1088/1741-4326/ac7813`.

[207] A. Loarte *et al.*, “Initial evaluations in support of the new ITER baseline and Research Plan”, Tech. Rep. ITR-24-004, Mar. 2024.

[208] S. Wiesen *et al.*, “Control of particle and power exhaust in pellet fuelled ITER DT scenarios employing integrated models”, *Nuclear Fusion*, vol. 57, no. 7, p. 076 020, May 2017, ISSN: 0029-5515. DOI: `10.1088/1741-4326/aa6ecc`.

[209] R. Preuss and U. Von Toussaint, “Global Optimization Employing Gaussian Process-Based Bayesian Surrogates”, *Entropy*, vol. 20, no. 3, p. 201, Mar. 2018. DOI: `10.3390/e20030201`.

[210] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, “Scientific Machine Learning Through Physics–Informed Neural Networks: Where we are and What’s Next”, *Journal of Scientific Computing*, vol. 92, no. 3, p. 88, Jul. 2022, ISSN: 1573-7691. DOI: `10.1007/s10915-022-01939-z`.

[211] A. S. Krishnapriyan, A. Gholami, S. Zhe, R. M. Kirby, and M. W. Mahoney, “Characterizing possible failure modes in physics-informed neural networks”, *arXiv:2109.01050 [cs.LG]*, Nov. 2021. DOI: `10.48550/arXiv.2109.01050`.

[212] P. Thörnig, “JURECA: Data Centric and Booster Modules implementing the Modular Supercomputing Architecture at Jülich Supercomputing Centre”, *Journal of large-scale research facilities JLSRF*, vol. 7, no. 0, p. 182, Oct. 2021, ISSN: 2364-091X. DOI: `10.17815/jlsrf-7-182`.

[213] C. R. Harris *et al.*, “Array programming with NumPy”, *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020, ISSN: 1476-4687. DOI: `10.1038/s41586-020-2649-2`.

[214] P. Virtanen *et al.*, “SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python”, *arXiv:1907.10121 [physics]*, Jul. 2019.

[215] W. McKinney, “Data Structures for Statistical Computing in Python”, in *Python in Science Conference*, Austin, Texas, 2010, pp. 56–61. DOI: `10.25080/Majora-92bf1922-00a`.

[216] T. pandas development team, *Pandas-dev/pandas: Pandas 1.1.3*, Oct. 2020. DOI: `10.5281/ZENODO.4067057`. [Online]. Available: `https://zenodo.org/record/4067057`.

[217] J. D. Hunter, “Matplotlib: A 2D Graphics Environment”, *Computing in Science Engineering*, vol. 9, no. 3, pp. 90–95, May 2007, ISSN: 1558-366X. DOI: `10.1109/MCSE.2007.55`.

[218] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python”, *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011, ISSN: 1533-7928.

# Figure attributions

Figure 1.1B: Reproduced from [19], Copyright © 2023 Meschini, Laviano, Ledda, Pettinari, Testoni, Torsello and Panella. Creative Commons Attribution License `https://creativecommons.org/licenses/by/4.0`. The letter for numeration of sub-figures was added for this thesis.

Figure 1.2A: Reproduced from [25], Copyright © 2014 S. Li et al. Creative Commons Attribution License `https://creativecommons.org/licenses/by/4.0`. The letter for numeration of sub-figures was added for this thesis.

Figure 1.2B: Reproduced from Wikimedia Commons, Copyright © Max-Planck Institut für Plasmaphysik. Creative Commons Attribution License 3.0 `https://creativecommons.org/licenses/by/3.0`. The letter for numeration of sub-figures was added for this thesis.