Future Generation Computer Systems 159 (2024) 474–488

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Refining computer tomography data with super-resolution networks to increase the accuracy of respiratory flow simulations

Xin Liu [a,1], Mario Rüttgers [a,b,c,*,1], Alessio Quercia [d], Romain Egele [e,g], Elisabeth Pfaehler [d], Rushikesh Shende [b], Marcel Aach [a,f], Wolfgang Schröder [b,c], Prasanna Balaprakash [e,h], Andreas Lintermann [a,c]

[a] Jülich Supercomputing Centre (JSC), Forschungszentrum Jülich GmbH, Germany
[b] Institute of Aerodynamics and Chair of Fluid Mechanics (AIA), RWTH Aachen University, Germany
[c] Jülich Aachen Research Alliance - Center for Simulation and Data Science (JARA-CSD), Germany
[d] Institute of Advanced Simulations 8 (IAS-8), Forschungszentrum Jülich GmbH, Germany
[e] Argonne National Laboratory, United States of America
[f] School of Engineering and Natural Sciences, University of Iceland, Iceland
[g] Université Paris-Saclay, France
[h] Oak Ridge National Laboratory, United States of America

ARTICLE INFO

Keywords:
Respiratory flows
Super-resolution
Computational fluid dynamics
Hyperparameter optimization
Machine learning
Data efficient training

ABSTRACT

Accurately computing the flow in the nasal cavity with computational fluid dynamics (CFD) simulations requires highly resolved computational meshes based on anatomically realistic geometries. Such geometries can only be obtained from computer tomography (CT) data with high spatial resolution, i.e., featuring a ≤ 1 *mm* slice thickness. In practice, CT images are, however, recorded at a lower resolution to not expose patients to high radiation and to reduce the overall costs. To overcome this problem and to provide patients with a detailed physics-based diagnosis, e.g., for surgery planning, the potential of super-resolution networks (SRNs) to increase the CT resolution is analyzed. Therefore, an SRN is developed and trained on CT data. Its predictive performance is improved by an automated hyperparameter optimization technique. The training time is further reduced without predictive accuracy degradation by oversampling images with challenging regions. The performance of the SRN is assessed by an analysis of the reconstructed 3D surfaces of the human upper airway and by comparing results of CFD simulations. That is, surfaces and simulation results based on SRN-generated CT data at 1 *mm* resolution are compared to those obtained from unmodified CT data-sets at low (3 *mm*) and high (1 *mm*) resolution, as well as from CT data interpolated to a 1 *mm* resolution from coarse data. The findings reveal the SRN-based approach to have the lowest deviations in the surfaces and CFD results when compared to those based on the original high-resolution data. The pressure loss between the inflow (nostrils) and outflow (pharynx) regions averaged for three test patients differs by only 1.3%, compared to 8.7% and 8.8% in the coarse and interpolated cases. It is concluded that the SRN-based method is a promising tool to enhance underresolved CT data to yield reliable numerical results of respiratory flows.

1. Introduction

Methods to diagnose pathologies in the human respiratory system have recently evolved to include results of computational fluid dynamics (CFD) simulations. CFD methods allow to numerically quantify the functions of the nasal cavity by analyzing fluid mechanical properties of respiratory flows, e.g., the pressure loss, the temperature distribution, or the mass flux distribution [1]. To accurately compute the flow in the nasal cavity by means of numerical simulations, highly resolved computational meshes based on realistic reconstructed geometries are necessary [2]. Such geometries can only be obtained from medical images that feature a high spatial resolution and enable to accurately identify the interface between tissue and the fluid domain. Computer tomography (CT) recordings, in contrast to magnet resonance tomography (MRT) images, fall into this category.

A CT scan protocol includes multiple parameters such as imaging matrix, slice thickness and gap, number of recordings, reconstruction

* Corresponding author at: Jülich Supercomputing Centre (JSC), Forschungszentrum Jülich GmbH, Germany.
E-mail addresses: xi.liu@fz-juelich.de (X. Liu), m.ruettgers@fz-juelich.de (M. Rüttgers).
[1] Equal contribution.

https://doi.org/10.1016/j.future.2024.05.020
Received 1 January 2024; Received in revised form 8 May 2024; Accepted 14 May 2024
Available online 17 May 2024

filter, etc., which determine the image quality, scanning time as well as radiation dose [3]. When defining the protocol, a compromise between a diagnostically acceptable image quality and feasible costs and radiation doses needs to be achieved. Low-radiation CT is generally preferred in clinical environments to avoid putting the patients' health at risk. However, this leads to low quality images in terms of the signal to noise ratio (SNR) and resolution [3]. Although the relationship of medical imaging radiation doses to cancer rates is still disputable, considering that a large amount of people undergoes this procedure with repeated exposure, it is a clinic's obligation to reduce the radiation dose as much as possible [4].

During a CT scan, consecutive cross-sectional slices in the x- and y-directions with a certain in-plane resolution are obtained along the z-axis (see Fig. 1 for reference of the coordinate system), to cover the 3D region of interest. An accurate reconstruction of geometries requires thin adjacent slices which will significantly increase the radiation dose. It is a trade-off between having a sufficient through-plane resolution in the z-direction, to preserve accurate anatomical features, and reducing the cost and cumulative radiation dose to patients. One can either decrease slice spacing in the scanning process to suffice through-plane resolution, or increase the resolution by employing super-resolution (SR) techniques a posterior the recording, i.e., without increasing the radiation exposure.

SR techniques refer to signal processing approaches to enhance image resolution, overcoming the inherent physical limitations of imaging systems. Traditional SR methods include nonuniform interpolation [5], frequency domain approaches [6], or adding regularization [7]. However, these approaches are often limited to a certain degradation model [8]. Machine learning (ML) tools such as artificial neural networks (ANNs) have recently gained increasing popularity in SR applications. Various architectures of SR networks (SRNs) have been investigated and often achieve or exceed state-of-the art performances. Specifically, the SR convolutional neural network (SRCNN) is considered the pioneering work utilizing deep learning for this task [9,10]. To address more than just one upscaling factor, the more flexible multi-scale deep super-resolution network (MSDSRN) was developed in [11]. In [12], a generative adversarial network (GAN) is combined with skip connections that directly process information at any level to higher layers. This super-resolution GAN (SRGAN) was the first framework capable of inferring photo-realistic natural images with a 4× upscaling factor. In an improved version of the SRGAN, the enhanced SRGAN (ESRGAN) [13], batch normalization is removed, the residual-in-residual approach is integrated into the GAN architecture, and a perceptual loss for outputs right after convolutional layers is added to the loss function. The ESRGAN was found to generate sharper edges and improved textures compared to the SRCNN or SRGAN approaches. Recently, transformer networks have been applied in [14] for non-blind SR tasks. Transformers use an attention mechanism to identify the most important elements of a sequence. The efficient SR transformer was found to be competitive with the previously mentioned networks, while being trained using less computational resources.

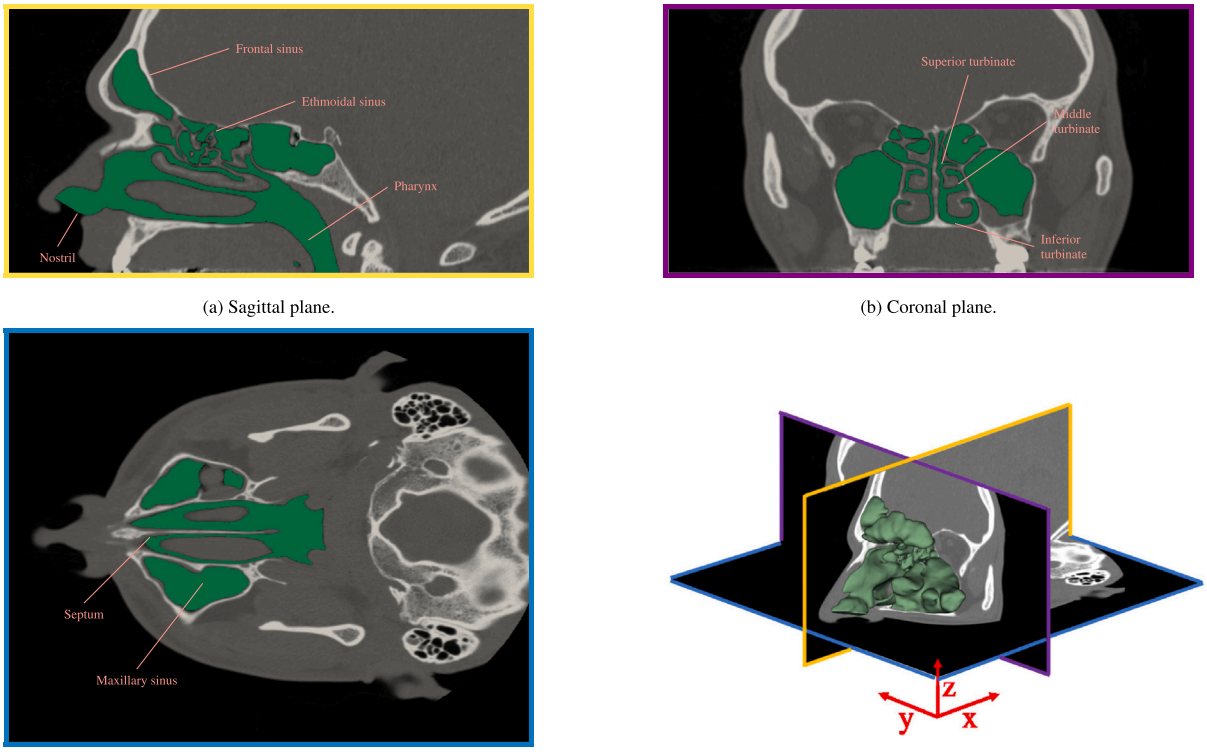
Despite significant progress in SR architectures for general images, their application to medical images requires some additional considerations. First, medical image data-sets are hard to acquire and limited by data privacy issues. This leads to relatively small size data-sets, especially when it comes to obtaining coarse and fine image pairs. Therefore, non-blind SRNs are typically employed for medical images, in which the coarse images are generated from the original images with a known degradation model. Data paucity can be partly tackled by augmentation or transfer learning. To reduce the generalization error, the model can be trained with modified copies of existing data by cropping, rotating, scaling, etc. [15]. In transfer learning, a model is first pre-trained using data with similar visual properties before it is fine-tuned with the medical image data-set [16]. Second, due to limitations in scanning time and radiation dose, medical images are often prone to a low resolution and SNR. Refined anatomical structures make it difficult to develop an SRN that enhances images while preserving crucial features.

The well known U-Net has been extremely successful in medical image analysis since it has been introduced in 2015 in [15]. There has been an explosive number of applications primarily in segmentation tasks [2], but also tasks related to medical image registration [17], SR [18], synthesis [19], removing artifacts [20], etc. The modular and symmetric design of the U-Net architecture makes it easy to incorporate additional components to improve performance. Many U-Net variants have been developed [21], including the inception U-Net [22], attention U-Net [23], or the deep residual U-Net (ResUnet) [24]. The ResUnet architecture [24] was proposed to combine the strength of both residual learning and U-Net, with residual blocks added into the U-Net architecture to ease model training for complex images. The ResUnet is chosen as the SRN in this study.

The above-mentioned ML-based SR models have in common, that the SRNs are trained with pairs of input and the corresponding output images with a higher resolution. For CT recordings in general, such an approach would be sufficient for increasing in-plane resolutions. But for the CT recordings studied here, it is more important to increase the through-plane resolution, because it is directly related to the number of slices, and, therefore, the radiation dose, the quality of reconstructed 3D models of the upper airway, and the picture archiving and communication system (PACS) storage size [4]. For the current data-set, the through-plane resolution along the z-axis could be increased by first reformatting axial slices into sagittal or coronal slices, and then apply single image SR. (see Fig. 1). However, the number of slices in the z-direction varies from patient to patient, which means that the images from the in- and output data-sets have no common size for different patients. A solution is to choose the largest possible size of the complete data-set and up-sample to it. The data-set used in this study includes only few CT recordings that also cover the pharynx region, i.e., having a large number of slices in the z-direction. Using such a method would hence unnecessarily increase memory usage. Another problem occurs, if the trained network is applied in practice and is fed with a CT recording containing more slices in the z-direction than any of the data used for training and testing the SRN. To overcome these limitations, the proposed SRN uses in-plane axial slices from CT data with a low through-plane resolution as input and predicts those axial slices that are missing for a high through-plane resolution. In this way, the proposed SRN can be compatible with other data-sets, leading to an easy deployment for broader clinical workflows and scenarios.

The performance of ML algorithms critically depends on hyperparameters during training [25], which traditionally have to be set by researchers manually. ANNs have a lot of hyperparameters, e.g., there are optimization-related hyperparameters such as the learning rate and batch size, but also architecture-related hyperparameters such as the layer size or the type of activation function. In addition, the optimal hyperparameters are task-specific, which makes their optimization necessary for different datasets. In practice, experts are often going through a trial-error process to guess optimal hyperparameters. Therefore, to ease this task, hyperparameter optimization (HPO) algorithms have been developed.

There exist many HPO algorithms in literature but a large portion of them can be classified as black-box optimization algorithms, i.e., they are based only on input–output pairs of the optimized function. Model-free methods such as grid and random search [26] are flexible but become exponentially inefficient with the number of hyperparameters to be optimized. On the other hand, Bayesian optimization (BO) uses surrogate models to learn the underlying mapping between the model hyperparameters and performance, which usually makes it more efficient [27]. A major hurdle in HPO is the cost to repeatedly train a model given a suggested set of hyperparameters during the process. A way to reduce this cost, also frequently employed in manual tuning, is to limit the maximum budget for each evaluation. For instance,



(a) Sagittal plane.

(b) Coronal plane.

(c) Axial plane.

(d) Coordinate system and combination of all views.

Fig. 1. Coordinate system and examples for sagittal (yellow), coronal (purple), and axial (blue) planes of CT imaging data showing the human upper airway. The current data-set for each patient consists of cross-sectional axial slices in the x and y-axis (c), stacked along the z-axis to cover the 3D region of interest. The 3D volumetric data is reformatted into sagittal (a) and coronal (b) planes for representation. A segmentation separating the airway from other matter is illustrated in green. The figure originates from [2], where it is used to explain the anatomy and functionality of the nose, and is used with the permission of the authors.

training only with a few iterations or on a subset of data. Multi-fidelity HPO methods adopt such heuristics by doing early discarding of low-performing candidates at early iterations. The choice of the multi-fidelity algorithm is essentially a trade-off between the quality of the approximated error and the available run-time (i.e, compute budget). The current study is guided by previous benchmarks [28], i.e., the HPO approach combines Bayesian and multi-fidelity optimization to determine a set of suitable hyperparameters for the SRN.

To the best of the authors' knowledge, the present study employs SRNs for the first time to enhance a CFD-based analysis of respiratory flows. To evaluate the quality of the predicted images, the output of trained SRNs is compared to (i) original CT data with a fine resolution, (ii) original CT data with a coarse resolution, and (iii) interpolated CT data. The comparison also takes physical properties into account by considering the results of numerical flow simulations through the nasal cavities. The surface of the airways is extracted from the CT data by employing the pipeline developed in [2]. Simulations are carried out with the highly scalable lattice-Boltzmann method (LBM) of the simulation framework multiphysics-Aerodynamisches Institut Aachen (m-AIA), which is the successor of the framework Zonal Flow Solver [29].

The manuscript is structured as follows. Section 2 explains the computational methods, the results are presented in Section 3, and Section 4 summarizes and concludes the work.

2. Methods

This section provides an overview over the computational methods used in the current study. Section 2.1 explains the CT data and the sampling strategy to generate coarse and fine training pairs. In Section 2.2, the SRN architecture is described, followed by a description of how to find a suitable set of hyperparameters in Section 2.3. A data-efficient method based on the sample importance is described in Section 2.4, which further accelerates training convergence. Finally, Section 2.5 provides information on the methods for generating the computational meshes, conducting simulations, and imposing the boundary conditions.

2.1. Data pre-processing

Anonymized head CT recordings from 65 patients are used. The data-set is composed of axial slices with a matrix size of 512 × 512 pixels. The in-plane pixel size ranges from 0.3 mm to 0.9 mm, and the slice thickness from 0.2 mm to 1.0 mm. To establish a uniform basis for the training data, the through-plane resolution along the z-axis is resliced to a common slice thickness for all patients. The uniformly resampled slices are used as ground truth in later steps. Studies on bone and skull reconstruction suggest using at least a slice thickness of 1.0 mm to keep the reconstruction error at a reasonable amount [30,31]. Therefore, 1.0 mm is chosen as the slice thickness for the ground truth data-set.

The sampling strategy used to generate pairs of input (coarse) and ground truth (fine) data to train the SRN is shown in Fig. 2. A lack of matching coarse and fine data of the same patient means that the focus is only on non-blind SR, a common practice for medical imaging applications where coarse data are down-sampled from fine data. The coarse data-set is generated from the fine data-set by keeping one slice out of three, mimicking two different clinical protocols with a 1.0 mm z-axis resolutions where thin slices are scanned adjacent to each other for the fine data-set, and a 3.0 mm z-axis resolutions where slices are scanned sparsely for the coarse data-set, respectively. As shown in Fig. 2, two axial slices with a spacing of 3.0 mm (red) are used as the input, and the two enclosed slices (green) are used as ground truth for the SRN. The prediction from the SRN is concatenated to the input

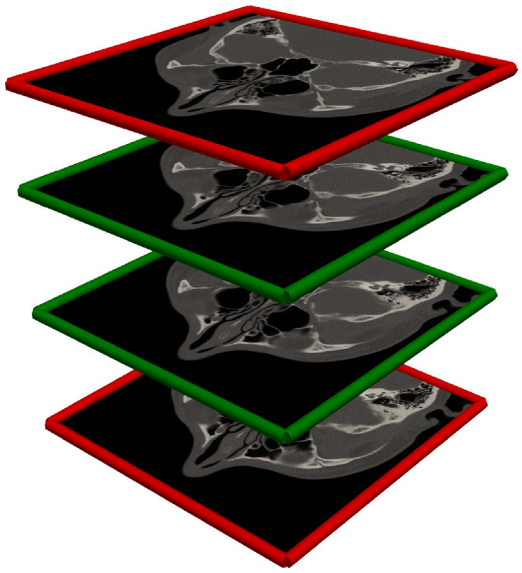


Fig. 2. Illustration of the sampling strategy to generate pairs of input (red) and ground truth (green) data for training the SRN. The coarse data-set with 3.0 mm through-plane resolution is generated by keeping one out of three 1.0 mm slices of the original fine data-set.

data to generate the final output. To generate as many training pairs as possible, a sliding window approach is applied along the z-axis, i.e., the same axial slice can appear in multiple input/output pairs.

In each of the above-described data-sets, images from 62 patients are used for training and validation, and images from 3 patients (I, II, III) for testing. In the training part, slices are randomly flipped along the x-axis to increase the number of training samples.

2.2. Network architecture

The employed SRN is inspired by the ResUnet introduced in [15,24]. The architecture of the SRN is depicted in Fig. 3. Residual units are added into the plain U-Net structure to ease training of deep networks [24]. Like in the original U-Net, the SRN has a contracting path to extract features and an expanding path to recover the resolution of the input data. On each resolution level in the contracting path, a convolutional block (CB), the combination of a 3×3 convolutional layer (Conv), batch normalization (BN) [32], and leaky rectified linear units (LReLU) [33] are applied twice. Subsequently, a 2×2 max-pooling (MP) layer with stride 2 is employed for down-sampling. The LReLU activation function is known to prevent the dying ReLU problem by setting a small but non-zero gradient for negative values, i.e.,

$$LReLU(\mathcal{X}) = \begin{cases} \mathcal{X} & \text{if} \quad \mathcal{X} \geq 0 \\ \alpha\mathcal{X} & \text{if} \quad \mathcal{X} < 0. \end{cases} \tag{1}$$

The slope α is set to 10^{-2}. Batch normalization is used to reduce overfitting and to increase the learning stability by shifting the layers' inputs to a zero-mean and unit variance. After the max-pooling step, the resolution is halved and the number of channels doubled. In the expanding path, up-sampling is implemented with deconvolution layers.

In total, there are four resolution levels in this network. On each level, the number of feature maps (nFC) is doubled. The number of feature maps at the first level is determined by the HPO process. Short skip connections are implemented between the two CBs for residual learning. Long skip connections to concatenate feature maps from the contracting path to the corresponding level in the expanding path are implemented to facilitate information propagation. The network is initialized with the Xavier method [34]. Optimizer configurations are selected later in the HPO process to minimize the mean-squared error (MSE) between the SRN output and ground truth. The training data-set X_{train} is normalized to zero mean and unit standard deviation before feeding it into the SRN, i.e., $X_{train} = (X_{train} - \overline{X}_{train})/\sigma_{X_{train}}$, where $\overline{X}_{train}$ and $\sigma_{X_{train}}$ are the mean and standard deviation of the training data-set. The network is implemented with the PyTorch framework [35].

2.3. Hyperparameter selection

The hyperparameter configuration of the SRN is selected by multi-fidelity BO, cf. Section 1 and the following Section 2.3.1. The HPO problem can be formulated as follows. Consider an ML algorithm $\mathcal{A}_\lambda$ that aims to learn the mapping $\mathcal{F}(\mathcal{X})$ to a minimized loss $\mathcal{L}(\mathcal{X}; \mathcal{F})$, where $\mathcal{X}$ are samples from a certain distribution $\mathcal{G}_\mathcal{X}$ and λ is the hyperparameter vector. The hyperparameter space can be denoted as $\boldsymbol{\Lambda} = \Lambda_1 \times \Lambda_2 \times \cdots \Lambda_N$, where N is the total number of hyperparameters. The goal of HPO is to find the hyperparameter vector $\lambda^* = \lambda_1, \lambda_2, \ldots, \lambda_N$ that minimizes the generalization loss

$$\lambda^* = \underset{\lambda \in \boldsymbol{\Lambda}}{argmin}\, \mathbb{E}_{X \sim \mathcal{G}_\mathcal{X}} \boldsymbol{V}(\mathcal{L}, \mathcal{A}_\lambda, X_{train}, X_{valid}), \tag{2}$$

where X_{train} and X_{valid} are training and validation data-sets that are a finite set of samples from $\mathcal{G}_\mathcal{X}$. The variable $\boldsymbol{V}$ represents the validation strategy. 20% data points of the training data-set are randomly selected to calculate the validation loss to select the hyperparameter configuration.

2.3.1. Multi-fidelity Bayesian optimization

A flow-chart depicting the multi-fidelity BO process is shown in Fig. 4. BO is a well-established model-based HPO method with two features that make it efficient: a surrogate model to learn the underlying mapping between model performance and hyperparameter configuration, and an acquisition function to decide which configuration is best to evaluate next. After each iteration, observations of the hyperparameter configurations and objective function are fit into the surrogate model. Here, random forest is chosen as the surrogate model as it is capable of handling a large number of evaluations [36]. The acquisition function calculates the predictive distribution of the surrogate model and determines the next sample point. The upper confidence bound (UCB) algorithm is used

$$UCB_\kappa(\lambda) = \mu(\lambda) + \mathcal{K}\sigma(\lambda), \tag{3}$$

where μ is the expected performance observed so far and σ is the uncertainty described by the standard deviation. The quantity $\mathcal{K}$, which explicitly trades off exploration and exploitation, is set to $\mathcal{K} = 1.96$. This corresponds to 95% of the confidence interval.

Multi-fidelity methods use cheap but lower fidelity approximations to evaluate the objective function. Examples of approximations include training the model on subsets of the data, using subsets of features, or lower number of epochs, etc. Multi-fidelity approaches include learning-curve extrapolation [37] and bandit-based algorithms [38]. Combining them with BO achieves further speed-up. The successive halving algorithm (SHA) [39] is a simple yet powerful bandit-based algorithm. After training all configurations with a small initial budget (where budget in most cases refers to the number of training iterations), only top-performing ones are assigned to an increased budget and the rest is cut. This process is repeated to subsequent cut-off points until the maximum budget is reached. The asynchronous SHA (ASHA) further increases the scalability by deciding which trials are promising on a rolling basis [40], mitigating the problem of idle workers. The ASHA uses parallel resources more efficiently and is used to select the hyperparameter configuration of the SRN.

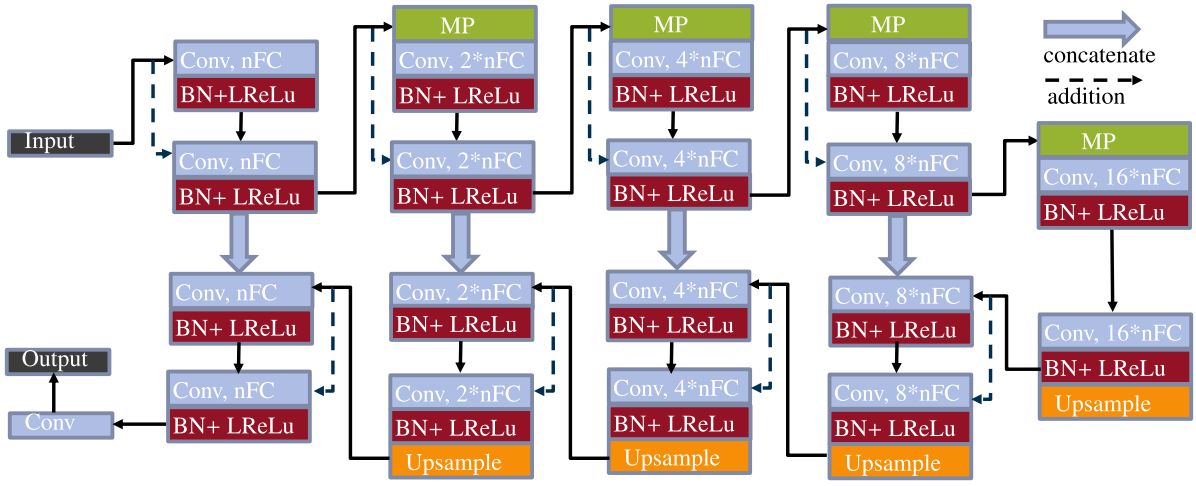


Fig. 3. Architecture of the SRN. Abbreviations: Conv: convolutional layer, BN: batch normalization, LReLu: leaky rectified linear units, MP: max-pooling, number of feature channels nFC.

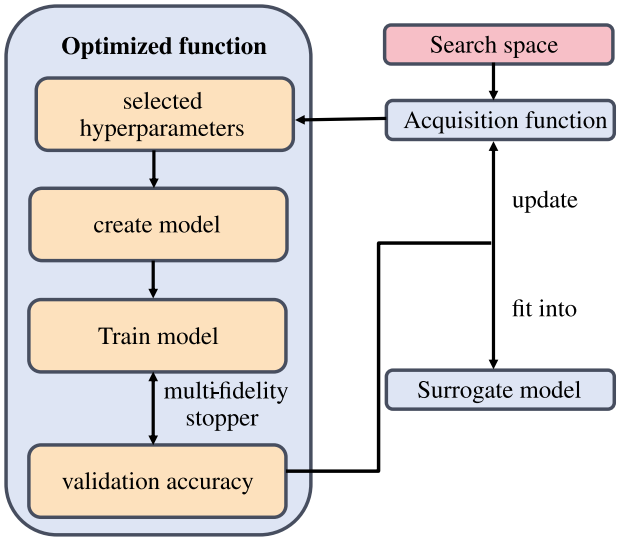


Fig. 4. Flow chart of multi-fidelity Bayesian optimization.

Table 1
HPO search space and result for the SRN. The optimizer settings are ADAM or SGD with *momentum* = 0.9 (the type is `[cat]` = categorial); LR = learning rate, sampled logarithmically; KS = kernel size; nFC = number of feature channels, each of type `[real]` or `[int]` = integer.

	Optimizer	LR	KS	nFC
Range	ADAM, SGD `[cat]`	10^{-4} to 1 `[real]`	3, 5 `[int]`	32, 64, 96 `[int]`
Results	SGD	0.017	3	96

2.3.2. Experimental settings using DeepHyper

In this work, the well-established AutoML package DeepHyper [41] is used for HPO. First, the search space is defined with a combination of categorical- (`[cat]`), real- (`[real]`), and integer- (`[int]`) valued hyperparameters, including optimizer settings, number of feature channels nFC, and convolution kernel size KS, as listed in Table 1. Two optimizers can be selected, i.e., the adaptive moments estimation (ADAM) optimizer [42] and the stochastic gradient descent (SGD) approach [43] with a momentum of 0.9. The learning rate LR is sampled logarithmically from 10^{-4} to 1.

An initial set of configurations is generated from the defined search space. Each trial is sent to a single graphics processing unit (GPU) for training by Ray [44], a compute framework to scale Python workloads. The returned objectives of trials (in this case the MSE) are gathered on the master node, to determine whether the trial should be cut-off or continued. The surrogate model and acquisition function are updated, and new configurations are sampled and distributed for evaluation. In the DeepHyper implementation, the halving schedule follows a geometric progression to reduce the number of configurations by a reduction factor r. The first halving step is done after a minimum number of steps ms. The next halving step is performed after $ms \times r$ steps, followed by another one after $ms \times r^2$ steps. The choice of r and ms determines how aggressive the reduction scheme acts. Excessive reduction can cause premature termination of good configurations, while inadequate reduction can lead to wasting resources on running poor configurations for too long. It is a trade-off between approximation error and saved computational resources, or within a fixed total amount of computational resources, the number of trials to be evaluated. Here, the default settings of DeepHyper are used with $ms = 1$ and $r = 3$.

2.4. Biasing SGD towards relevant image regions

The data-set features a high percentage of homogeneous regions with, e.g., fully black or gray regions, which are easy to super-resolve. It has been hypothesized that oversampling more complex image regions after a short pre-training phase can lead to faster convergence. To achieve this, a data efficient method inspired by the work in [45] is employed together with image tiling. On top of the pre-processing steps described in Section 2.1, every 512 × 512 image is tiled into 64 tiles of size 64 × 64, and the batch size is increased by a factor of 64 to 256. The SRN hyperparameters are, however, taken based on the search described in Section 2.3, to keep the same amount of information in a single step compared to the baseline using the full images. The SRN is trained using uniform SGD without replacement for $\mathcal{E} = 10$ epochs. At epoch $\mathcal{E}$, the loss is stored and ranked for each training sample. From this ranking, a probability distribution is derived for the subsequent training epochs.

2.5. Numerical methods

Unstructured hierarchical cartesian meshes are created with the massively parallel mesh generator of m-AIA [46]. The meshes are based on an octree structure derived from iteratively subdividing an initial cube surrounding the region of interest (ROI), i.e., the nasal cavity [47]. In more detail, the initial cube is decomposed into eight sub-cubes, which are then further refined for a pre-defined number of steps. The parent–child relations of cubes and sub-cubes constitute the octree

structure of the mesh. Cells that are located outside the ROI are deleted. For the parallelization of the computation, the domain is decomposed by a *Hilbert* decompositioning method using space filling curves as described in [48] on a pre-defined refinement level. *Z-curves* [49] are additionally used to index the remaining levels. The resulting mesh is stored by efficient parallel I/O routines using the *network common data form* (NetCDF) format [50]. For more details, the reader is referred to [46].

Simulations are conducted with a thermal version of the LBM. The governing equations are based on the BGK model approximation [51]:

$$\frac{\partial \boldsymbol{f}}{\partial t} + \boldsymbol{\xi} \cdot \boldsymbol{f} = -\omega(\boldsymbol{f} - \boldsymbol{f}^{eq}), \tag{4}$$

$$\boldsymbol{f}^{neq} = \boldsymbol{f} - \boldsymbol{f}^{eq}, \tag{5}$$

with the particle probability distribution functions (PPDFs) $\boldsymbol{f} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\xi}, t)$, the time t, the location $\boldsymbol{x}$, the microscopic velocity $\boldsymbol{\xi}$, the collision frequency ω, and the Boltzmann-Maxwellian distribution function $\boldsymbol{f}^{eq}$. In [51], Eq. (4) is discretized for a small time increment δt yielding

$$\begin{aligned}&\boldsymbol{f}_i(\boldsymbol{x} + \boldsymbol{\xi}_i \delta t, \boldsymbol{\xi}, t + \delta t) - \boldsymbol{f}_i(\boldsymbol{x}, \boldsymbol{\xi}_i, t) = \\ &- \omega(\boldsymbol{f}_i(\boldsymbol{x}, \boldsymbol{\xi}_i, t) - \boldsymbol{f}_i^{eq}(\boldsymbol{x}, \boldsymbol{\xi}_i, t)),\end{aligned} \tag{6}$$

using on the D3Q27 model [52], with $i \in \{1, 2, 3, \ldots, Q\}$ directions. The discrete Boltzmann-Maxwellian distribution function reads as

$$\boldsymbol{f}_i^{eq} = w_i \rho \left(1 + \frac{\boldsymbol{\xi}_i \cdot \boldsymbol{u}}{c_s^2} + \frac{1}{2}\left(\frac{\boldsymbol{\xi}_i \cdot \boldsymbol{u}}{c_s^2}\right)^2 - \frac{\boldsymbol{u} \cdot \boldsymbol{u}}{2c_s^2}\right), \tag{7}$$

with the isothermal speed of sound $c_s = 1/\sqrt{3}$, the density ρ, the fluid velocity vector $\boldsymbol{u} = (u, v, w)^T$, and the weight coefficients w_i [52].

To simulate the temperature distribution, a second set of PPDFs g_i is solved by

$$\begin{aligned}&\boldsymbol{g}_i(\boldsymbol{x} + \boldsymbol{\xi}_i \delta t, \boldsymbol{\xi}_i, t + \delta t) - \boldsymbol{g}_i(\boldsymbol{x}, \boldsymbol{\xi}_i, t) = \\ &- \omega_t(\boldsymbol{g}_i(\boldsymbol{x}, \boldsymbol{\xi}_i, t) - \boldsymbol{g}_i^{eq}(\boldsymbol{x}, \boldsymbol{\xi}_i, t)) \\ &+ (\omega_t - \omega)\left(\boldsymbol{\xi} \cdot \boldsymbol{u} - \frac{\boldsymbol{u} \cdot \boldsymbol{u}}{2}\right)(\boldsymbol{f}_i(\boldsymbol{x}, \boldsymbol{\xi}_i, t) - \boldsymbol{f}_i^{eq}(\boldsymbol{x}, \boldsymbol{\xi}_i, t)),\end{aligned} \tag{8}$$

where the thermal collision frequency ω_t is related to the thermal conductivity κ by [53]

$$\kappa = c_s^2 \left(\frac{\delta t}{\omega_t} - \frac{\delta t}{2}\right). \tag{9}$$

The thermal equilibrium distribution reads

$$\begin{aligned}\boldsymbol{g}_i^{eq} = w_i \rho c_s^2 \Bigg[&\frac{\boldsymbol{\xi}_i \cdot \boldsymbol{u}}{c_s^2} + \left(\frac{\boldsymbol{\xi}_i \cdot \boldsymbol{u}}{c_s^2}\right)^2 \\ &- \frac{\boldsymbol{u}^2}{c_s^2} + \frac{1}{2}\left(\frac{\boldsymbol{\xi}_i^2}{c_s^2} - D\right)\Bigg] + E f_i^{eq},\end{aligned} \tag{10}$$

with the spatial dimension D, the total energy $E = \epsilon + 1/2|\boldsymbol{u}|^2$, the internal energy $\epsilon = (Dc_s^2)/2T$, the velocity magnitude $|\boldsymbol{u}|$, and the temperature T.

The macroscopic variables ρ, $\boldsymbol{u}$, and T can be computed by

$$\rho = \sum_{i=1}^{Q} f_i, \tag{11}$$

$$\rho \boldsymbol{u} = \sum_{i=1}^{Q} \boldsymbol{\xi}_i \cdot f_i \tag{12}$$

$$\rho E = \sum_{i=1}^{Q} g_i. \tag{13}$$

The static pressure p_s is obtained from the density by $p_s = c_s^2 \rho$. Note that p_{tot} is the total pressure, expressed as the sum of the static pressure p_s and the dynamic pressure $p_d = \rho|\boldsymbol{u}|^2/2$.

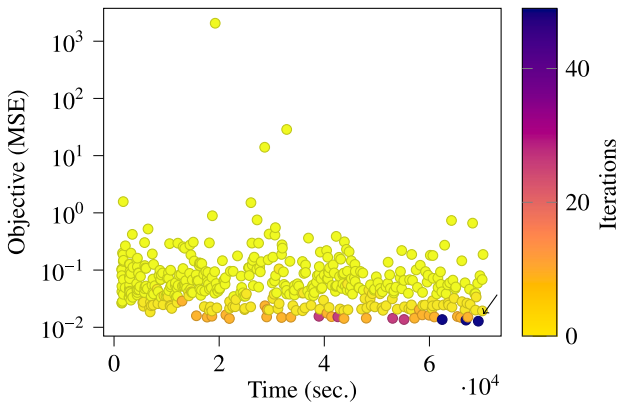


Fig. 5. MSE, time of submission, and the number of iterations for each HPO configuration. The best performing job is highlighted by a black arrow.

An interpolated bounce-back scheme is used to satisfy the no-slip condition at the inner walls of the nasal cavity [54]. Furthermore, the inner walls are modeled as isothermal walls at body temperature $T_B = 309.15$ K. At the inlets (nostrils), the equation of Saint-Venant and Wantzel is applied [1] and the temperature is set to the ambient air temperature of $T_{amb} = 293.15$ K. At the outlet (pharynx), the pressure is iteratively adapted to fit the prescribed volume flux $\dot{V}_P = 250$ ml/s [1], i.e., to the corresponding Reynolds number $Re_P = (\dot{V}_P \cdot d_P)/(A_P \cdot \nu)$, which is calculated from the hydraulic diameter of the pharynx d_P, the pharyngeal cross-sectional area A_P, and the kinematic viscosity $\nu = 1.63 \times 10^{-5}$ m^2/s. The velocity and temperature are extrapolated from the inner cells. All computational meshes have a resolution of $\Delta x = 0.09$ mm to accurately resolve thin channels and boundary layers [1,55].

3. Results

In this section, the effect of HPO and data efficiency on the baseline SRN is analyzed. The performance of the trained SRNs is evaluated by means of simulating the flow through 3D models extracted from CT data. Section 3.1 focuses on identifying suitable hyperparameters with HPO. The outcome is used in Section 3.2 to further accelerate the ML pipeline with a data efficient method, without degrading its predictive capabilities. Section 3.3 concentrates on a comparison between simulation results based on the original, interpolated, and SRN-generated CT data.

3.1. Hyperparameter tuning of the network

The HPO experiment described in Section 2.3 is conducted on the GPU partition of the JURECA-DC cluster [56] installed at the Jülich Supercomputing Centre (JSC), Forschungszentrum Jülich. Each node is equipped with four NVIDIA A100 GPUs and two AMD EPYC 7742 CPU with 64 cores clocked at 2.25 GHz. The total computation budget assigned to the HPO experiment is 24 h on 8 GPU nodes. In total, 407 configurations are investigated.

In Fig. 5, the objective (MSE) achieved with all investigated hyperparameter configurations is plotted against the time of submission to the master node. Each dot represents a single configuration and is colored by the number of training iterations. The best performing job is highlighted by a black arrow in Fig. 5. Its configuration, i.e., using $KS = 3$, $nFC = 96$, $LR = 0.017$, and SGD as optimizer, see Table 1, is chosen for the next steps.

3.2. Data efficient training with tiled images

In the following, the SR results obtained using the baseline network and the data efficient method described in Section 2.4 are validated. In addition to the MSE, the following commonly used metrices are reported to measure the quality of the SR-generated images in terms of similarity between low-resolution and super-resolved images: structural similarity index measure (SSIM), which quantifies the perceived quality by the human visual system, and peak signal to noise ratio (PSNR), which quantifies the differences at the pixel level [57].

Fig. 6 shows the data efficient method using tiled images [45] (SGD-biased) to be faster and more stable to achieve a similar validation SR quality (in terms of MSE loss, SSIM, and PSNR) compared to the baseline U-Net (SGD-baseline). The hyperparameters tuned for the baseline are re-used and the method parameter $\mathcal{E}$ (representing the pre-training epochs) is set to $\mathcal{E} = 10$ as suggested in the original paper [45]. In fact, the method achieves in only ≈6 h a similar SR quality compared to training the baseline, which requires ≈14 h, i.e., a ≈2.3× speed-up is achieved. This allows the applications early stopping criteria and save time during training without affecting the SR quality. An early stopping criterion is used for the validation MSE loss, which stops training if the validation MSE does not improve for 3 consecutive validation epochs. Then, the best checkpoint is selected (after 19 epochs, ≈5 h and 30 minutes of run time) out of 4 independent runs with different random seeds to run the simulations, see Section 3.3.

3.3. CFD simulations

For each of the three test patients I, II, and III, five simulations are conducted:

(A) based on the fine CT data-set (ground truth),
(B) the coarse CT data-set (input to SRN),
(C) interpolation,
(D) the output of the baseline SRN,
(E) and the output from accelerated SRN training using the data efficient method [45] and image tiling.

Each simulation is run on 5 nodes (20 GPUs) of the NVIDIA A100 GPU partition of the JURECA-DC cluster [56]. In case (C), coarse input data is up-sampled along the z-axis with cubic spline interpolation.

The performance of the SRN is first assessed by comparing the 3D nasal cavity models that are extracted from the CT data of cases A-E. They are the basis for each CFD simulation. Fig. 7 illustrates the 3D nasal cavity models of each patient and case, which are obtained using the automated pipeline presented in [2]. In Fig. 7(a) it is shown that in cases I(B) and I(C) the airway in the inferior turbinate of the right nasal passage (from the patient's view) is narrowed by an obstacle (red circles), which is not present at the same location for cases I(A), I(D), and I(E). Together with the middle and the superior turbinates, the inferior turbinates play an important role by warming and humidifying inspired air and by regulating nasal airflow, see Fig. 1(b). Similar observations can be made for test patient II in Fig. 7(b), except that the obstacle at the inferior turbinates is larger and blocks almost the complete turbinate of the left nasal passage (green circles). Additionally, in cases II(B), II(C), and II(E) all frontal sinuses are connected, whereas in cases II(A) and II(D) only the right part of the frontal sinuses is included. Fig. 7(c) reveals narrowed inferior turbinates in the left and right nasal passages for cases III(B) and III(D) (blue circles), which are not visible in cases III(A), III(D), and III(E). For all three test patients, the edges of the inflow areas of case (B) are less smooth compared to the remaining cases.

Apart from comparing the 3D models of cases A-E, analyzing numerical properties helps to assess the predictive performance of the SRN. Table 2 shows Re_p, the number of cells N_c, the total pressure loss between the inlets and outlet Δp_{tot}, and the temperature increase between the inlets and outlet ΔT for all cases. Percentage deviations of

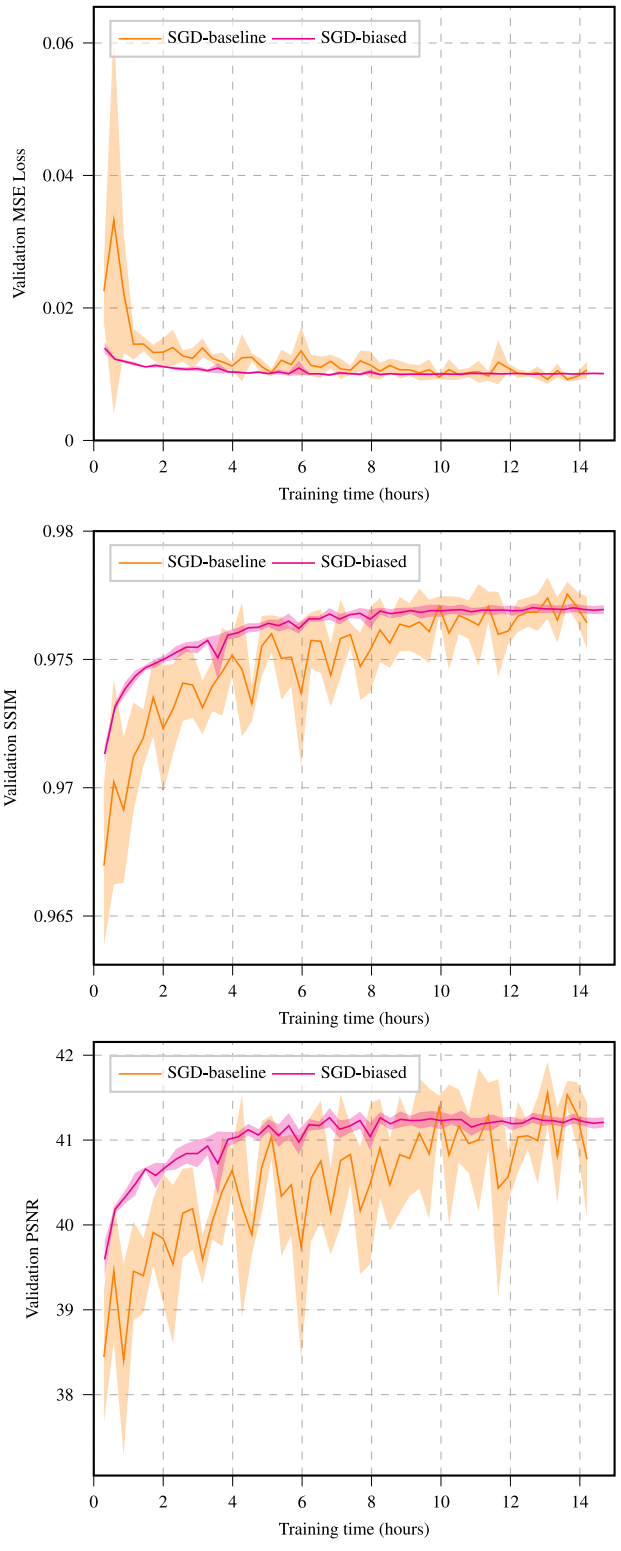
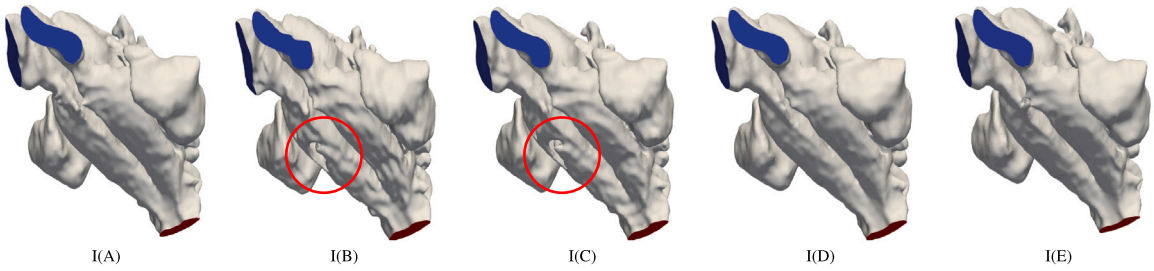


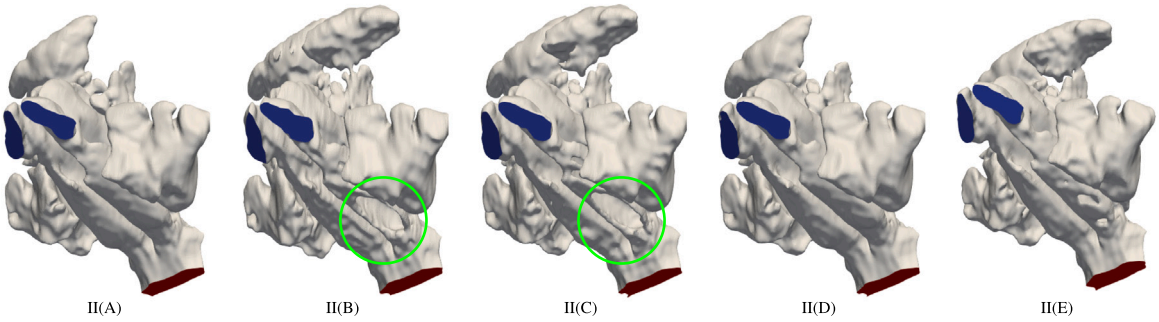
Fig. 6. Validation mean-squared error (MSE) loss, structural similarity index measure (SSIM), and peak signal to noise ratio (PSNR) of the baseline SGD (using full images) and the SGD-biased method (using tiled images).

cases (B)–(E) compared to the reference case (A) are given in brackets. The pressure loss Δp and temperature increase ΔT are defined as

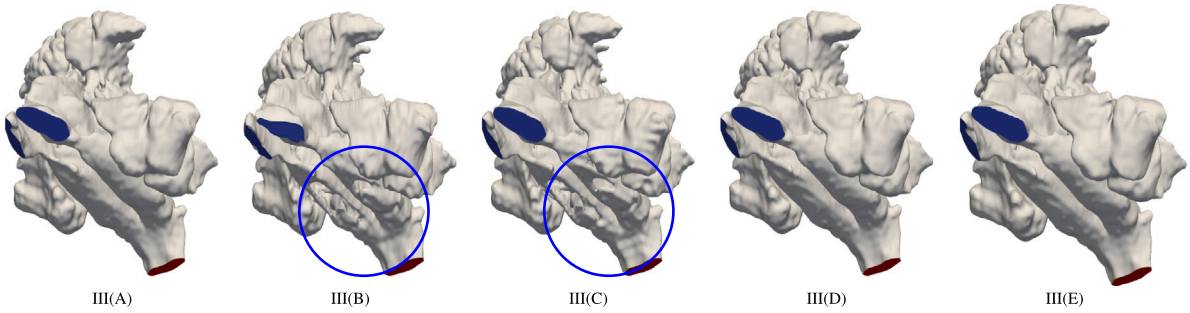
$$\Delta p_{tot} = \frac{1}{N_a} \sum_{i=1}^{N_a} \left(\frac{1}{H_{out}} \sum_{j=0}^{H_{out}} p^i_{tot,j} - p_{amb} \right), \tag{14}$$



(a) Test patient I.



(b) Test patient II.



(c) Test patient III.

Fig. 7. 3D models based on the fine CT data-set (A), coarse CT data-set (B), interpolated CT data (C), CT data generated by the baseline SRN (D), and CT data generated by the SRN improved using the data efficient method (E). The inflow areas (nostrils) are colored blue and the outflow area (pharynx) is colored red. The colored circles highlight examples for differences between the five cases.

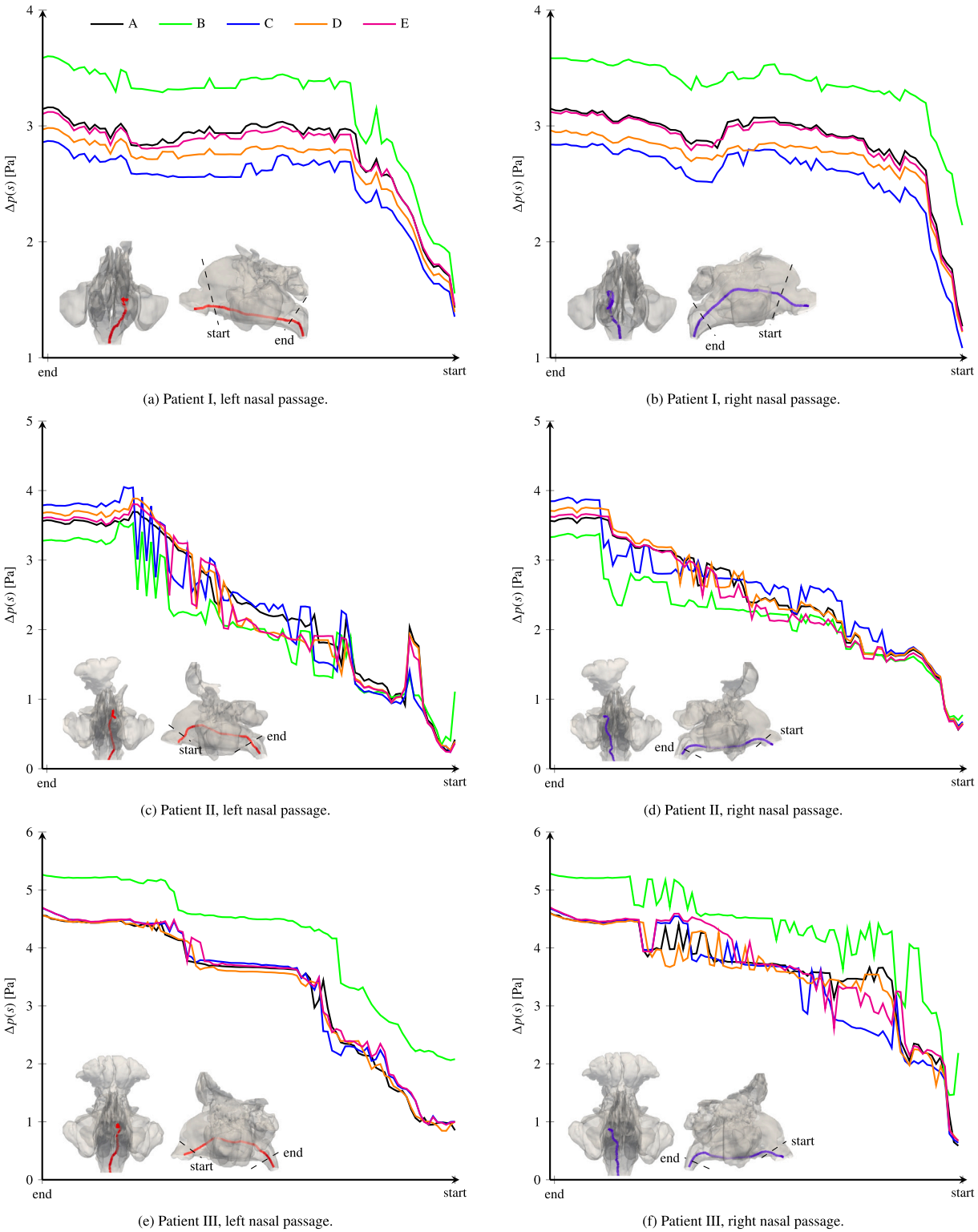
$$\Delta T = \frac{1}{N_a}\sum_{i=1}^{N_a}\left(\frac{1}{H_{out}}\sum_{j=0}^{H_{out}} T_j^i - T_{amb}\right), \tag{15}$$

where N_a is the number of time steps used for temporal averaging, H_{out} is the number of boundary cells at the pharynx, and p_{amb} is the ambient pressure. Additionally, the average absolute deviations (AAD) of cases (B)–(E) compared to case (A) are provided.

In all cases, the Reynolds numbers at the pharynx based on the two SRN outputs show the smallest deviations from Re_P based on the fine CT data. In fact, Re_p of case (E) has with AAD = 0.6% the smallest mean deviation. The same trend is observed for the number of cells and for results from flow computations in terms of Δp_{tot} and ΔT, where quantities based on the SR-generated CT data have the lowest AADs. Whereas case (E) has a lower AAD for Δp_{tot} (1.3%) compared to case (D) (2.5%), case (D) performs slightly better for ΔT (0.2%) compared to case (E) (0.3%). Although computations based on the interpolated data have a high accuracy for some cases (Δp_{tot} for patient III, ΔT for patient I), large fluctuations in most of the remaining cases (Δp_{tot} for patient I and II, ΔT for patients II and III) indicate a low reliability. Computations based on coarse CT data show no advantage in any of the cases.

Figs. 8 and 9 provide a quantitative assessment of the simulation results of the different cases. The figures show the total pressure loss $\Delta p(s) = p_{amb} - \hat{p}(s)$ and temperature increase $\Delta T(s) = \hat{T}(s) - T_{amb}$, where $\hat{p}(s)$ and $\hat{T}(s)$ are area-averaged quantities at a downstream cross section of a location s along the left or right centerline. The left and right centerlines of case (A) of each test patient are illustrated in the corresponding subfigures. The normal vector of a cross section is computed based on two consecutive locations along the corresponding centerline. The area of a cross section is determined with a region growing algorithm whose seed point lies at s. The centerlines are computed using the Vascular Modeling Toolkit (VMTK) [58].

The results shown in Figs. 8 and 9 do not start and end at the first and last location of a centerline. At these locations, the cross sections tend to cut through the inlets or outlet. Here, area-averaged flow fields do not represent the flow realistically. Therefore, the plots in Figs. 8 and 9 are bounded by locations inside of the nasal cavity, whose start and end points are illustrated for each test patient by the dashed black lines. Note that large fluctuations along the centerlines can occur due to turbinates being connected to the maxillary sinuses in some cross sections. In other cross sections only the flow through the turbinates is computed. Small fluctuations occur when two consecutive cross-sectional areas intersect.



(a) Patient I, left nasal passage.

(b) Patient I, right nasal passage.

(c) Patient II, left nasal passage.

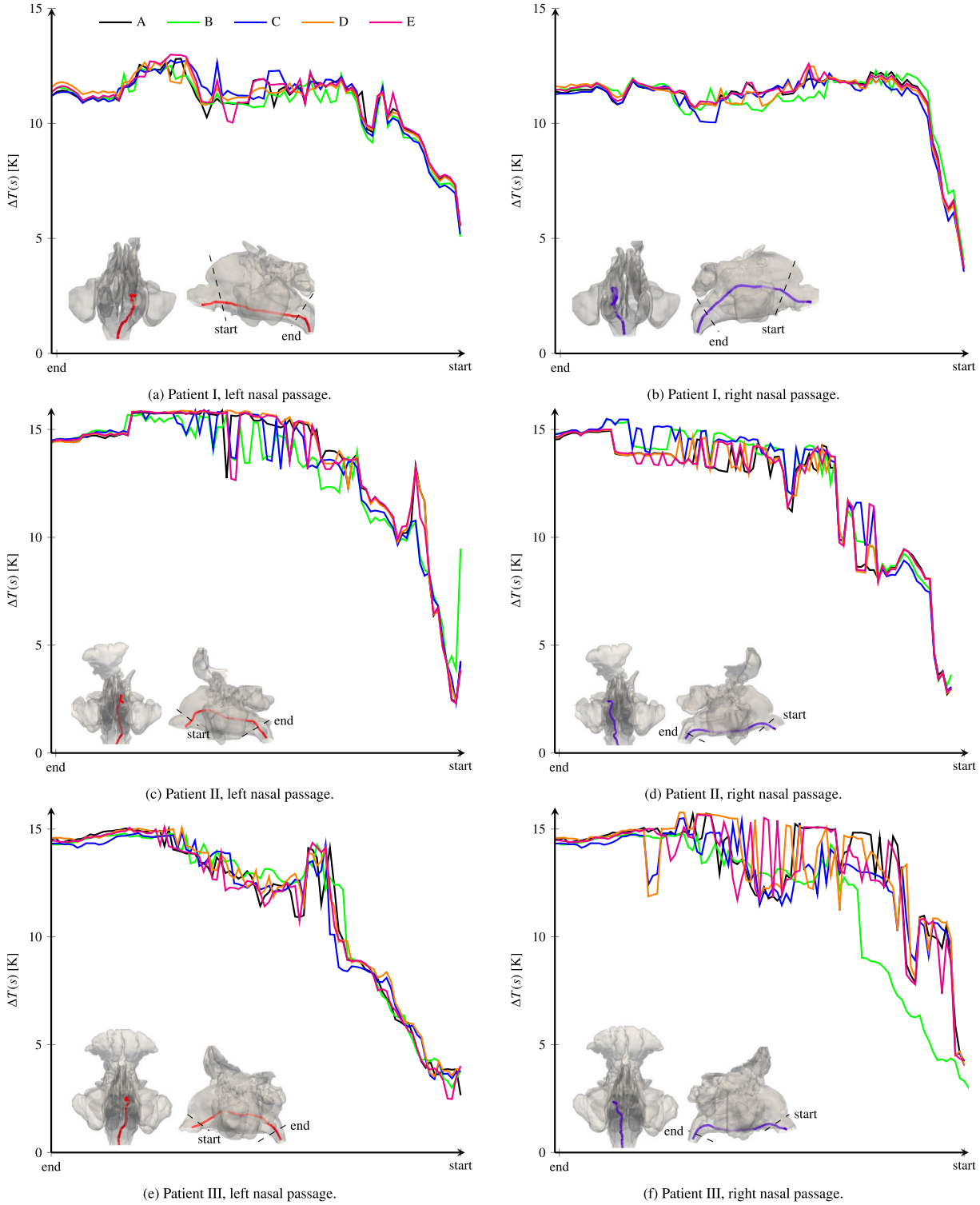
(d) Patient II, right nasal passage.

(e) Patient III, left nasal passage.

(f) Patient III, right nasal passage.

Fig. 8. Total pressure loss averaged over cross-sectional areas along centerlines through the left (red) and right (violet) nasal passages. The black lines represent simulation results based on the CT data with high resolution (A), green lines simulation results based on CT data with low resolution (B), blue lines simulations results based on interpolated CT data (C), orange lines simulation results based on the CT data generated by the SRN (D), and magenta lines stand for improved results of the SRN by employing the data efficient method (E). The start and end points in the nasal cavity are illustrated for each test patient by the dashed black lines.

Figs. 8(a) and 8(b) reveal that the pressure loss is continuously overpredicted in case I(B) and underpredicted in case I(C). The simulation of case I(D) slightly underpredicts the pressure loss, whereas the results of case I(E) are in line with those from case I(A). For test patient II, Figs. 8(c) and 8(d) show that case II(B) tends to underpredict the pressure loss. The pressure loss in case II(C) coincides well with case



(a) Patient I, left nasal passage.

(b) Patient I, right nasal passage.

(c) Patient II, left nasal passage.

(d) Patient II, right nasal passage.

(e) Patient III, left nasal passage.

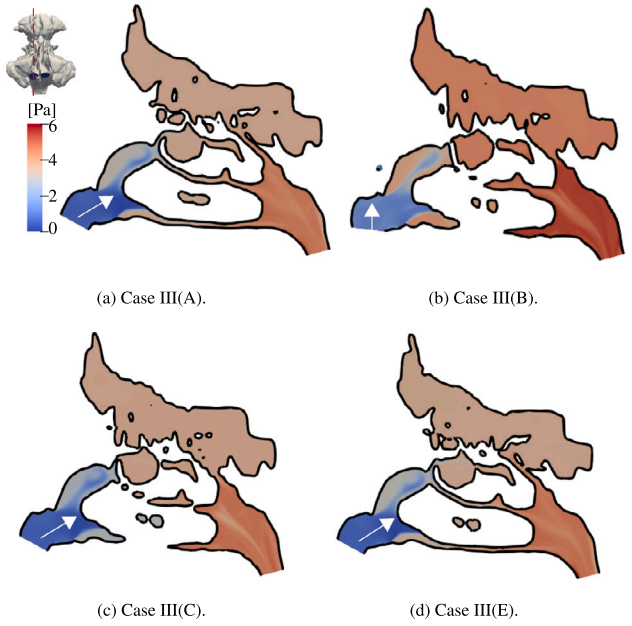
(f) Patient III, right nasal passage.

Fig. 9. Total temperature increase averaged over cross-sectional areas along centerlines through the left (red) and right (violet) nasal passages. The black lines represent simulation results based on the CT data with high resolution (A), green lines simulation results based on CT data with low resolution (B), blue lines simulations results based on interpolated CT data (C), orange lines simulation results based on the CT data generated by the SRN (D), and magenta lines stand for improved results from the SRN by employing the data efficient method (E). The start and end points in the nasal cavity are illustrated for each test patient by the dashed black lines.

II(A) at the beginning of the left and right centerlines in Figs. 8(c) and 8(d), but is then characterized by strong fluctuations and overpredicts the pressure loss further downstream towards the end of the centerlines. Although both cases II(D) and II(E) are in good agreement with case II(A), case II(E) predicts the pressure loss near the outlet better than case II(D). Fig. 8(e) shows that the cases III(C)–III(E) have no relevant deviations of the pressure loss from case III(A) along the left centerline. However, Fig. 8(f) demonstrates a large underprediction of the pressure loss in case III(C) in the first half of the right centerline compared to case III(A), whereas the cases III(D) and III(E) match well with case

Table 2
The Reynolds number at the pharynx Re_P, the number of cells N_c, the total pressure loss between the inlets and outlet Δp_{tot}, and the temperature increase between the inlets and outlet ΔT are given for simulations of each patient I, II, and III, based on the CT data with high resolution (A), the CT data with low resolution (B), interpolation (C), the SRN (D), and the data efficient method (E). The quantities in brackets provide the deviations of cases (B), (C), (D), and (E) to case (A). Furthermore, the average absolute deviation is provided for cases (B), (C), (D), and (E) compared to case (A).

Cases	Re_P	N_c [$\times 10^6$]	Δp_{tot} [Pa]	ΔT [K]
I(A)	1580.7	114	9.116	11.660
I(B)	1526.1(−3.5%)	116(+1.8%)	8.596(−6.7%)	11.431(−1.5%)
I(C)	1548.7(−2.0%)	116(+1.8%)	7.257(−20.4%)	11.646(−0.1%)
I(D)	1562.8(−1.1%)	114(0.0%)	8.677(−4.8%)	11.637(−0.2%)
I(E)	1567.2(−0.9%)	114(0.0%)	8.827(−3.2%)	11.683(+0.7%)
II(A)	856.3	187	3.725	14.460
II(B)	840.1(−1.9%)	200(+7.0%)	3.398(−8.8%)	14.562(+0.7%)
II(C)	821.9(−4.0%)	196(+4.8%)	3.934(+5.6%)	14.650(+1.3%)
II(D)	842.6(−1.6%)	198(−1.0%)	3.820(+2.6%)	14.457(−0.1%)
II(E)	849.1(−0.8%)	199(−0.5%)	3.746(+0.6%)	14.475(+0.1%)
III(A)	1264.7	246	5.292	14.600
III(B)	1220.7(−3.5%)	248(+1.8%)	5.860(+10.7%)	14.439(−1.1%)
III(C)	1255.4(−0.7%)	243(−1.2%)	5.267(−0.5%)	14.326(−1.9%)
III(D)	1262.3(−0.2%)	246(−0.8%)	5.298(+0.1%)	14.564(−0.2%)
III(E)	1262.7(−0.2%)	246(−0.8%)	5.279(−0.2%)	14.597(0.0%)
AAD(B)	3.0%	3.5%	8.7%	1.1%
AAD(C)	2.2%	2.6%	8.8%	1.1%
AAD(D)	1.0%	0.6%	2.5%	0.2%
AAD(E)	0.6%	0.4%	1.3%	0.3%



(a) Case III(A). (b) Case III(B). (c) Case III(C). (d) Case III(E).

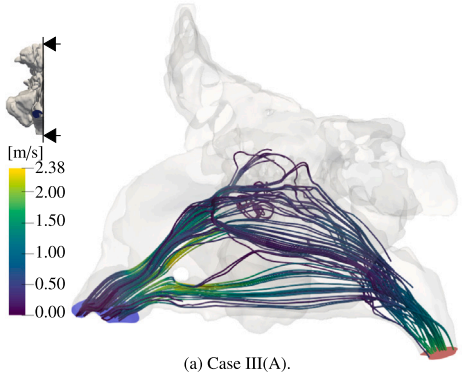
Fig. 10. Total pressure loss at the coronal cross sections represented by the dashed red line for the cases III(A), III(B), III(C), and III(E). The main inflow direction is expressed by the white arrows.

III(A). Figs. 8(e) and 8(f) display a continuously overpredicted pressure loss for case III(B) in both nasal passages.

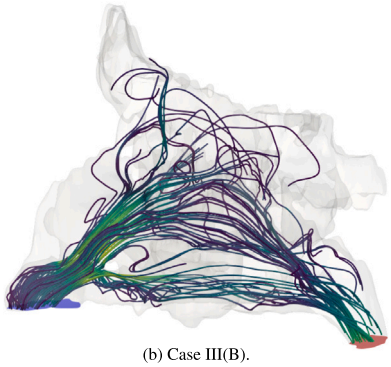
In general, the temperature distributions in Fig. 9 show less deviations from the reference case compared to the pressure loss in Fig. 8. Fig. 9(a) reveals that the temperature increase for cases I(B)–I(E) is in line with case I(A), except for fluctuations of case I(B) at the mid locations of the left centerline. Fig. 9(b) presents an overall good agreement between all cases along the right centerline, except for slight underpredictions at the mid locations for the cases I(B) and I(C). Fig. 9(c) indicates that cases II(B) and II(C) tend to underpredict the temperature increase in the first three quarters of the left centerline, the temperature increase is, however, captured well by the cases II(D) and II(E) at the same locations. All cases match well near the outlet. The results for the right centerline in Fig. 9(d) show that only the cases II(D) and II(E) come close to the temperature increase of case II(A). Cases II(B) and II(C) have the tendency to overpredict the temperature increase, except for the part of the centerline near the outlet. In case of patient III, Fig. 9(e) indicates that there is a good agreement between the temperature increase for all cases in the left nasal passage. However, it is from Fig. 9(f) evident that the temperature increase of case III(B) is underpredicted in the anterior part of the right nasal cavity, compared to case III(A). Case III(C) is characterized by few deviations from case III(A), e.g., shortly before the outflow region at the pharynx. The cases III(D) and III(E) show large fluctuations at the mid locations of the right centerline, but recover towards the outlet.

An example of a detailed view on the flow field is given in terms of a coronal cross section through the right nasal passage of patient III in Fig. 10. Since the flow fields of the cases III(D) and III(E) show only minor qualitative differences, only case III(E) is depicted. The flow fields of the cases III(A) in Fig. 10(a) and III(E) in Fig. 10(d) show no significant differences. The flow field of case III(B) in Fig. 10(b) provides insights about the reasons for the continuous overprediction of the pressure loss visible in Fig. 8(f). Not only the inferior turbinate is narrowed by an obstacle, as already shown in Fig. 7(c), but also the middle and superior turbinates. This can be inferred from the air not flowing continuously between the nostril and the pharynx through all turbinates, see Fig. 10(b). Furthermore, the nostril and, therefore, the main inflow direction, visualized by a white arrow, differ in case III(B) from the remaining cases. The diagonal main inflow direction in the remaining cases distributes incoming air to all turbinates. In contrast, incoming air in case III(B) is first led to the upper airway-nostril interface and then guided towards the narrowed turbinates horizontally. The combination of deflected incoming air and narrowed turbinates causes the overall high pressure loss. Fig. 10(c) indicates that there is less narrowing of the turbinates for case III(C), i.e., incoming air flows into the superior turbinate, which is hindered in case III(B). Additionally, the main inflow direction does not deviate from case III(A) and incoming air is also guided towards the superior and middle turbinates. The underprediction in the anterior part of the right nasal passage is induced by the cross sections along the right centerline cutting only through the superior turbinate with a relatively low pressure loss in this region, compared to a higher combined pressure loss in all three turbinates in the cases III(A) and III(D), and the much higher pressure loss in the superior turbinate in case III(B).

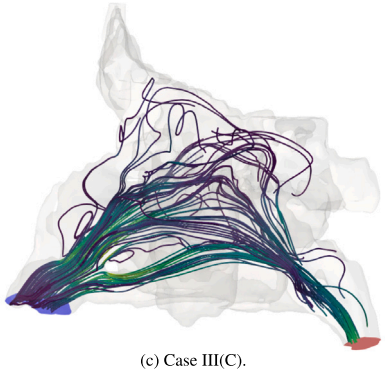
The previously described observations are additionally underlined by results shown in Fig. 11. Here, streamlines through the right nasal passage are depicted for the cases III(A), III(B), III(C), and III(E). The streamlines start at the right inlet (blue area), end at the outlet (red area), and are colored by the velocity magnitude $|\boldsymbol{u}|$. Again, case III(D) is excluded as it is only marginally different to case III(E). Similar to Fig. 10, the streamlines of the cases III(A) in Fig. 11(a) and III(E) in Fig. 11(d) show no significant differences. In both cases, the incoming air is distributed smoothly to all three turbinates. In contrast, the incoming air is deflected by the blocked channels and the streamlines get spread all over the nasal cavity and also distribute into parts of the ethmoidal sinuses in cases III(B) and III(C), see Figs. 11(b) and 11(c). In case III(B), the streamlines through the inferior turbinate are characterized by spinning motion which indicates a deflected flow at the narrowed passage. Furthermore, as a consequence of the incorrect prediction of the inflow region, the incoming streamlines collide with the upper airway-nostril interface before they find their way towards the turbinates. In case III(C), the streamlines are deflected to the outer part of the turbinates which are located higher in the z-direction than the blocked part.



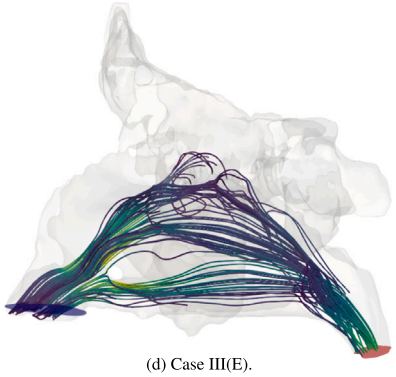
(a) Case III(A).



(b) Case III(B).



(c) Case III(C).



(d) Case III(E).

Fig. 11. Streamlines through the right nasal passage for the cases III(A), III(B), III(C), and III(E), colored by the velocity magnitude $|\boldsymbol{u}|$.

4. Summary and discussion

In this study, an SRN pipeline is developed to increase the through-plane resolution of CT data from 3.0 mm to 1.0 mm. The CT data are used to reconstruct 3D models of the nasal cavity, which define the spatial domain for numerical simulations of respiratory flows. A sufficient through-plane resolution is required for reliable CFD simulations. First, the hyperparameters of the SRN are determined with an automated HPO framework. Second, a data efficient method [45] has been integrated into the SRN pipeline in combination with image tiling, improving the convergence speed and model reliability, without degrading its predictive performance. Finally, it has been analyzed how simulation results of flow through 3D models reconstructed from the SRN-generated data compete with simulation results based on CT data of a reference case with a 1.0 mm resolution, a case with a coarse 3.0 mm resolution, and a case with data that have been interpolated from 3.0 mm to 1.0 mm.

It could be shown that geometric parameters of the 3D models generated based on the CT data from the optimized SRN have the smallest deviations from the reference case. Furthermore, simulations based on the CT data generated by the optimized SRN have only negligible deviations from the reference case in terms of the physical quantities pressure loss and temperature increase between the inlets (nostrils) and outlet (pharynx). In contrast, simulations based on coarse or interpolated data are characterized by large deviations. A further analysis of the pressure and temperature distributions along centerlines through the left and right nasal passages confirmed that simulations based on SRN-generated CT data outperform simulations based on coarse or interpolated CT data. These findings are consistent with the findings in [30,31], who state that an increased slice thickness leads to decreased geometric accuracy, especially in regions with large changes in surface geometry along the scan axis. The human nasal cavity is a complex and intricate geometry, and, therefore, 3D models reconstructed from coarse data exhibit low anatomical fidelity and lead to large deviations in fluid mechanical properties from the ground truth data. However, they are recovered successfully using an SRN. Hence, if the original data have a resolution coarser than 1.0 mm, the SRN is a valid tool to increase the through-plane resolution along the z-axis and guarantee reliable respiratory flow simulations.

It should be noted that training the SRN is more expensive than interpolation. However, once the SRN is trained and saved, inference on new, unseen data is extremely fast while at the same time the reconstructions are more accurate and hence also the simulation outputs, as compared to the interpolation case. When juxtaposing the end-to-end solution between the two approaches, which both only take maximum a few seconds on desktop hardware, the difference in time-to-solution is not of importance.

Another important aspect of this study is to build and optimize an automated pipeline. With the advancement of AutoML methods and computational power, HPO methods can match or even outperform human experts [27], making them a great solution to assist medical professionals that utilize existing ML methods with limited expertise in the field. A practical HPO method should possess several features: flexibility, scalability, and efficiency [26]. Bayesian optimization with asynchronized successive halving is employed due to it's high efficiency in using a surrogate model and parallel resources. Two types of hyperparameters are searched in a network, one to control the learning algorithm and the other one is related to the architecture. The search space is defined to include both algorithm hyperparameters, i.e., the optimizer and learning rate, and hyperparameters related to the architecture such as the number of feature channels. The HPO results reported in Section 3.1 suggest to use more feature channels, which is consistent with the findings in [21], who state that more feature channels improve performance.

The data efficient method described in Section 2.4 allows the model to be focused on relevant image tiles by over-sampling them and avoids wasting time on easier image tiles by down-sampling them, which speeds up the learning task. The results described in Section 3.2 suggest that the method does not only achieve a similar SR quality compared to the baseline network, the method achieves the results also much faster than the baseline case. Furthermore, the biasing method leads to more stable results, i.e., a smaller standard deviation of the MSEs per epoch.

This is important for CFD-based medical applications because of a high demand for reliability.

The current study focuses on using a classic network architecture with multiple optimization steps. It demonstrates the benefit of incorporating an SRN into an automated respiratory flow simulation pipeline. In medical SRNs, it is critical to maintain anatomical fidelity of fine-grain structures, e.g., to be able to do accurate surgery planning. In a follow-up study, the integration of shape priors of the nasal cavity into the SRN will be investigated. This way, the learning process might be able to better capture the underlying anatomy due to its highly constrained nature [59,60]. In addition, other state-of-the-art HPO techniques and specific implementations will be investigated and compared for a better understanding of their performance and efficiency. Evolutionary algorithms have proven to produce good results and are robust against noise and local minima. Although they comparably require larger resources [61], they can handle well complex settings of search space where other optimizers may fail [62]. Therefore, a larger and more complex search space will be explored by evolutionary algorithms or other population-based methods.

CRediT authorship contribution statement

Xin Liu: Investigation, Conceptualization. **Mario Rüttgers:** Methodology, Formal analysis, Conceptualization. **Alessio Quercia:** Methodology. **Romain Egele:** Supervision, Methodology. **Elisabeth Pfaehler:** Methodology. **Rushikesh Shende:** Formal analysis. **Marcel Aach:** Methodology. **Wolfgang Schröder:** Supervision. **Prasanna Balaprakash:** Supervision. **Andreas Lintermann:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors intent to make the super-resolution tool open access via a web application that can be used by any patient.

Acknowledgments

The research leading to these results has been conducted in the *Joint Laboratory for Extreme Scale Computing* (JLESC) project: Architecture and Hyperparameter Search for Super-Resolution Networks Operating on Medical Images. Furthermore, the research has been performed in the CoE RAISE project, which receives funding from the European Union's *Horizon 2020 – Research and Innovation Framework Programme* H2020-INFRAEDI-2019-1 under grant agreement no. 951733. The authors gratefully acknowledge the computing time granted by the JARA Vergabegremium and provided on the JARA Partition part of the supercomputer JURECA [56] at *Forschungszentrum Jülich*. This work was performed as part of the *Helmholtz School for Data Science in Life, Earth and Energy* (HDS-LEE).

References

[1] Andreas Lintermann, Matthias Meinke, Wolfgang Schröder, Fluid mechanics based classification of the respiratory efficiency of several nasal cavities, Comput. Biol. Med. 43 (11) (2013) 1833–1852.

[2] Mario Rüttgers, Moritz Waldmann, Wolfgang Schröder, Andreas Lintermann, A machine-learning-based method for automatizing Lattice-Boltzmann simulations of respiratory flows, Appl. Intell. 52 (8) (2022) 9080–9100.

[3] Goldman W. Lee, Principles of CT: radiation dose and image quality, J. Nucl. Med. Technol. 35 (4) (2007) 213–225.

[4] Stephen P Power, Fiachra Moloney, Maria Twomey, Karl James, Owen J O'Connor, Michael M Maher, Computed tomography and patient risk: Facts, perceptions and uncertainties, World J. Radiol. 8 (12) (2016) 902.

[5] S.P. Kim, Nirmal K. Bose, Reconstruction of 2-D bandlimited discrete signals from nonuniform samples, in: IEE Proceedings F (Radar and Signal Processing), Vol. 137, IET, 1990, pp. 197–204.

[6] Raymond Y. Tsai, Multiple frame image restoration and registration, Adv. Comput. vis. Image Process 1 (1989) 1715–1989.

[7] Nirmal K. Bose, Surapong Lertrattanapanich, Jaehoon Koo, Advances in superresolution using L-curve, in: ISCAS 2001. the 2001 IEEE International Symposium on Circuits and Systems, Cat. No. 01CH37196, Vol. 2, IEEE, 2001, pp. 433–436.

[8] Sung Cheol Park, Min Kyu Park, Moon Gi Kang, Super-resolution image reconstruction: a technical overview, IEEE Signal Process. Mag. 20 (3) (2003) 21–36.

[9] Chao Dong, Chen Change Loy, Kaiming He, Xiaoou Tang, Learning a deep convolutional network for image super-resolution, in: David Fleet, Tomas Pajdla, Bernt Schiele, Tinne Tuytelaars (Eds.), Computer Vision – ECCV 2014, Springer International Publishing, Cham, 2014, pp. 184–199.

[10] Chao Dong, Chen Change Loy, Kaiming He, Xiaoou Tang, Image super-resolution using deep convolutional networks, IEEE Trans. Pattern Anal. Mach. Intell. 38 (2) (2016) 295–307.

[11] Eirikur Agustsson, Radu Timofte, NTIRE 2017 challenge on single image super-resolution: Dataset and study, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, 2017, pp. 1122–1131.

[12] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi, Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network, 2017, pp. 105–114.

[13] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, Chen Change Loy, ESRGAN: Enhanced super-resolution generative adversarial networks, in: Laura Leal-Taixé, Stefan Roth (Eds.), Computer Vision – ECCV 2018 Workshops, Springer International Publishing, Cham, 2019, pp. 63–79.

[14] Zhisheng Lu, Juncheng Li, Hong Liu, Chaoyan Huang, Linlin Zhang, Tieyong Zeng, Transformer for single image super-resolution, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, IEEE Computer Society, Los Alamitos, CA, USA, 2022, pp. 456–465.

[15] Olaf Ronneberger, Philipp Fischer, Thomas Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, Springer, 2015, pp. 234–241.

[16] YiNan Zhang, MingQiang An, Deep learning-and transfer learning-based super resolution reconstruction from single medical image, J. Healthc. Eng. 2017 (2017).

[17] Zhangpei Cheng, Kaixuan Guo, Changfeng Wu, Jiankun Shen, Lei Qu, U-net cascaded with dilated convolution for medical image registration, in: 2019 Chinese Automation Congress, CAC, IEEE, 2019, pp. 3647–3651.

[18] Junyoung Park, Donghwi Hwang, Kyeong Yun Kim, Seung Kwan Kang, Yu Kyeong Kim, Jae Sung Lee, Computed tomography super-resolution using deep convolutional neural network, Phys. Med. Biol. 63 (14) (2018) 145011.

[19] Guang Li, Lu Bai, Chuanwei Zhu, Enhe Wu, Ruibing Ma, A novel method of synthetic CT generation from MR images based on convolutional neural networks, in: 2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, CISP-BMEI, IEEE, 2018, pp. 1–5.

[20] Yushi Liu, Brian Tracey, Shuchin Aeron, Eric Miller, Tao Sun, Nathan McDannold, James Murphy, Artifact suppression for passive cavitation imaging using U-net CNNs with uncertainty quantification, in: 2019 IEEE 4th International Conference on Signal and Image Processing, ICSIP, IEEE, 2019, pp. 1037–1042.

[21] Nahian Siddique, Sidike Paheding, Colin P Elkin, Vijay Devabhaktuni, U-net and its variants for medical image segmentation: A review of theory and applications, IEEE Access 9 (2021) 82031–82057.

[22] Ziang Zhang, Chengdong Wu, Sonya Coleman, Dermot Kerr, DENSE-INception U-net for medical image segmentation, Comput. Methods Programs Biomed. 192 (2020) 105395.

[23] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al., Attention u-net: Learning where to look for the pancreas, 2018, arXiv preprint arXiv:1804.03999.

[24] Zhengxin Zhang, Qingjie Liu, Yunhong Wang, Road extraction by deep residual U-net, IEEE Geosci. Remote Sens. Lett. 15 (5) (2018) 749–753.

[25] Marcel Aach, Rocco Sedona, Andreas Lintermann, Gabriele Cavallaro, Helmut Neukirchen, Morris Riedel, Accelerating hyperparameter tuning of a deep learning model for remote sensing image classification, in: IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium, IEEE, 2022, pp. 263–266.

[26] James Bergstra, Yoshua Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2) (2012).

[27] Matthias Feurer, Frank Hutter, Hyperparameter optimization, Autom. Mach. Learn.: Methods Syst. Chall. (2019) 3–33.

[28] Katharina Eggensperger, Philipp Müller, Neeratyoy Mallik, Matthias Feurer, René Sass, Aaron Klein, Noor Awad, Marius Lindauer, Frank Hutter, HPOBench: A collection of reproducible multi-fidelity benchmark problems for HPO, 2021, arXiv preprint arXiv:2109.06716.

[29] Andreas Lintermann, Matthias Meinke, Wolfgang Schröder, Zonal flow solver (ZFS): a highly efficient multi-physics simulation framework, Int. J. Comput. Fluid Dyn. 34 (7–8) (2020) 458–485.

[30] Beat Schmutz, Martin E. Wullschleger, Michael A. Schuetz, The effect of CT slice spacing on the geometry of 3D models, in: 6th Australasian Biomechanics Conference, 2007, pp. 93–94.

[31] Jonathan M. Ford, Summer J. Decker, Computed tomography slice thickness and its effects on three-dimensional reconstruction of anatomical structures, J. Forensic Radiol. Imaging 4 (2016) 43–46.

[32] Sergey Ioffe, Christian Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, PMLR, 2015, pp. 448–456.

[33] Andrew L. Maas, Awni Y. Hannun, Andrew Y. Ng, et al., Rectifier nonlinearities improve neural network acoustic models, in: Proc. Icml, Vol. 30, Atlanta, Georgia, USA, 2013, p. 3.

[34] Xavier Glorot, Yoshua Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, Soumith Chintala, PyTorch: An imperative style, high-performance deep learning library, in: Proceedings of the 33rd International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA, 2019.

[36] Frank Hutter, Holger H. Hoos, Kevin Leyton-Brown, Sequential model-based optimization for general algorithm configuration, in: Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome, Italy, January 17-21, 2011. Selected Papers 5, Springer, 2011, pp. 507–523.

[37] Tobias Domhan, Jost Tobias Springenberg, Frank Hutter, Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves, in: Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015.

[38] Zohar Karnin, Tomer Koren, Oren Somekh, Almost optimal exploration in multi-armed bandits, in: International Conference on Machine Learning, PMLR, 2013, pp. 1238–1246.

[39] Kevin Jamieson, Ameet Talwalkar, Non-stochastic best arm identification and hyperparameter optimization, in: Artificial Intelligence and Statistics, PMLR, 2016, pp. 240–248.

[40] Liam Li, Kevin Jamieson, Afshin Rostamizadeh, Ekaterina Gonina, Jonathan Ben-Tzur, Moritz Hardt, Benjamin Recht, Ameet Talwalkar, A system for massively parallel hyperparameter tuning, Proc. Mach. Learn. Syst. 2 (2020) 230–246.

[41] Prasanna Balaprakash, Michael Salim, Thomas D Uram, Venkat Vishwanath, Stefan M Wild, DeepHyper: Asynchronous hyperparameter search for deep neural networks, in: 2018 IEEE 25th International Conference on High Performance Computing, HiPC, IEEE, 2018, pp. 42–51.

[42] Diederik P. Kingma, Jimmy Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[43] Herbert Robbins, Sutton Monro, A stochastic approximation method, Ann. Math. Stat. (1951) 400–407.

[44] Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I Jordan, et al., Ray: A distributed framework for emerging {AI} applications, in: 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18), 2018, pp. 561–577.

[45] Alessio Quercia, Abigail Morrison, Hanno Scharr, Ira Assent, SGD biased towards early important samples for efficient training, in: 2023 IEEE International Conference on Data Mining, ICDM, IEEE, 2023.

[46] A. Lintermann, S. Schlimpert, J.H. Grimmen, C. Günther, M. Meinke, W. Schröder, Massively parallel grid generation on HPC systems, Comput. Methods Appl. Mech. Engrg. 277 (2014) 131–153.

[47] Daniel Hartmann, Matthias Meinke, Wolfgang Schröder, An adaptive multilevel multigrid formulation for cartesian hierarchical grid methods, Comput. & Fluids 37 (9) (2008) 1103–1125.

[48] Hans Sagan, Hilbert's space-filling curve, in: Space-Filling Curves, Springer New York, New York, NY, 1994, pp. 9–30.

[49] G.M. Morton, A Computer Oriented Geodetic Data Base; and a New Technique in File Sequencing, Technical report, IBM Ltd., Armonk, NY, 1966.

[50] Jianwei Li, Wei keng Liao, A. Choudhary, R. Ross, R. Thakur, W. Gropp, R. Latham, A. Siegel, B. Gallagher, M. Zingale, Parallel netCDF: A high-performance scientific I/O interface, in: SC '03: Proceedings of the 2003 ACM/IEEE Conference on Supercomputing, 2003, pp. 39–39.

[51] Xiaoyi He, Li-Shi Luo, Theory of the Lattice Boltzmann method: From the Boltzmann equation to the Lattice Boltzmann equation, Phys. Rev. E 56 (6) (1997) 6811–6817.

[52] Yue-Hong Qian, Dominique D'Humières, Pierre Lallemand, Lattice BGK models for Navier-Stokes equation, Europhys. Lett. 17 (6) (1992) 479–484.

[53] Zhaoli Guo, Chuguang Zheng, Baochang Shi, T.S. Zhao, Thermal Lattice Boltzmann equation for low mach number flows: Decoupling model, Phys. Rev. E 75 (2007) 036704.

[54] M'hamed Bouzidi, Mouaouia Firdaouss, Pierre Lallemand, Momentum transfer of a Boltzmann-Lattice fluid with boundaries, Phys. Fluids 13 (11) (2001) 3452–3459.

[55] Georg Eitel, Rainhill K. Freitas, Andreas Lintermann, Matthias Meinke, Wolfgang Schröder, Numerical simulation of nasal cavity flow based on a Lattice-Boltzmann method, in: Andreas Dillmann, Gerd Heller, Michael Klaas, Hans-Peter Kreplin, Wolfgang Nitsche, Wolfgang Schröder (Eds.), New Results in Numerical and Experimental Fluid Mechanics VII, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 513–520.

[56] Dorian Krause, Philipp Thörnig, JURECA: Modular supercomputer at Jülich supercomputing centre, J. Large-Scale Res. Facil. 4 (2018) A132.

[57] Yufei Li, Bruno Sixou, Francois Peyrin, A review of the deep learning methods for medical images super resolution problems, Irbm 42 (2) (2021) 120–133.

[58] Luca Antiga, Marina Piccinelli, Lorenzo Botti, Bogdan Ene-Iordache, Andrea Remuzzi, David A. Steinman, An image-based modeling framework for patient-specific computational hemodynamics, Med. Biol. Eng. Comput. 46 (2008) 1097–1112.

[59] Xiaozheng Xie, Jianwei Niu, Xuefeng Liu, Zhengsu Chen, Shaojie Tang, Shui Yu, A survey on incorporating domain knowledge into deep learning for medical image analysis, Med. Image Anal. 69 (2021) 101985.

[60] Ozan Oktay, Enzo Ferrante, Konstantinos Kamnitsas, Mattias Heinrich, Wenjia Bai, Jose Caballero, Stuart A Cook, Antonio De Marvao, Timothy Dawes, Declan P O'Regan, et al., Anatomically constrained neural networks (ACNNs): application to cardiac image enhancement and segmentation, IEEE Trans. Med. Imaging 37 (2) (2017) 384–395.

[61] Bernd Bischl, Martin Binder, Michel Lang, Tobias Pielok, Jakob Richter, Stefan Coors, Janek Thomas, Theresa Ullmann, Marc Becker, Anne-Laure Boulesteix, et al., Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges, Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 13 (2) (2023) e1484.

[62] Xin He, Kaiyong Zhao, Xiaowen Chu, AutoML: A survey of the state-of-the-art, Knowl.-Based Syst. 212 (2021) 106622.

Xin Liu works as a postdoctoral researcher at the Simulation and Data Laboratory "Highly Scalable Fluid & Solids Engineering" of the Jülich Supercomputing Centre (JSC), Forschungszentrum Jülich. Her research interests include high-performance computing, artificial intelligence and its applications in medical imaging.

Mario Rüttgers is working as a doctoral researcher at the Simulation and Data Laboratory "Highly Scalable Fluid & Solids Engineering" (SDL FSE) of the Jülich Supercomputing Centre (JSC), Forschungszentrum Jülich, and at the Institute of Aerodynamics (AIA), RWTH Aachen University. He is associated with the Helmholtz School for Data Science in Life, Earth and Energy (HDSLEE). His research focuses on combining computational fluid dynamics and machine learning techniques to improve diagnoses and treatments of respiratory diseases.

Alessio Quercia received his bachelor's and master's degrees in Computer Science from Sapienza University of Rome in 2017 and University of Milan in 2020. He was a Research Intern at IBM Research Zurich from 2019 to 2020 and a Computer Science Research Fellow at Sapienza University of Rome from 2020 to 2021. Currently, he is a Ph.D. student at the Data Analytics and Machine Learning (IAS-8) Insitute, Forschungszentrum Jülich, and at the Computer Science Department, RWTH Aachen University. He is associated with the Helmholtz School for Data Science in Life, Earth and Energy (HDS-LEE). His research focus is mainly on Data Efficiency and Transfer Learning.

Romain Egele is a Ph.D. student with a joint appointment at Argonne National Laboratory (USA) and Université Paris-Saclay (France). His research focuses on the optimization of learning workflows at large scale on HPC platforms. His work spans hyperparameter and neural architecture search as well as uncertainty quantification in machine learning. He is the lead developer of the DeepHyper Python package which provides parallel hyperparameter optimization algorithms for HPC.

Elisabeth Pfaehler is currently working as postdoctoral researcher at the Data Analytics and Machine Learning (IAS-8) Insitute, Forschungszentrum Jülich. Her research focus on the application and reproducibility of machine learning for medical images.

Rushikesh Shende received his bachelor's in Mechanical Engineering from College of Engineering Pune in 2019 and his master's in Computational Mechanics from RWTH Aachen University, Germany in 2023. Currently, he is working as a doctoral researcher at the Institute of Aerodynamics (AIA), RWTH Aachen University. His research focuses on aeroacoustic simulations of aircrafts with a goal to avoid generation of condensation trails during high altitude flights.

Marcel Aach is a doctoral researcher at the "Highly Scalable Fluids & Solids Engineering" Simulation and Data Laboratory at the Jülich Supercomputing Centre and the University of Iceland. His research focuses on accelerating machine learning and hyperparameter optimization methods on HighPerformance Computing systems.

Wolfgang Schröder holds the Chair of Fluid Mechanics and heads the AIA, RWTH Aachen University. He is the Editor-in-Chief of the European Journal of Mechanics/B Fluids and the series Notes on Numerical Fluid Mechanics and Multidisciplinary Design. His main research is in the fields of computational fluid dynamics and aeroacoustics, measurement techniques, turbulence, single and multiphase flows, and biomedical flows.

Prasanna Balaprakash is the Director of AI Programs and Distinguished R&D Scientist at Oak Ridge National Laboratory, where he directs laboratory research, development and application of artificial intelligence and machine learning (AI/ML) to solve problems of national importance. His research interests span artificial intelligence, machine learning, optimization, and highperformance computing. He is a recipient of the U.S. Department of Energy's 2018 Early Career Award. Prior to joining Oak Ridge, He was a R&D lead and computer scientist at Argonne National Laboratory. He earned his Ph.D. from CoDE-IRIDIA at the Université Libre de Bruxelles in Brussels, Belgium, where he was a recipient of the European Commission's Marie Curie and Belgian F.R.S-FNRS Aspirant fellowships.

Andreas Lintermann is a postdoctoral researcher and group leader of the SDL FSE at JSC, Forschungszentrum Jülich. He is coordinating the European Center of Excellence in Exascale Computing "Research on AI- and SimulationBased Engineering at Exascale" (CoE RAISE), leads the activities in the EuroCC/EuroCC2 and interTwin projects from Jülich's side, and is involved in the Industry Relations Team of the institute. His research focuses on highperformance computing, heterogeneous hardware, modular supercomputing, artificial intelligence, bio-fluidmechanical analyses of respiratory diseases, latticeBoltzmann methods, high-scaling meshing methods, and efficient multi-physics coupling strategies.