Received: 4 June 2022 | Revised: 14 October 2022 | Accepted: 13 November 2022

DOI: 10.1002/aic.17971

AIChE JOURNAL

RESEARCH ARTICLE

Process Systems Engineering

Graph machine learning for design of high-octane fuels

Jan G. Rittig[1] | Martin Ritzert[2] | Artur M. Schweidtmann[3] | Stefanie Winkler[4] | Jana M. Weber[5] | Philipp Morsch[6] | Karl Alexander Heufer[6] | Martin Grohe[4] | Alexander Mitsos[1,7,8] | Manuel Dahmen[8]

[1]Process Systems Engineering (AVT.SVT), RWTH Aachen University, Aachen, Germany

[2]Department of Computer Science, Aarhus University, Aarhus, Denmark

[3]Department of Chemical Engineering, Delft University of Technology, Delft, The Netherlands

[4]Chair of Computer Science 7, RWTH Aachen University, Aachen, Germany

[5]Delft Bioinformatics Lab, Intelligent Systems, Delft University of Technology, Delft, The Netherlands

[6]Chair of High Pressure Gas Dynamics, RWTH Aachen University, Aachen, Germany

[7]JARA-ENERGY, Aachen, Germany

[8]Forschungszentrum Jülich GmbH, Institute for Energy and Climate Research IEK-10: Energy Systems Engineering, Jülich, Germany

Correspondence
Manuel Dahmen, Forschungszentrum Jülich GmbH, Institute for Energy and Climate Research IEK-10: Energy Systems Engineering, Jülich 52425, Germany.
Email: m.dahmen@fz-juelich.de

Funding information
Deutsche Forschungsgemeinschaft, Grant/Award Numbers: 466417970, EXC 2186, GRK 2236; Helmholtz-Gemeinschaft, Grant/Award Number: HDS-LEE (HIDSS-0004)

Abstract

Fuels with high-knock resistance enable modern spark-ignition engines to achieve high efficiency and thus low CO_2 emissions. Identification of molecules with desired autoignition properties indicated by a high research octane number and a high octane sensitivity is therefore of great practical relevance and can be supported by computer-aided molecular design (CAMD). Recent developments in the field of graph machine learning (graph-ML) provide novel, promising tools for CAMD. We propose a modular graph-ML CAMD framework that integrates generative graph-ML models with graph neural networks and optimization, enabling the design of molecules with desired ignition properties in a continuous molecular space. In particular, we explore the potential of Bayesian optimization and genetic algorithms in combination with generative graph-ML models. The graph-ML CAMD framework successfully identifies well-established high-octane components. It also suggests new candidates, one of which we experimentally investigate and use to illustrate the need for further auto-ignition training data.

KEYWORDS
computer-aided molecular design, fuel design, graph machine learning, graph neural networks, machine learning, optimization, renewable fuels, spark-ignition engines

1 | INTRODUCTION

With a share of 23% of total CO_2 emissions, transportation is a major CO_2 emission source.[1] Replacing fossil fuels with renewable alternatives may provide a path toward carbon neutrality for the transportation sector and is investigated actively.[2–5] An important step toward renewable fuels is the search for suitable gasoline substitutes for use in advanced high compression, turbocharged *spark-ignition* (SI) engines. A property of paramount importance for a renewable SI engine fuel is knock resistance, traditionally indicated by the

Jan G. Rittig and Martin Ritzert contributed equally to this study.

research octane number (RON),[6] the *motor octane number* (MON),[7] and more recently the *octane sensitivity* (OS), that is, the difference between RON and MON values. The weighted sum of RON and OS is referred to as the *octane index* (OI).[8] For modern SI engines, fuels with both high RON and high OS, hence high OI, are desired as they enable engine operation at conditions associated with particularly high efficiency.[9–15] To boost the OI of a fuel, chemical species with high RON and high OS such as ethanol and MTBE can be added.[16,17] Identification of further molecules providing octane boosting is of great practical relevance and is studied actively, for example, see references 17,18. Herein, we aim to identify such promising candidates exhibiting both high RON and high OS by *computer-aided molecular design* (CAMD). In particular, we investigate the role of novel methods from the domain of *graph machine learning* (graph-ML).

Traditionally, the search for molecules with desired properties for a given application has been mostly guided by human experts and experimentation. CAMD can enhance this process by utilizing computational methods to efficiently pre-screen a large number of molecular structures so that experiments can be dedicated to the most promising candidates. A wide variety of methods and tools for CAMD has been proposed over the last decades; we refer the interested reader to review articles for a detailed CAMD overview.[19–25] Generally, the CAMD process incorporates the computational generation of candidate structures and the model-based prediction of their physicochemical properties. Well-established approaches for the generation of candidate structures include formulating optimization problems in which structural groups are pieced together to form molecules,[24,26] exhaustive generation of molecular structures in a sequential generate-and-test manner,[27] and utilizing evolutionary theory to evolve molecular structures.[28] For predicting application-relevant properties of the formed candidate structures, CAMD typically employs quantitative structure–property relationships (QSPRs).[29] QSPRs first describe the molecular structure by so-called molecular descriptors, for example, atom counts, and secondly map those descriptors to a property of interest by linear or nonlinear models. Today, nonlinear ML models such as feedforward neural networks or random forests are often utilized in this regression step.[30–32]

For classical CAMD, a broad range of applications[25] can be found in the process systems engineering (PSE) literature, covering the design of single molecules (e.g., ionic liquids,[33] polymers[22]), the design of mixtures,[34–36] as well as integrated product and process design.[37,38] Classical CAMD techniques have also been applied extensively in the context of fuel design.[2,39–43] For example, in two previous articles,[40,44] we used enumeration-based generation of oxygenated hydrocarbons and subsequently screened the obtained molecules via QSPR models with respect to engine-relevant properties. We previously also developed a generate-and-test approach where molecular candidates are generated by iteratively refunctionalizing bioderived intermediates based on pre-defined transformation rules.[2] Also, Cai et al.[45] proposed a gasoline design model that employs rule-based transformation of molecules in combination with QSPR for property prediction to identify molecules with desired fuel properties such as high RON.

ML has recently been utilized for molecular structure generation by means of generative ML models, leading to novel, fully *ML-based CAMD* approaches.[25,46] In generative ML for molecules, two main directions can be distinguished: String-based approaches, for example, based on SMILES strings,[47] and graph-based approaches, the latter directly working on the molecular graph. For both directions, a range of models has been developed such as recurrent neural networks (RNNs), variational or adversarial autoencoders (VAEs/AAEs), generative adversarial networks (GANs), and reinforcement learning (RL).[46,48] The goal of such generative ML techniques is the unsupervised learning from a data set of molecular structures to generate new, chemically feasible structures that were not seen during training, thereby designing molecules. Specifically, generative ML models typically learn to encode molecules into a continuous space, the so-called latent space, and then decode samples from the latent space back to molecular structures. The continuous latent space is assumed to capture chemical information about molecules and embed molecules with similar structure or even similar properties close to each other.[49] Depending on the model architecture, ML-based CAMD typically relies either on strategic sampling of molecules from the latent space of the generative model using optimization strategies, for example, with VAEs,[50–52] or on direct generation of molecules with desired properties, for example, by GANs[53,54] or RL.[55,56] In contrast to classical CAMD, generative models in ML-based CAMD replace discrete molecule representations such as combinations of structural groups, molecular graphs, or SMILES strings with a continuous representation, thus enabling the use of continuous optimization approaches for molecular design.[57]

ML has also recently enabled end-to-end learning of physicochemical properties from molecular structure by means of graph neural networks (GNNs).[58–60] GNNs are graph-ML architectures that directly operate on the underlying graph structure of a molecule and thus circumvent the need for selecting meaningful molecular descriptors, a step that is inherent to all QSPR/QSAR approaches. Instead, GNNs enable a data-driven end-to-end learning framework for molecular property prediction.

Up to now, fully ML-driven CAMD has mainly focused on drug design.[46,61–63] A particular reason might be the availability of large training data sets and the incorporation of multiple drug design targets such as logP and drug-likeness in benchmarking platforms such as MOSES[64] and GuacaMol.[65] Such ML-driven CAMD approaches often combine molecule generation and property prediction (e.g., VAEs[51,52]), and sometimes optimization (e.g., GANs[53,54] or RL[56]) in a single ML model which needs to be retrained once the design target property changes and typically requires large property data sets for training.

In contrast to drug design, PSE applications, in particular model-based fuel design, often take place in a data-scarce environment, making ML-based CAMD challenging. In fact, there is only one very recent study using generative ML for fuel design: Liu et al.[66] employed a string-based VAE to generate a large database of non-oxygenated hydrocarbons for subsequent screening of candidates with respect to fuel properties, followed by sampling further candidates from the most promising regions of the VAE's latent space. However, ML-driven CAMD has not

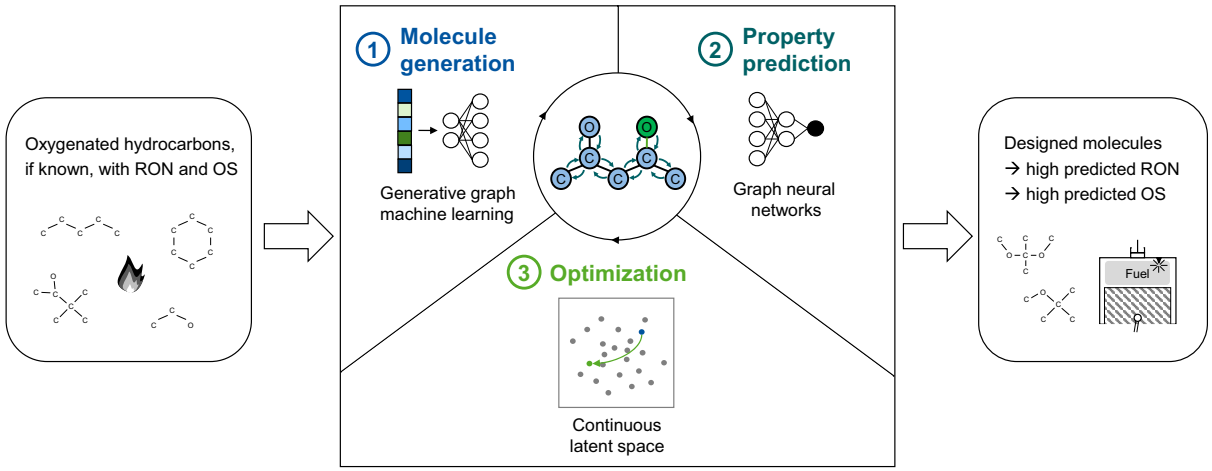


FIGURE 1 Schematic overview of the modular graph-ML CAMD framework for identification of high-octane fuels

yet been utilized for fuel design focusing on high SI engine efficiency including oxygenated hydrocarbons. Moreover, graph-ML approaches have not yet been applied to computer-aided fuel design.

In the present contribution, we propose a modular graph-ML CAMD framework* that integrates state-of-the-art graph-based ML methods and tools from the ML and drug design community and apply our framework to computer-aided design of high-octane fuel components for SI engines. Our framework is depicted in Figure 1 and consists of three distinct modules: (1) molecule generation by generative graph-ML models that learn a continuous molecular space from which new molecules can be generated; (2) property prediction through our recently published GNN model for fuel ignition quality prediction[68]; (3) optimization for strategic sampling from the continuous space of the generative graph-ML models to identify vectors that correspond to molecules with high predicted RON and OS values. Our framework has a modular architecture requiring minimal changes to the model structures if an additional property shall be targeted, that is, only a new property model needs to be trained and added, but the molecule generation and optimization modules do not need to be altered. Thus, the modular setup enhances reusability and therefore reduces the training effort compared to a single ML model approach, as indicated by Winter et al.[69]

We explore three different generative graph-ML models and two different optimization strategies. Importantly, we propose an applicability domain approach for GNN-based property prediction that allows us to focus the design process on molecules that presumably come with reliable predictions. We analyze the influence of the different ML methods on the structure and properties of the resulting molecules and compile a list of most promising high-octane fuel candidates. Finally, we perform an experimental investigation of one selected high-octane fuel candidate that emphasizes the importance of experimental validation of CAMD results and discuss potential pitfalls of the fully data-driven approach, particularly in a data-scarce environment.

The article is structured as follows: In Section 2, we briefly introduce the main principles behind graph-ML for molecules with regard to both molecule generation and property prediction. In Section 3, we present the modular graph-ML CAMD framework for design of high-octane fuels. The application of the framework in Section 4 includes a comparative analysis of the candidates obtained with different graph-ML modules and the experimental investigation of one particular candidate. Section 5 concludes our work.

2 | PRELIMINARIES OF GRAPH MACHINE LEARNING

Graph-ML relies on a graph representation of molecules that can be utilized for generating molecular structures from a continuous space and for property prediction, as we briefly describe in the following. The interested reader is referred to references 70–72 for further details on graph-ML.

2.1 | Molecular graph

The *molecular graph* of a molecule is an undirected graph $G_{mol} = \{V, F_v, E, F_e\}$; the nodes V represent the atoms; pairs of atoms u, $v \in V$ that share a bond are connected by edges $(u, v) \in E$. Additional features of nodes (e.g., type of atom, degree of hybridization) are stored in F_v, while additional features of edges (e.g., bond length or type) are stored in F_e.

2.2 | Generative models

Generative ML, the unsupervised learning from input data to generate new data that is similar to the provided data, allows to perform fully data-driven molecule generation and is an active research area.[46,48,63,73] Various works have developed string-based ML models in order to generate molecules with optimal properties based on SMILES,[74–81] InChI,[49] or SELFIES,[76] the latter being a more robust string representation of molecules. In contrast, graph-ML directly

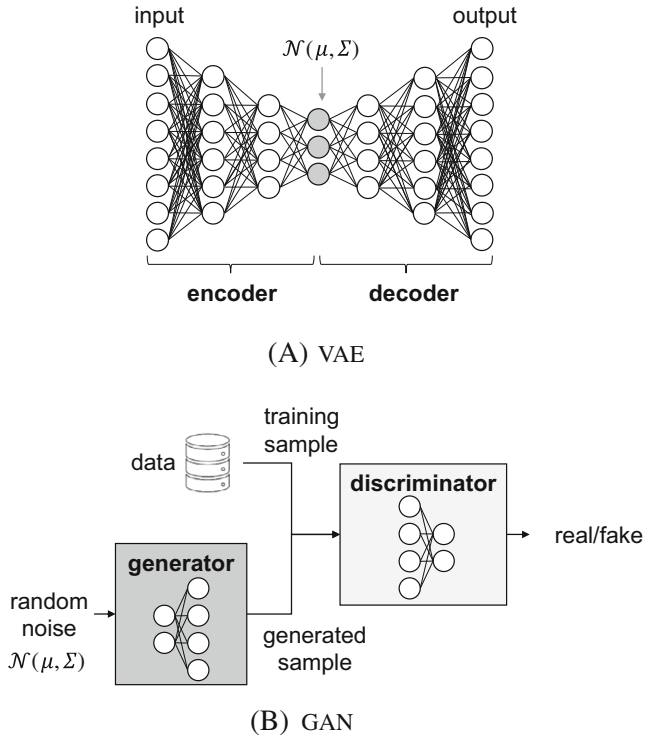


FIGURE 2 Schematic structure of (A) VAEs and (B) GANs

works on the molecular graph which is arguably the more natural representation of a molecule and provides permutation invariance,[82] that is, there is exactly one molecular graph for each molecule (neglecting steric effects). In this article, we focus on two frequently employed generative graph-ML approaches[46,48,63]: VAEs and GANs. Both methods construct a latent space where molecules are encoded as high-dimensional continuous vectors, referred to as latent vectors (LVs), which we denote as $\mathbf{h}_{LV} \in \mathbb{R}^n$ with the dimension n being a hyperparameter. We denote the encoding of a molecular graph into the latent space as a function

$$e_{GEN} : G_{mol} \mapsto \mathbf{h}_{LV}. \tag{1}$$

Autoencoders, and specifically VAEs, are a class of neural network architectures that employs an hourglass shape (cf. Figure 2A). They are trained to reproduce the input data at the output layer, a non-trivial task as the information has to be moved through some narrow layers in the middle of the network, that is, the hourglass shape forces VAEs to learn $\mathbf{h}_{LV}$ as a low-dimensional representation of the input data at the most narrow layer. The left part of the network (from input to the latent vector) is called the *encoder* and the right part (from the latent vector to the output) is referred to as the *decoder*. The main difference between a standard autoencoder and a variational autoencoder (VAE) is that the latter assumes an underlying distribution for the data that it tries to learn in the latent vector space, for example, a multivariate Gaussian distribution $\mathbf{h}_{LV} \sim \mathcal{N}(\mu, \Sigma)$ with parameters μ and Σ. VAEs can therefore be used to generate new data from presumably the same distribution as the input data. In the molecular context, VAEs map discrete molecule representations such as graphs to a continuous distribution from which new molecules can be sampled.

GANs generate objects from a latent representation in a different manner (cf. Figure 2B). Instead of trying to reproduce an input sample, a GAN consists of two neural networks, a *generator* and a *discriminator*, where the discriminator is trained to distinguish between output data produced by the generator and real data, that is, the training samples. The generator thus learns to produce output data that resembles a given training data based on random input vectors $\mathbf{h}_{LV}$ that are, for example, sampled from a Gaussian distribution, that is, $\mathbf{h}_{LV} \sim \mathcal{N}(\mu, \Sigma)$. In a GAN, the latent space therefore corresponds to the input space of the generator. We denote the decoding of the latent vector $\mathbf{h}_{LV}$ to the molecular graph in case of both generators, VAE and GAN, with the function

$$d_{GEN} : \mathbf{h}_{LV} \mapsto G_{mol}. \tag{2}$$

2.3 | Graph-based property prediction

A GNN[59,60] is a type of neural network that operates directly on the graph structure and thus enables end-to-end learning in molecular property prediction. Thereby, GNNs avoid the need for the often subjective manual selection process of molecular descriptors in QSPR/QSAR modeling that requires intuition and experience of the modeler.

GNNs for molecular property prediction are typically structured into two parts, a message passing phase and a readout phase[83,84] (cf. Figure 3). In the message passing phase, structural information is extracted from a local neighborhood of atoms by means of graph convolutions. In each graph convolution, every node sends a message to all its neighbors and thus also receives a message from each of its neighbors. The node uses the received messages, typically in form of a weighted sum, to update its current state (e.g., in GCN[70] and GAT[85]). The update of the state $\mathbf{h}_v^l$ of a node v in a graph convolutional layer l can then be written as

$$\mathbf{h}_v^{l+1} = \sigma_{\text{ReLU}}\left(\mathbf{h}_v^l \mathbf{W}_1 + \sum_{u \in N(v)} \mathbf{h}_u^l \mathbf{W}_2\right), \tag{3}$$

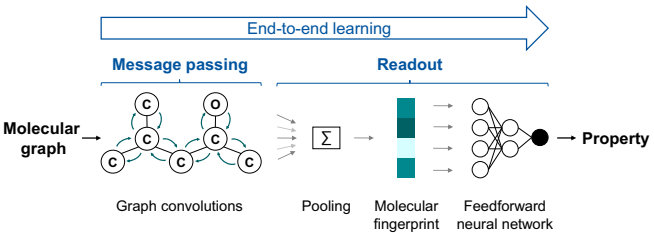


FIGURE 3 Schematic structure of a graph neural network for molecular property prediction

where $\mathbf{W}_1$, $\mathbf{W}_2$ are trainable weight matrices, $N(v)$ is the one-hop neighborhood of v, and σ_{ReLU} denotes the elementwise application of the ReLU activation function. Many different update functions have been proposed in the last years, see, for example, references 71,86,87, to advance the basic Equation (3) into a more powerful model for extracting information from the graph during message passing.[88] For instance, inter-atomic distances and angles between atom pairs[89–92] are commonly considered. Higher-order GNNs[93,94] and approaches where the information exchange is also based on individual edges[95] constitute further extensions to the basic GNN approach.

Subsequent to the message passing phase, a GNN employs a readout phase, where the molecular structure information that is stored in the nodes is aggregated into a single vector for the complete molecule, the so-called *molecular fingerprint* $\mathbf{h}_{\text{FP}}$. This aggregation, also called pooling, is typically performed by summing up the states of all nodes in the molecular graph after the last graph convolutional layer L, that is, $\mathbf{h}_{\text{FP}} = \sum_{v \in V} \mathbf{h}_v^L$. We denote the GNN encoding of the molecular graph into the molecular fingerprint with the function

$$g_{\text{GNN}} : G_{\text{mol}} \mapsto \mathbf{h}_{\text{FP}}. \tag{4}$$

Note that although the molecular fingerprint $\mathbf{h}_{\text{FP}}$ in a GNN and the latent vector $\mathbf{h}_{\text{LV}}$ in a generative ML model both represent a molecule in a continuous space, they are not related. In the GNN, the molecular fingerprint $\mathbf{h}_{\text{FP}}$ is passed through a feedforward neural network (cf. Figure 3) to yield the property prediction $\hat{p} = MLP(\mathbf{h}_{\text{FP}})$. Here, a multi-layer perceptron (MLP) is one of the simplest feedforward neural architectures and most frequently employed. We denote the entire end-to-end prediction process of a GNN as a function f_{GNN} that maps the molecular graphs to a property prediction, that is,

$$f_{\text{GNN}} : G_{\text{mol}} \mapsto \hat{p}. \tag{5}$$

3 | GRAPH-ML CAMD FRAMEWORK FOR HIGH-OCTANE FUELS

In this section, we propose a fully data-driven, modular graph-ML CAMD framework for identification of high-octane fuels. The framework utilizes recent methods from the field of generative graph-ML and GNNs to design molecules with high-knock resistance for modern SI engines. Specifically, we set out to maximize the sum of RON and OS, hence the OI, as high-efficiency SI engines require both a high RON and a high OS.[9–15] We show a high-level overview of our framework in Figure 1 and provide a detailed framework overview including our choices for algorithms and models in the three different modules in Figure 4. We combine the three modules to form an iterative molecular design loop: The *optimization module* proposes initial latent vectors from a continuous space, $\mathbf{h}_{\text{LV}}$, that are translated to corresponding molecules by the *molecule generation module*, cf. Equation (2). Then, the *property prediction module* performs the property evaluation, cf. Equation (5), and based on the property predictions, the optimization algorithm suggests new latent vectors to be tested. This iterative procedure is repeated until a pre-defined stopping criterion is met, for example, a certain number of molecules has been evaluated.

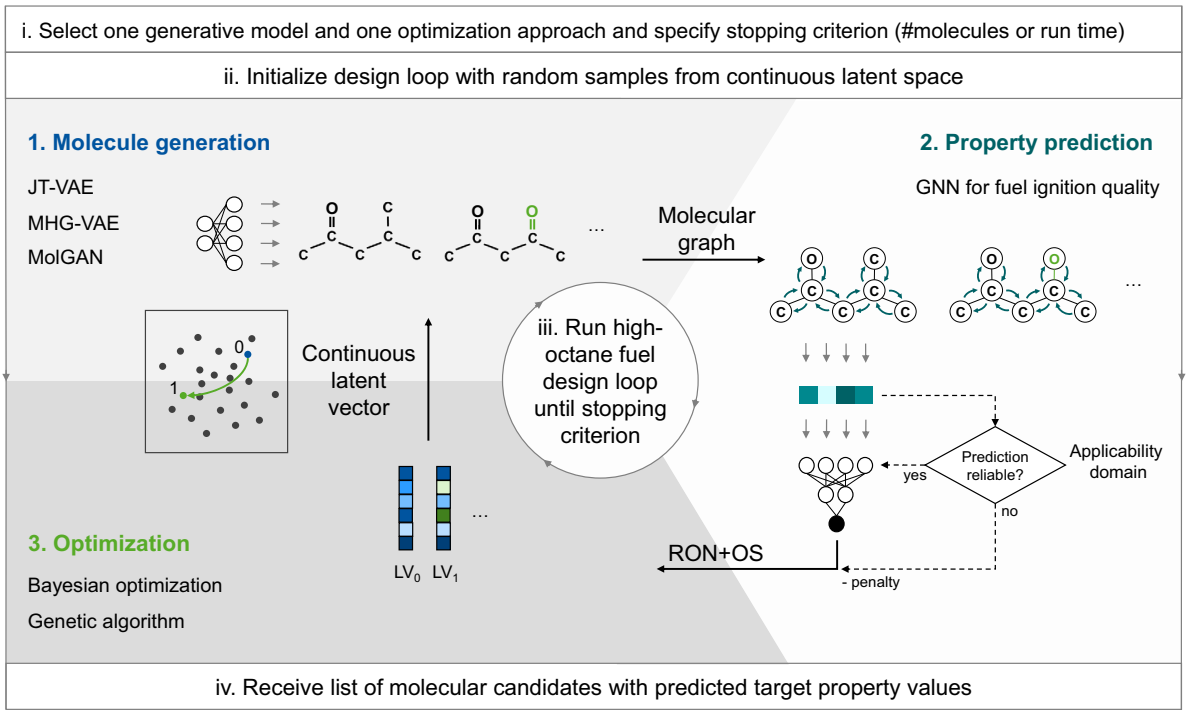


FIGURE 4 Detailed overview of the modular graph-ML CAMD framework for identification of high-octane fuels including methods for the individual modules

An important observation with the graph-ML CAMD framework though is that not all molecules come with physically reasonable predictions. For instance, we have observed a molecule with predicted OS > 400 and negative RON and negative MON.† In fact, the optimization often exploits weak spots of the GNN prediction model. Those weak spots typically appear for molecules that are strongly dissimilar from the molecules used for training the GNN. To focus on molecules with more reasonable property predictions, we extend the iterative design loop by an *applicability domain* (AD) for the GNN property prediction model. To this end, we build upon the AD approach from our previous study[96] where we proposed to use a one-class classification model to identify the AD of feedforward NNs. The classification model learns from the data on which the NN is trained to decide if a new data point is similar to the training data and thus considered within the input domain for which the NN presumably provides reliable predictions. To transfer the AD approach to GNNs, we apply the classification model to the molecular fingerprint that serves as input to the MLP part of the GNN (cf. Section 2.3). If the AD is included, GNN predictions considered unreliable by the AD are ignored and instead a penalty value (−1000) is returned to the optimization approach so that the corresponding molecules are assigned a low objective value.

The design loop runs can be formulated as an optimization problem that aims to find the molecules with the highest predicted value of a certain target property $\hat{p}$ of interest, that is,

$$\begin{aligned} \max_{\mathbf{h}_{LV}} \quad & \hat{p} \\ \text{s.t.} \quad & G_{mol} = d_{GEN}(\mathbf{h}_{LV}), \\ & \hat{p} = f_{GNN}(G_{mol}), \\ & \mathbf{h}_{FP} = g_{GNN}(G_{mol}), \\ & AD(\mathbf{h}_{FP}) \geq 0, \end{aligned} \tag{6}$$

whereby the constraint with $AD(\mathbf{h}_{FP}) \geq 0$ denotes a positive decision by the AD model.

Due to the high dimensionality of the search space that corresponds to the latent space of the generator models (see Equation (6)), deterministic global optimization is too computationally costly and practically impossible with current methods (cf. Section 3.3 below). Instead, we employ black-box optimization approaches that direct a heuristic search toward molecules with high $\hat{p}$. Note that uncertainties in the prediction model prohibit a strict ranking of molecular candidates with similar $\hat{p}$ values. Practically, we therefore compile a list of molecules sampled by the optimizer and perform an investigation of the top candidates, that is, the molecules with the highest $\hat{p}$ values. Having multiple top candidates, also allows to take additional desired properties into account in later investigations, for example, availability for procurement and low production costs.

In the following, we briefly describe the three generative graph-ML models used in this article for the generation module, the GNN model used for the property prediction module, the two optimization algorithms used in the optimization module, and our AD approach.

3.1 | Molecule generation

We consider two graph VAE models as generators: The Junction-Tree VAE by Jin et al.[51] (JT-VAE) and the Molecular Hypergraph Grammar VAE by Kajino[52] (MHG-VAE). Furthermore, we employ MolGAN, a GAN for molecular graphs published by De Cao and Kipf.[54] Those three models have close to 100% chemical validity, that is, almost 100% of the generated molecules are chemically feasible,[51,52,54] a feature that earlier generative methods struggled with, cf. references 97,98. Apart from achieving high validity, the three models have strong conceptual differences, presumably leading to molecules with somewhat different characteristics.

The JT-VAE[51] utilizes two graph representations of a molecule in parallel: The molecular graph and its associated junction tree, which is a contracted cycle-free graph generated by merging cycles of atoms into a single node. For encoding, the JT-VAE learns molecular structure information, represented as high-dimensional vectors, from the molecular graph and the junction tree through graph convolutions (cf. Section 2). For decoding, first, the junction tree's latent vector is decoded resulting in the general molecular structure. Then, the molecular graph's latent vector is decoded to determine the characteristics of the nodes within the junction tree, that is, (re)generating the local structure of the molecule. Jin et al. report a molecule reconstruction rate of 76.7% and 100% chemical validity of the decoded molecules.[51]

The MHG-VAE[52] generates a graph grammar from the given training molecules which is used for the reconstruction of molecules. In this automatically generated graph grammar, terminal symbols can refer to either single atoms or complete functional groups and the rules of the grammar describe how such atoms of partial molecules can be combined into a chemically valid molecule. During the generation of the grammar, MHG-VAE ensures that the grammar accounts for chemical feasibility constraints such as valency rules, explaining the validity of 100%.

MolGAN[54] only partially relies on graphs. Its adaptation to our case of high-octane fuel design is illustrated in Figure 5. The generator tries to directly predict a molecular graph's adjacency matrix with corresponding atom and bond features by using an MLP with a fixed output size, that is, the maximal size of a molecule that can be predicted

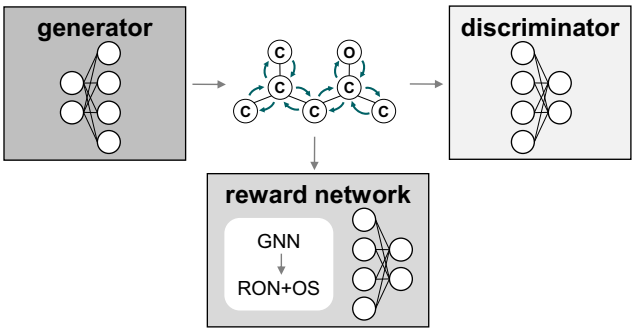


FIGURE 5 Adapted MolGAN for high-octane fuels, modified from reference 54. The reward network is coupled with our GNN[68] for predicting RON and OS values.

by MolGAN is bounded. On the other hand, the discriminator is a GNN. One conceptual difference to the VAEs is that MolGAN is able to focus the generation on molecules with desirable properties by using a "reward network", that is, a third network that encourages the generator to output molecules with high RON and OS. We use our GNN model[68] to provide RON and OS predictions such that, in contrast to the VAEs, the training of MolGAN partially depends on the property prediction module. De Cao and Kipf state that while MolGAN generates novel molecules with desirable properties and almost 100% chemical validity, it also outputs many duplicates with only about one in 10 molecules being unique.[54]

3.2 | Property prediction

We recently developed a GNN for predicting the RON, MON, and the derived cetane number (DCN) of a wide range of oxygenated and non-oxygenated hydrocarbons,[68] for example, (cyclo-) alkanes, (cyclo-) alkenes, alcohols, esters, ethers, aromatics, and ketones. The model architecture is based on higher-order GNNs[93] and additionally leverages the increased stability and accuracy of ensemble methods,[99,100] that is, the final property prediction is the average of multiple higher-order GNN predictions. Further, our GNN incorporates multitask learning[101,102] as it was trained on RON, MON, and DCN values simultaneously allowing the model to capture and exploit correlations between octane and cetane numbers.

As described in detail in reference 68, we compiled a data set comprising 335 RON, 318 MON, and 236 DCN values for 505 unique molecules in total to train the GNN. 85% of the data was used for training and validation, and 15% was used for testing. Note that for most molecules, both RON and MON values and thus OS were available. The mean absolute prediction error of the GNN model was 4.5 on the RON test set and 4.4 on the MON test set, indicating an overall high prediction quality on par with state-of-the-art QSPR- and ML-based RON and MON prediction models, cf. reference 68. The test sets also contain few outliers: Six predictions for RON and seven predictions for MON have a deviation >10, which we attribute, similarly to vom Lehn et al.,[103] to some of these molecules having unique characteristics that are not well represented in the training data, a relatively small number of data points available with low RON and MON values, and potential disruptive factors in experimental data assembled from different sources.

3.3 | Optimization

To sample molecules with high RON and OS from the latent space of the generative models, we employ numerical optimization using the RON + OS score predicted by the GNN model as objective function. Specifically, we seek to maximize $\widehat{p} = \mathrm{RON} + \mathrm{OS} = 2 \cdot \mathrm{RON} - \mathrm{MON}$ (cf. Equation (6)). We explore two derivative-free stochastic global optimization methods to perform the molecule sampling: A Bayesian optimization algorithm and a genetic algorithm.

Bayesian optimization (BO) is a probabilistic approach for global optimization[104] commonly used for optimization of black-box models that are costly to evaluate. Usage of BO is well-established in ML-based CAMD, see, for example, references 51,52,75, as well as in chemical engineering applications, for example, the design of experiments in automated reaction platforms.[105–107] BO uses a surrogate model, typically a Gaussian process (GP), to map the input variables to the objective. Based on the surrogate model, an acquisition function locates input variable values that have a high potential of maximizing the objective by accounting for both exploitation and exploration. For running BO, the GP is initialized with a set of feasible points. Then, the following steps are repeated until a termination criterion is reached: The acquisition function is optimized to determine the next sampling points, the sampling points are evaluated with respect to the objective function, and the objective values are used to refine the surrogate model. Note that different optimization algorithms can be used for maximizing the acquisition function, cf. reference 104.

A *genetic algorithm* (GA) is a meta-heuristic, population-based approach for global optimization that is inspired from evolutionary theory.[108,109] It is typically applied to optimization problems with cheap and fast evaluations of the objective function. In GAs, a set of feasible points is called population. Each feasible point has genes corresponding to specific values for the input variables of the optimization problem and constitutes a fitness related to the objective value. To solve an optimization problem, an initial population evolves in an iterative manner over multiple generations by promoting points with high fitness and using evolutionary heuristics, for example, combining genes of high fitness points, to replace points with low fitness. We choose the fitness to be RON + OS to directly optimize for high-octane ratings.

A major challenge in ML-based CAMD is the high dimensionality of the generators' latent space which typically requires a large number of sampling points for optimization, for example, in case of our generative models, we have latent space dimensionalities of 56 (JT-VAE),[51] 72 (MHG-VAE),[52] and 32 (MolGAN).[54] BO, however, employs a GP as surrogate model that in standard form has cubic scaling in complexity with respect to the number of sampling data points. Following the strategy by Kajino,[52] we thus use PCA to reduce the dimensions of both the JT-VAE and the MHG-VAE before performing BO. Since the execution time of the evolutionary-based heuristics in the GA does not suffer from a high number of sampling points, we run the GA without dimensionality reduction. Note that the effects of PCA-based dimensionality reduction on the obtained molecules as well as the use of other mitigation strategies, such as reduction of the latent dimension within the generator or modification of BO for high-dimensional problems, see, for example, references 104,110–113, are beyond the scope of this work.

3.4 | Applicability domain

The AD of a model is a well-established concept in QSPR/QSAR modeling and is based on the general assumption that the prediction model would provide most reliable predictions for molecules that are similar to the ones seen during training.[114–117] Molecular similarity is

usually assessed by means of a distance metric, for example, the Euclidean distance between the descriptor values of two molecules.[115,118] For molecular property prediction with GNNs, determination of the AD is largely unexplored. Only very recently first approaches to quantify the AD of GNNs based on uncertainty quantification methods were proposed.[119–121] Conceptually, defining the AD of a GNN requires handling the varying input sizes of molecular graphs and measuring the degree of similarity between different graphs. In this work, we address these challenges by extending our recently developed AD approach based on one-class support vector machines (SVMs)[96] to GNNs. A one-class SVM is a ML model that can be used to identify outliers by classifying whether an input is similar or dissimilar to the training data. We train one-class SVMs on the molecular fingerprint of the GNN (cf. Figure 3) to determine the GNN's AD. We then restrict our molecular design loop to molecules which are accepted by the SVM (cf. Equation (6)) which formally means $AD(\cdot) = SVM_{\text{AD}}(\mathbf{h}_{\text{FP,train}}) \geq 0$ where $\mathbf{h}_{\text{FP,train}}$ is the molecular fingerprint computed by the GNN and SVM_{AD} denotes the trained SVM. The underlying idea for the AD is that the GNN computes similar molecular fingerprints whenever two molecules are structurally similar. Since our prediction model is an ensemble of multiple GNNs, we train one SVM for each GNN model in the ensemble and apply a majority vote. That is, each SVM j evaluates $SVM_{\text{AD},j}(\mathbf{h}_{\text{FP}})$ and returns 1 if the molecule lies within the AD or -1 if not. Subsequently, we sum up the votes to decide if the prediction of the GNN ensemble (EL) for a new molecules is classified as reliable, that is, $SVM_{\text{AD-EL}}(\mathbf{h}_{\text{FP}}) = \sum_j SVM_{\text{AD},j}(\mathbf{h}_{\text{FP}}) \overset{!}{>} 0$. Note that further details on the AD are described in the Supporting Information S1.

3.5 | Implementation and hyperparameters

We implement our graph-ML CAMD framework in Python with the cheminformatic package RDKit[122] and the ML frameworks pytorch and tensorflow, accounting for the different implementations of the generators, and provide our code open-source, see, reference 67. Moreover, we follow the implementation of the MHG-VAE by Kajino[52] and use Luigi[123] to automate computational experiments. For the three generators, JT-VAE,[51] MHG-VAE,[52] and MolGAN,[54] we use the original implementations and hyperparameters as provided in the respective study and code repository and only extend the code to work in our framework. We train the molecule generation models on all HCO-molecules in the QM9 data set,[124,125] that is, all molecules within QM9 that contain exclusively hydrogen, carbon, and oxygen atoms. QM9 contains approximately 50,000 HCO-molecules from various molecular classes. We use the original implementation and model parameters of our GNN[68] which is based on pytorch-geometric.[126] The SVMs for the AD are implemented with scikit-learn[127] building on our AD study.[96] For BO, we use GPyOpt.[128] Note that we did not attempt deterministic global optimization of the acquisition function within the BO, for example, by using our tool MeLOn,[129,130] due to the high dimensionality (cf. Section 3.3) and associated high computational cost. Thus, we use the local optimization algorithm L-BFGS[131] implemented in GPyOpt.[128] As GA, we use the python package geneticalgorithm.[132] For both BO and GA, we apply default settings. We follow the study of MHG-VAE by Kajino[52] and reduce the dimensionality of the latent space within the VAEs by means of PCA aiming for an explained variance ratio of 99.9% (JT-VAE: from 56 to 41, MHG-VAE: from 72 to 38) before performing BO. Further details on the hyperparameter choice can be found in the Supporting Information S1. We run all computations on the HPC-cluster (CLAIX-2018) of RWTH Aachen University using one Supermicro 1029GQ-TVRT-01 node of an Intel Platinum 8160 core with 192 GB RAM, of which we used at most 8 GB, plus one NVIDIA Volta V100-SXM2 16 GB GPU. For reproducibility, we fixed random seeds for training the models and running the design loop that we provide with our code.

4 | RESULTS AND DISCUSSION

We first present the computational results of our graph-based CAMD of high-octane fuels and then provide a discussion of the top candidates to demonstrate both strengths and potential weaknesses of the fully data-driven design approach.

4.1 | CAMD results

We test all combinations of the three generator models (JT-VAE, MHG-VAE, and MolGAN) and the two optimization approaches (BO and GA) as well as two different stopping criteria (SC), that is, a limit on the number of candidate molecules generated ($SC_{\#molecs}$) and an upper limit on the wall-clock run time (SC_{time}). For $SC_{\#molecs}$, we consider both the number of unique molecules (1000) and the total number of molecules (2000) generated, as the number of duplicates can otherwise cause an unlimited run time. In the $SC_{\#molecs}$ setting, the design loop will typically run for 0.5–8 h. The run time limit in SC_{time} is set to 12 h to investigate the effects of keeping the design loop running for a longer time. Furthermore, we distinguish between runs with and without the AD. All design loop runs are run five times (initialized with different random seeds) and the results are aggregated.

The top 12 molecules identified with $SC_{\#molecs}$ and active AD for the respective generators are shown in Figure 6 together with the predicted RON and OS values. The results demonstrate that the generators successfully propose molecules with high predicted RON and OS. Moreover, the top molecules are from a variety of different molecular classes, for example, ethers, alcohols, and ketones, some of which are known to contain promising SI engine fuel candidates.[2] The majority of molecules has at least one oxygen atom. Almost all top molecules generated by MolGAN include a cyclic structure, often associated with a cyclopropane feature, which we attribute to the high RON and OS for components with a cyclopropane substructure in the training set of the GNN model.[133] Most top molecules generated by the two VAE models include strongly branched non-cyclic

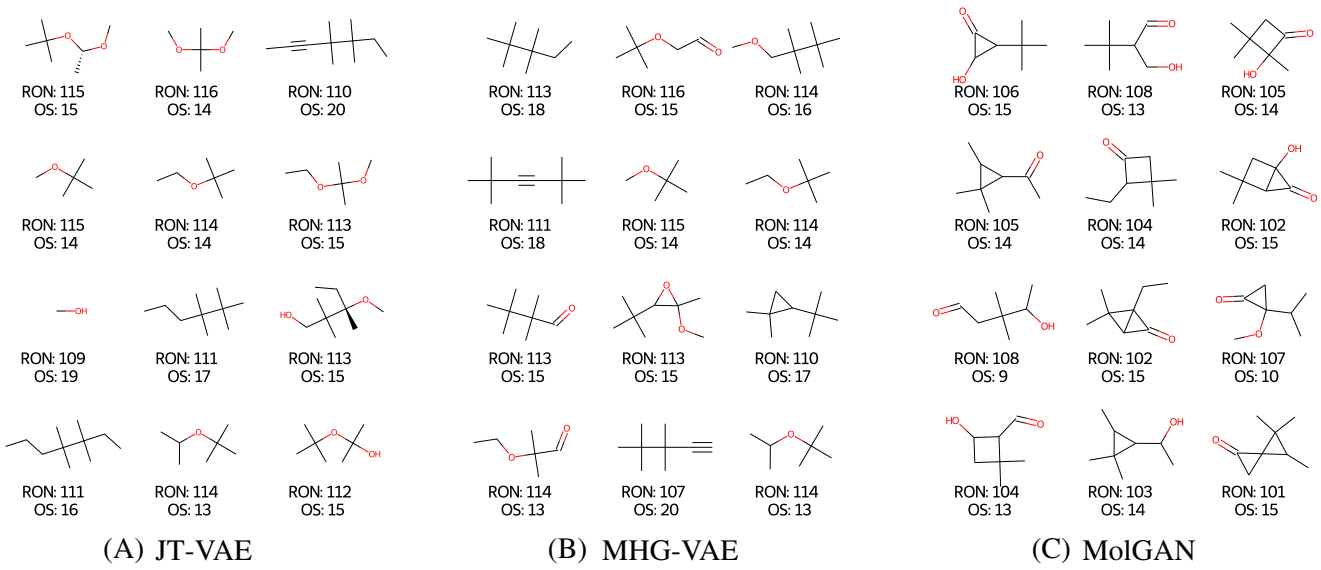


FIGURE 6 Top 12 candidates identified by the three different generator models with stopping criterion $SC_{\#molecs}$ (max. 1000 unique molecules or max. 2000 total molecules) and applicability domain. RON and OS values are predicted by the graph neural network.[68]

TABLE 1 Results of optimization over five runs each

		JT-VAE				MHG-VAE				MolGAN			
Predicted RON + OS		**BO**	**BO + AD**	**GA**	**GA + AD**	**BO**	**BO + AD**	**GA**	**GA + AD**	**BO**	**BO + AD**	**GA**	**GA + AD**
$SC_{\#molecs}$ (1000 unique molecules, 2000 total) #runs: 5	Max	205	130	129	130	138	129	136	131	121	121	121	121
	Mean top 20	181	125	125	126	131	125	132	128	110	111	116	116
	# unique mol.	2390	1347	3472	3712	4671	4308	4683	4427	21	21	46	46
	# promising mol.	117	10	15	19	45	9	52	30	0	0	0	0
SC_{time} (12 h run time) #runs: 5	Max	205	130	187	131	138	129	145	131	121	121	121	121
	Mean top 20	183	126	180	130	133	126	140	129	111	112	118	118
	# unique mol.	2996	1935	109,830	80,818	6710	7081	55,255	46,989	22	23	193	172
	# promising mol.	140	12	2096	376	104	15	678	142	0	0	0	0

Note: A molecule is considered promising if both RON > 110 and OS > 10. Runs with applicability domain are indicated by +AD.

components, often in combination with one or two oxygen atoms, which are also known for high RON and OS values. Both VAE models generate the popular octane enhancers MTBE and ETBE, and some related small, branched ether structures. The JT-VAE also identifies ethanol, the prototype biofuel for SI engines.

Table 1 shows the statistics of all the runs with and without the AD, whereby each entry corresponds to the aggregated results over five runs. Both the maximum and the mean predicted RON + OS are typically lower if the AD is used. In most cases, also the total number of molecules generated is lower if the AD is considered. The observation that the AD often reduces the exploration performance is expected and in fact intended as the AD prohibits the generators from exploring structures that are far from the training data by strongly extrapolating the GNN model. We want to emphasize that we find the generators to mainly produce chemically valid molecules. Otherwise, for example, MolGAN sometimes generates disconnected substructures, the generated molecule is dropped so that effectively no chemically invalid structures are provided to the GNN and AD. Note that generated molecules, which are considered highly dissimilar to the training molecules by the AD, can still be chemically valid. We show examples of such chemically valid molecules well outside the GNN's AD in Figure 7, where the top candidates identified by the two VAEs with SC_{time} are depicted; we refer to the Supporting Information S1 for further examples.

When visually inspecting the top molecules from the design runs without AD, we find that the obtained molecules are typically huge, strongly branched hydrocarbons, for example, with up to almost 50 carbon atoms. As such compounds are presumably solid at room temperature, they are not suitable as fuels. To avoid the formation of solids within the fuel blend, a constraint on the melting point could be included in the design loop. However, the melting point can only serve as a rough proxy for the suitability of a compound as an octane booster, since miscibility and volatility also depend on the composition of the base fuel and the blending ratio.[42,134] Some of the proposed large molecules might be soluble in a fuel blend, which could be evaluated in further investigations of mixture properties, but is beyond the scope of this work. Furthermore, the RON and OS predictions for the molecules identified with the JT-VAE without AD (cf. Figure 7A) are visibly higher than the maximum RON (of 120 for 1,3,5-trimethylbenzene[68,135]) and the maximum OS (of 36 for 1,4-cyclohexadiene[68,135]) of the data used to train the GNN prediction model, indicating strong extrapolation. In the following, we therefore present and discuss only those results that have been obtained with the AD.

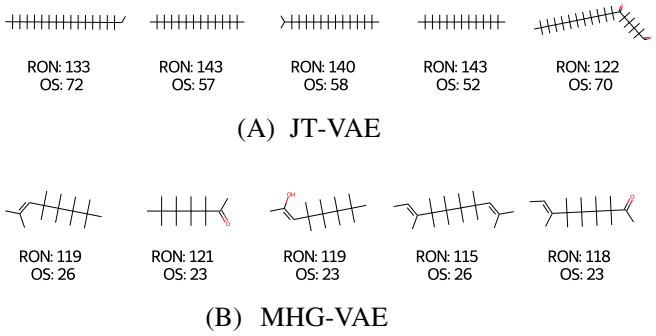


FIGURE 7 Top five candidates identified by the two VAE generator models with stopping criterion SC_{time} (12 run time) and without applicability domain. All RON and OS values are GNN predictions.[68]

We observe that the VAE generators predict molecules with a maximum RON + OS of about 130 while MolGAN achieves a maximum of only 121 (cf. Table 1). The maximum RON + OS values of slightly above 130 for the two VAE models are in good agreement with known high-octane fuels such as MTBE with its experimentally validated RON + OS of 135. The encouraging performance of both VAE generators thus shows the general feasibility of our graph-ML CAMD framework utilizing the SVM-based AD.

To further compare the different generator and optimization combinations, we analyze the number of distinct molecules generated as well as the number of molecules with promising ignition properties, that is, the molecules with both a predicted RON > 110 and a predicted OS > 10. Both VAEs find a large number of distinct molecules irrespective of the employed stopping criteria (cf. Table 1). Specifically for $SC_{\#molecs}$, both VAEs generate more than 3500 unique molecules out of 5000 maximally possible unique molecules (1000 unique molecules each over five runs). This means that not only do the VAEs find a large number of distinct molecules in each run, but the identified molecules also vary greatly between different runs, thus leading to an overall small number of duplicates. In contrast, MolGAN mainly generates duplicates of which none are considered promising (cf. Table 1). Comparing the results for $SC_{\#molecs}$ and SC_{time} (cf. Table 1), it can be seen that the VAE-GA combinations significantly increase the number of both explored and promising candidates with longer run time. Apparently, this observation does not extend to BO, with one possible explanation being that BO becomes inherently slower as more data points are added to the surrogate model, thereby reducing the number of predictions per time, whereas the corresponding rate remains unchanged in the GA (cf. Section 3.3).

The predicted RON and OS values of all promising molecules obtained with the two stopping criteria are shown in Figure 8. We also highlight those molecules identified in the $SC_{\#molecs}$ setting that are commercially available at chemical suppliers. Commercial

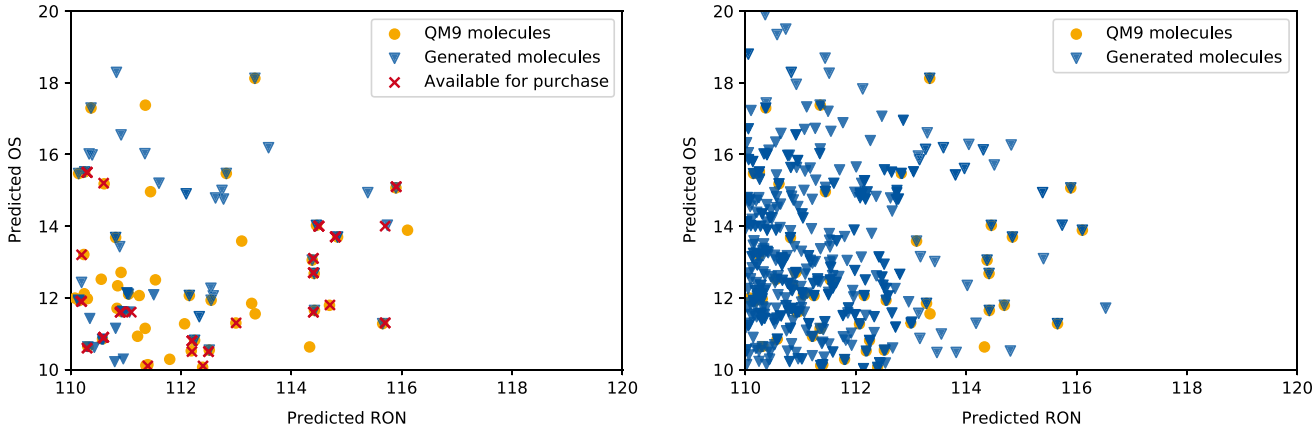


FIGURE 8 Promising candidates (predicted RON > 110 and OS > 10). Commercially availability (red crosses) determined by manual search on Sigma-Aldrich and Chemspider websites[136,137]

availability was assessed by a manual search on Sigma-Aldrich[136] and Chemspider[137] websites without imposing a price limit but only including those molecules with an explicitly stated price; we did not search for the lowest price on different websites. For SC_{time}, Figure 8B, the effort for a manual search was considered disproportional due to the high number of promising candidates. We further indicate molecules with high predicted RON + OS in the QM9 database[124,125] that is used for training the generative models; additional QM9 statistics are provided in the Supporting Information S1. Figure 8 demonstrates that the graph-ML CAMD framework is able to generate molecules with high predicted RON and high predicted OS that are not in the QM9 database. This observation is emphasized in case of SC_{time} (cf. Figure 8B). The capabilities of the generator models to generalize therefore allow to explore novel molecules for further investigation.

4.2 | Discussion of top candidates

In the discussion of the top molecules, we restrict our analysis to the promising molecules (RON > 110 and OS > 10) generated using $SC_{\#molecs}$, as the number of molecules generated with SC_{time} is very large; we refer to the Supporting Information S1 for a detailed list of all generated promising molecules. The top molecules that are also commercially available are illustrated in Table 2, including RON and OS predictions, literature values for RON and OS (where available), price category, and the respective combinations of generator and optimizer that identified the molecule.

4.2.1 | Promising classes of molecules

We find both pure hydrocarbons and oxygenated hydrocarbons (cf. Table 2), molecules already in use as octane boosters and molecules that constitute interesting candidates for further experimental investigation. The two identified alkanes, ethane and cyclopropane, are gaseous under ambient conditions, whereas the one aromatic hydrocarbon, 3-ethyltoluene, is liquid. The known RON + OS scores from literature for ethane and 3-ethyltoluene of 122 and 124, respectively, are in good agreement with the GNN predictions. We want to emphasize that gaseous compounds, such as ethane and cyclopropane, are difficult to implement as octane boosters. To prevent gases within the candidate list, one could include boiling point constraints in the design loop. However, the normal boiling point is, similar to the melting point discussed at the beginning of this section, only a rough preselection criterion, since the miscibility and volatility of a potential octane booster in a fuel blend strongly depend on the overall blend composition. Next to alkanes, three ethers are identified, including methyl tert-butyl ether (MTBE) and ethyl tert-butyl ether (ETBE) that are used as octane boosters in practical applications.[16,17] Their experimentally RON + OS scores of 135 and 134[138,139] are slightly higher than the predicted scores. Furthermore, molecules from the class of aldehydes are identified. It has been found, however, that the formation of aldehydes during the combustion process of high-octane, oxygenated hydrocarbons results in increased exhaust emissions,[143] indicating a lower suitability of aldehydes as fuels. Polyfunctional molecules with an aldehyde and an ether group are generated as well, which also entail the problem of aldehyde emissions. Further polyfunctional molecules containing an ether group and a ketone group are generated, with ketones being prominent high-octane fuels.[144,145] Most of the molecules containing an ether, a ketone, and/or an aldehyde functionality have a compact, branched structure with similarities to MTBE and ETBE, making them interesting high-octane fuel candidates; however, they also have a high price, hindering experimental investigation.

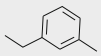
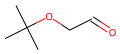
The last top candidate in Table 2, namely 2,2-dimethoxypropane (2,2-DMP), belongs to the class of acetales. It is a compact structure similar to ETBE, with the difference being that one carbon atom is replaced by a second oxygen atom. 2,2-DMP also has a low price, making it an attractive target for experimental investigation. A DCN measurement of 31 is known from literature[146] which, however, is not suggestive of a very high RON, as molecules with RON > 110 typically correspond to DCN values below 10, cf. references 2,147. Our high RON + OS prediction (cf. Table 2), however, is consistent with the RON + OS value of 143 stated in a recent study by Li et al.[18] who used a ML-QSPR prediction model combining both ML and a group contribution approach. Another ML-based QSPR model for RON and OS recently developed by vom Lehm et al.[103] likewise predicts a high RON + OS value of 156.

4.2.2 | Comparison to previous fuel design studies

Our commercially available top candidates (cf. Table 2) generally match the molecular classes identified in previous fuel design/screening studies for SI engine fuels, for example, in references 2,18,40,148. Specifically, prominent molecular classes from previous studies include the herein identified groups of ethers,[2,18,148] ketones,[2,18,40,148] aromatics,[148] aldehydes,[18,40] alkanes,[148] and acetals.[18] Interestingly, our top candidates do not include any esters, alcohols, and furans that have often been identified in the literature.[2,18,148] When inspecting all molecules generated in our design loop runs with $SC_{\#molecs}$ and with AD, we indeed find esters (e.g., methyl acetate), alcohols (e.g., ethanol and methanol), as well as furans (e.g., 2-methylfuran). However, these are not considered top candidates as predicted OS is below 10 for most esters and predicted RON is slightly below 110 in case of furans and alcohols. Such RON and OS predictions are generally in accordance with the literature values for representative molecules of these classes, cf. references 42,68,135,149,150.

The polyfunctional molecules identified in our study are hardly discussed in the literature. It should be noted that the availability of experimental RON and MON values for polyfunctional molecules is very limited, indicating a high uncertainty in the GNN predictions.

TABLE 2 All 16 commercially available molecules with predicted RON > 110 and OS > 10 (identified in $SC_{\#molecs}$ setting and active applicability domain)

Class	Structure	SMILES	RON	OS	Price category	Generator (optimizer)
Alkanes		C1CC1 cyclopropane	110	16	Medium	JT (BO, GA), MHG (BO, GA)
		CC ethane	110 (111[135])	12 (11[135])	Low	JT (BO, GA)
Aromatics		CCc1cccc(C)c1 3-ethyltoluene	110 (112[135])	11 (12[135])	High	JT (GA)
Ethers		COC(C)(C)C MTBE	115 (118[138])	14 (17[138])	Low	JT (BO, GA), MHG (BO, GA)
		CCOC(C)(C)C ETBE	114 (118[139])	14 (16[139])	Medium	JT (BO, GA), MHG (GA)
		CC(C)OC(C)(C)C tert-butyl isopropyl ether	114	13	High	JT (GA), MHG (GA)
Aldehydes		CC(C=O)C(C)(C)C 2,3,3-trimethylbutanal	111	12	High	MHG (GA)
		CC(C)(C)C=O trimethylacetaldehyde	111	11	Medium	JT (GA), MHG (BO)
Polyfunctional (aldehyde + ether)		CC(C)(C)OCC=O tert-butoxyacetaldehyde	116	15	High	MHG (GA)
		CCOC(C)(C)C=O 2-ethoxy-2-methylpropanal	114	13	High	MHG (GA)
		COC(C)(C)C=O 2-methoxy-2-methylpropanal	116	11	High	MHG (GA)
		CC(C)OC(C)(C)C=O 2-methyl-2-propan-2-yloxypropanal	114	12	High	MHG (GA)
		COC(C)C=O 2-methoxypropanal	112	11	High	MHG (BO, GA)
Polyfunctional (ketone + ether)		COC(C)(C)C(C)=O 3-methoxy-3-methyl-2-butanone	113	11	High	MHG (GA)
		COC(C)C(=O)C(C)(C)C 4-methoxy-2,2-dimethylpentan-3-one	111	12	High	MHG (GA)
Acetals		COC(C)(C)OC 2,2-dimethoxypropane	116	14	Low	JT (GA)

Note: RON and OS data available in the literature are stated in parentheses. Prices are categorized based on data from different chemical suppliers[136,140–142]: ≤1000$/l (low), >1000 $/l and ≤10,000$/l (medium), >10,000 $/l (high).

The generated top candidate of acetals, 2,2-DMP, has also been identified in the fuel screening by Li et al.[18] and will be investigated experimentally in the following.

4.2.3 | Experimental assessment of 2,2-DMP

Experimental investigation of 2,2-DMP was conducted in dedicated test engines according to the DIN EN ISO 5164[151] and DIN EN ISO 5163 standards,[152] respectively, by an external company. Measurement of RON and MON of pure 2,2-DMP, however, could not be performed. Instead, blends of 2,2-DMP with 90%, 80%, and 60% (v/v) of gasoline were investigated. The extrapolation to pure component values yielded a RON of 91.75 (±0.25) and a MON of 87.27 (±0.3), hence a RON + OS score of about 96, indicating a strong misprediction by our GNN model as well as the models by Li et al.[18] and by vom Lehn et al.[103] To further clarify the ignition properties of 2,2-DMP, we experimentally measured ignition delay times (IDT) in a rapid compression machine (RCM)[153,154] and compared the chemical reactivity of 2,2-DMP to that of a typical RON95E10 pump station fuel. IDT measurements for 2,2-DMP were performed at an end-of-compression pressure of 20 bar for a stoichiometric mixture and with an nitrogen-to-oxygen dilution ratio of 3.762 in the temperature range of 647–793 K. Details on the RCM measurements can be found in the Supporting Information S1. The ignition took place via a two-stage process in the investigated temperature regime indicating strong low-temperature chemistry, cf. Figure 9, not representative for a high-octane fuel. Compared to the RON95E10 fuel, 2,2-DMP shows a distinctively higher reactivity between 647 and 750 K pointing toward a lower knock resistance and thus RON value. The RCM results suggest

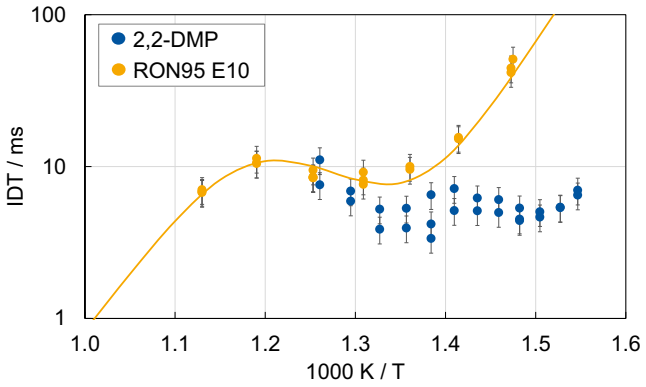


FIGURE 9 Measured ignition delay time in a rapid compression machine for 2,2-dimethoxypropane and a commercially available RON95E10 pump station fuel. The error bars indicate ±20% scatter of the measured ignition delay time. The yellow line corresponds to a threefold Arrhenius model fit to the RON95E10 ignition delay times.[155]

a slightly worse knock resistance of 2,2-DMP compared to RON95E10 pump station fuel, supporting the extrapolated RON and MON measurements.

The case of 2,2-DMP shows the potential weaknesses of a fully data-driven approach in a data-scarce environment. We account the large model prediction error of our GNN as well as those of the models by Li et al.[18] and by vom Lehn et al.[103] to the comparatively little training data available for RON and MON modeling. Specifically, our RON and MON training database includes just five ethers, a single acetal (not 2,2-DMP), no aldehydes, eight ketones, and only two molecules with more than one type of oxygen functionality (cf. reference 68). Similar data limitations apply to the other RON and MON prediction models,[18,103] explaining their similarly bad predictions in case of 2,2-DMP. Furthermore, we want to stress the fact that no RON and MON values for aldehydes are included in the training data, so our GNN may not sufficiently distinguish between aldehydes and ketones. The RON + OS predictions of the identified molecules with an aldehyde group are therefore considered subject to large uncertainty. In the case of 2,2-DMP, a DCN data point was available and used in the training of our multitask GNN for simultaneous RON, MON, and DCN prediction (cf. Section 3.2). As expected, our AD approach based on majority voting (cf. Section 3.4) considers 2,2-DMP within the region of reliable predictions as it was part of the training data. Yet, only 31 out of 40 SVMs voted for 2,2-DMP. Increasing the AD consensus level, for example, 80% instead of 50%, may provide some protection against such strong mispredictions, at the cost of a smaller search space. A systematic investigation of the relationship between the AD consensus level and the prediction accuracy for molecules proposed by the design loop, however, is beyond the scope of this work. The weak spots of prediction models for fuel ignition quality remain a huge challenge for model-based fuel design, even when utilizing state-of-the-art ML[68,103] and an applicability domain. Therefore, acquiring more training data is absolutely crucial.

5 | CONCLUSION

We propose a fully data-driven CAMD approach based on recent methods from graph-ML for the identification of molecules with desired ignition characteristics for modern SI engines. Our graph-ML CAMD framework utilizes a representation of molecules as graphs and incorporates three modules for building a molecular design loop: (1) molecule generation from a continuous molecular space with generative graph-ML, (2) molecular property prediction through GNNs, and (3) optimization for strategic sampling from the continuous molecular space to find molecules with high predicted RON + OS. The modular structure enables the exploration of different ML models in combination with different optimization approaches. We additionally present a novel approach to identify the applicability domain (AD) of GNN models for molecular property prediction. By predicting promising high-octane fuel molecules in a fully data-driven fashion, our study exemplifies how recent developments in ML can be utilized for CAMD and its automation.

The top molecular candidates identified with our graph-ML CAMD framework are from well-known molecular classes for high-octane fuels, for example, ethers and ketones, and include both well-established components like MTBE and ETBE as well as new promising candidates for further experimental investigation. The comparison of different generative graph-ML models, namely JT-VAE,[51] MHG-VAE,[52] and MolGAN,[54] in combination with different optimization approaches, BO and GA, shows that the choice of the generative model and optimization strategy influences the number and type of identified candidate molecules. Both VAEs provide a diverse continuous molecular space with a large number of potential molecules, while MolGAN generates a comparatively low number of candidates and yields lower target property values. We conclude that the GA is well suited for exploring large portions of the continuous molecular space of the generative models, especially when working with high dimensions where BO struggles but still finds some promising candidates. Our AD approach additionally enables us to focus the exploration on candidates with presumably more accurate predictions. The experimental investigation of one candidate within the AD, namely 2,2-dimethoxypropane, shows lower RON and OS values than predicted by our GNN model, demonstrating the limitations of CAMD in a comparatively data-scarce environment. We thereby highlight the importance of experimental validation to fuel design and the need for further RON and OS training data. Furthermore, the correlation between the AD threshold, that is, the consensus level, and the prediction accuracy for molecules proposed by the design loop should be investigated.

Future work could include additional physical and chemical properties in the design, for example, melting point, boiling point, vapor pressure, toxicity, or viscosity, similar to previous studies.[2,18,40,148] The framework, in principle, is not bound to fuel design as application but could also be applied to other CAMD applications such as drug discovery, design of catalysts, pesticides, and so forth.

AUTHOR CONTRIBUTIONS

Jan G. Rittig: Conceptualization (equal); data curation (equal); formal analysis (equal); funding acquisition (supporting); investigation (equal);

methodology (equal); software (equal); validation (equal); visualization (lead); writing – original draft (lead); writing – review and editing (supporting). **Martin Ritzert:** Conceptualization (equal); data curation (equal); formal analysis (equal); investigation (equal); methodology (equal); software (equal); validation (equal); visualization (supporting); writing – original draft (supporting); writing – review and editing (supporting). **Artur M. Schweidtmann:** Conceptualization (equal); funding acquisition (supporting); methodology (supporting); supervision (supporting); writing – review and editing (supporting). **Stefanie Winkler:** Data curation (supporting); formal analysis (supporting); methodology (equal); software (equal); validation (supporting); writing – review and editing (supporting). **Jana M. Weber:** Methodology (supporting); writing – review and editing (supporting). **Philipp Morsch:** Investigation (equal); writing – original draft (supporting); writing – review and editing (supporting). **K. Alexander Heufer:** Funding acquisition (supporting); supervision (supporting); writing – review and editing (supporting). **Martin Grohe:** Conceptualization (supporting); funding acquisition (equal); supervision (equal); writing – review and editing (supporting). **Alexander Mitsos:** Conceptualization (supporting); funding acquisition (equal); supervision (equal); writing – review and editing (supporting). **Manuel Dahmen:** Conceptualization (supporting); formal analysis (equal); supervision (equal); writing – review and editing (lead).

ACKNOWLEDGMENTS

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—466417970—within the Priority Programme "SPP 2331: Machine Learning in Chemical Engineering." It was also funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy—Cluster of Excellence 2186 "The Fuel Science Center." This work was also performed as part of the Helmholtz School for Data Science in Life, Earth and Energy (HDS-LEE). Further, this work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) within the GRK 2236 UnRAVel. Simulations were performed with computing resources granted by RWTH Aachen University under project "rwth0664." The authors thank Florian vom Lehn for providing RON and OS predictions for 2,2-dimethoxypropane by his model. MD received funding from the Helmholtz Association of German Research Centres. Open Access funding enabled and organized by Projekt DEAL.

CONFLICT OF INTEREST

The authors have no conflict of competing interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in our GitLab repository "Graph machine learning for design of high-octane fuels" at https://git.rwth-aachen.de/avt-svt/public/graph_ML_fuel_design, reference 67.

ORCID

Jan G. Rittig https://orcid.org/0000-0003-4645-5716
Martin Ritzert https://orcid.org/0000-0002-5322-3684
Artur M. Schweidtmann https://orcid.org/0000-0001-8885-6847
Jana M. Weber https://orcid.org/0000-0002-2867-0087
Philipp Morsch https://orcid.org/0000-0001-6249-0648
Karl Alexander Heufer https://orcid.org/0000-0003-1657-5231
Martin Grohe https://orcid.org/0000-0002-0292-9142
Alexander Mitsos https://orcid.org/0000-0003-0335-6566
Manuel Dahmen https://orcid.org/0000-0003-2757-5253

ENDNOTES

* Code is openly available, see reference 67.

† Found by GA when optimizing for OS only.

REFERENCES

1. IEA. *The Role of CCUS in Low-Carbon Power Systems* [Tech. rep.]; 2020. Accessed September 8, 2021. https://www.iea.org/reports/the-role-of-ccus-in-low-carbon-power-systems
2. Dahmen M, Marquardt W. Model-based design of tailor-made biofuels. *Energy Fuel*. 2016;30(2):1109-1134.
3. Leitner W, Klankermayer J, Pischinger S, Pitsch H, Kohse-Höinghaus K. Advanced biofuels and beyond: chemistry solutions for propulsion and production. *Angew Chem Int Ed Engl*. 2017;56(20):5412-5452.
4. Gschwend D, Soltic P, Wokaun A, Vogel F. Review and performance evaluation of fifty alternative liquid fuels for spark-ignition engines. *Energy Fuel*. 2019;33(3):2186-2196.
5. König A, Siska M, Schweidtmann AM, et al. Designing production-optimal alternative fuels for conventional, flexible-fuel, and ultra-high efficiency engines. *Chem Eng Sci*. 2021;237:116562.
6. American Society for Testing, Materials. ASTM D 2699: Standard Test Method for Research Octane Number of Spark-Ignition Engine Fuel; 2018.
7. American Society for Testing, Materials. ASTM D 2700: Standard Test Method for Motor Octane Number of Spark-Ignition Engine Fuel; 2019.
8. Kalghatgi G. Fuel anti-knock quality—Part I. Engine studies. *SAE Trans*. 2001;110:1993-2004.
9. Kalghatgi GT. Auto-Ignition Quality of Practical Fuels and Implications for Fuel Requirements of Future SI and HCCI Engines. SAE Technical Paper 2005-01-0239; 2005.
10. Bell A. Modern SI Engine Control Parameter Responses and Altitude Effects with Fuels of Varying Octane Sensitivity. SAE Technical Paper 2010-01-1454; 2010.
11. Mittal V, Heywood JB. *The Relevance of Fuel RON and MON to Knock Onset in Modern SI Engines*. SAE Technical Paper 2008-01-2414; 2008.
12. Amer A, Babiker H, Chang J, et al. Fuel effects on knock in a highly boosted direct injection spark ignition engine. *SAE Int J Fuels Lubr*. 2012;5(3):1048-1065.
13. Kalghatgi GT. The outlook for fuels for internal combustion engines. *Int J Engine Res*. 2014;15(4):383-398.
14. Szybist JP, Splitter DA. Pressure and temperature effects on fuels with varying octane sensitivity at high load in SI engines. *Combust Flame*. 2017;177:49-66.
15. Abdul-Manan AFN, Kalghatgi G, Babiker H. Exploring alternative octane specification methods for improved gasoline knock resistance in spark-ignition engines. *Front Mech Eng*. 2018;4:259.
16. Demirbas A, Balubaid MA, Basahel AM, Ahmad W, Sheikh MH. Octane rating of gasoline and octane booster additives. *Petrol Sci Tech*. 2015;33(11):1190-1197.
17. Badia JH, Ramírez E, Bringué R, Cunill F, Delgado J. New octane booster molecules for modern gasoline composition. *Energy Fuel*. 2021;35(14):10949-10997.
18. Li R, Herreros JM, Tsolakis A, Yang W. Integrated machine learning-quantitative structure property relationship (ML-QSPR) and chemical kinetics for high throughput fuel screening toward internal combustion engine. *Fuel*. 2022;307(1):121908.

19. Joback KG. *Designing Molecules Possessing Desired Physical Property Values* [PhD thesis]. Massachusetts Institute of Technology, USA; 1989.
20. Achenie LEK, Gani R, Venkatasubramanian V. *Computer Aided Molecular Design: Theory and Practice. Vol. 12 of Computer-Aided Chemical Engineering*. 1st ed. Elsevier; 2003.
21. Gani R. Chemical product design: challenges and opportunities. *Comput Chem Eng*. 2004;28(12):2441-2457.
22. Zhang L, Cignitti S, Gani R. Generic mathematical programming formulation and solution for computer-aided molecular design. *Comput Chem Eng*. 2015;78:79-84.
23. Ng LY, Chong FK, Chemmangattuvalappil NG. Challenges and opportunities in computer-aided molecular design. *Comput Chem Eng*. 2015;81:115-129.
24. Austin ND, Sahinidis NV, Trahan DW. Computer-aided molecular design: an introduction and review of tools, applications, and solution techniques. *Chem Eng Res Des*. 2016;116:2-26.
25. Alshehri AS, Gani R, You F. Deep learning and knowledge-based methods for computer-aided molecular design—toward a unified approach: state-of-the-art and future directions. *Comput Chem Eng*. 2020;141:107005.
26. Samudra AP, Sahinidis NV. Optimization-based framework for computer-aided molecular design. *AIChE J*. 2013;59(10):3686-3701.
27. Harper PM, Gani R, Kolar P, Ishikawa T. Computer-aided molecular design with combined molecular modeling and group contribution. *Fluid Phase Equilibria*. 1999;158–160(6/7):337-347.
28. Douguet D, Munier-Lehmann H, Labesse G, Pochet S. LEA3D: a computer-aided ligand design for structure-based drug design. *J Med Chem*. 2005;48(7):2457-2468.
29. Katritzky AR, Kuanar M, Slavov S, et al. Quantitative correlation of physical and chemical properties with chemical structure: utility for prediction. *Chem Rev*. 2010;110(10):5714-5789.
30. Mitchell JBO. Machine learning methods in chemoinformatics. *WIREs Comput Mol Sci*. 2014;4(5):468-481.
31. Lo YC, Rensi SE, Torng W, Altman RB. Machine learning in chemoinformatics and drug discovery. *Drug Discov Today*. 2018;23(8):1538-1546.
32. Muratov EN, Bajorath J, Sheridan RP, et al. QSAR without borders. *Chem Soc Rev*. 2020;49(11):3525-3564.
33. Karunanithi AT, Mehrkesh A. Computer-aided design of tailor-made ionic liquids. *AIChE J*. 2013;59(12):4627-4640.
34. Austin ND, Sahinidis NV, Trahan DW. A COSMO-based approach to computer-aided mixture design. *Chem Eng Sci*. 2017;159(2):93-105.
35. Austin ND, Sahinidis NV, Konstantinov IA, Trahan DW. COSMO-based computer-aided molecular/mixture design: a focus on reaction solvents. *AIChE J*. 2018;64(1):104-122.
36. Liu Q, Zhang L, Liu L, et al. OptCAMD: an optimization-based framework and tool for molecular and mixture product design. *Comput Chem Eng*. 2019;124(1):285-301.
37. Lampe M, Stavrou M, Schilling J, Sauer E, Gross J, Bardow A. Computer-aided molecular design in the continuous-molecular targeting framework using group-contribution PC-SAFT. *Comput Chem Eng*. 2015;81(4):278-287.
38. Schilling J, Gross J, Bardow A. Integrated design of ORC process and working fluid using process flowsheeting software and PC-SAFT. *Energy Procedia*. 2017;129(10):129-136.
39. Hechinger M, Voll A, Marquardt W. Towards an integrated design of biofuels and their production pathways. *Comput Chem Eng*. 2010;34(12):1909-1918.
40. Hoppe F, Heuser B, Thewes M, et al. Tailor-made fuels for future engine concepts. *Int J Engine Res*. 2016;17(1):16-27.
41. Whitmore LS, Davis RW, McCormick RL, et al. BioCompoundML: a general biofuel property screening tool for biological molecules using random forest classifiers. *Energy Fuel*. 2016;30(10):8410-8418.
42. McCormick RL, Fioroni G, Fouts L, et al. Selection criteria and screening of potential biomass-derived streams as fuel blendstocks for advanced spark-ignition engines. *SAE Int J Fuels Lubr*. 2017;10(2):442-460.
43. Lunderman S, Fioroni GM, McCormick RL, Nimlos MR, Rahimi MJ, Grout RW. Screening fuels for autoignition with small-volume experiments and gaussian process classification. *Energy Fuel*. 2018;32(9):9581-9591.
44. Dahmen M, Hechinger M, Villeda JV, Marquardt W. Towards model-based identification of biofuels for compression ignition engines. *SAE Int J Fuels Lubr*. 2012;5(3):990-1003.
45. Cai G, Liu Z, Zhang L. *Transformation Rule-Based Molecular Evolution for Automatic Gasoline Molecule Design*. Authorea preprint; 2021 doi: 10.22541/au.163976199.99452600/v1
46. Elton DC, Boukouvalas Z, Fuge MD, Chung PW. Deep learning for molecular design—a review of the state of the art. *Mol Syst Des Eng*. 2019;4(4):828-849.
47. Weininger DS. A chemical language and information system. 1. Introduction to methodology and encoding rules. *J Chem Inf Model*. 1988;28(1):31-36.
48. Faez F, Ommi Y, Baghshah MS, Rabiee HR. Deep graph generators: a survey. *IEEE Access*. 2021;9:106675-106702.
49. Winter R, Montanari F, Noé F, Clevert DA. Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chem Sci*. 2019;10(6):1692-1701.
50. Sanchez-Lengeling B, Aspuru-Guzik A. Inverse molecular design using machine learning: generative models for matter engineering. *Science*. 2018;361(6400):360-365.
51. Jin W, Barzilay R, Jaakkola T. Junction tree variational autoencoder for molecular graph generation. *Proceedings of the 35th International Conference on Machine Learning. Vol. 80 of Proceedings of Machine Learning Research*. PMLR; 2018:2323-2332.
52. Kajino H. Molecular hypergraph grammar with its application to molecular optimization. *Proceedings of the 36th International Conference on Machine Learning. Vol. 97 of Proceedings of Machine Learning Research*. PMLR; 2019:3183-3191.
53. Guimaraes GL, Sanchez-Lengeling B, Outeiral C, Farias PLC, Aspuru-Guzik A. *Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models*; 2018. arXiv preprint arXiv:1705.10843v3.
54. De Cao N, Kipf T. *MolGAN: An Implicit Generative Model for Small Molecular Graphs*; 2018. arXiv preprint arXiv:1805.11973.
55. You J, Liu B, Ying R, Pande V, Leskovec J. Graph convolutional policy network for goal-directed molecular graph generation. *Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18*. Curran Associates; 2018:6412-6422.
56. Zhou Z, Kearnes S, Li L, Zare RN, Riley P. Optimization of molecules via deep reinforcement learning. *Sci Rep*. 2019;9(1):10752.
57. Coley CW, Eyke NS, Jensen KF. Autonomous discovery in the chemical sciences. Part I: Progress. *Angew Chem Int Ed Engl*. 2020;59(51):22858-22893.
58. Sperduti A, Starita A. Supervised neural networks for the classification of structures. *IEEE Trans Neural Netw*. 1997;8(3):714-735.
59. Gori M, Monfardini G, Scarselli F. A new model for earning in graph domains. *Proc Int Joint Conf Neural Netw*. 2005;2:729-734.
60. Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G. Computational capabilities of graph neural networks. *IEEE Trans Neural Netw*. 2009;20(1):81-102.
61. Xia X, Hu J, Wang Y, Zhang L, Liu Z. Graph-based generative models for de Novo drug design. *Drug Discov Today Technol*. 2019;32–33:45-53.
62. Xiong J, Xiong Z, Chen K, Jiang H, Zheng M. Graph neural networks for automated de novo drug design. *Drug Discov Today*. 2021;26(6):1382-1393.
63. Gaudelet T, Day B, Jamasb AR, et al. Utilizing graph machine learning within drug discovery and development. *Brief Bioinform*. 2021;22(6). https://doi.org/10.1093/bib/bbab159

64. Polykovskiy D, Zhebrak A, Sanchez-Lengeling B, et al. Molecular sets (MOSES): a benchmarking platform for molecular generation models. *Front Pharmacol*. 2020;11:565644.
65. Brown N, Fiscato M, Segler MHS, Vaucher AC. GuacaMol: benchmarking models for de novo molecular design. *J Chem Inf Model*. 2019;59(3):1096-1108.
66. Liu R, Liu R, Liu Y, Wang L, Zhang X, Li G. Design of fuel molecules based on variational autoencoder. *Fuel*. 2022;316(2):123426.
67. Rittig JG, Ritzert M, Schweidtmann AM, et al. *Open-Source Graph Machine Learning for Design of High-Octane Fuels*; 2022. Accessed June 3, 2022. https://git.rwth-aachen.de/avt-svt/public/graph_ML_fuel_design
68. Schweidtmann AM, Rittig JG, König A, Grohe M, Mitsos A, Dahmen M. Graph neural networks for prediction of fuel ignition quality. *Energy Fuel*. 2020;34(9):11395-11407.
69. Winter R, Montanari F, Steffen A, Briem H, Noé F, Clevert DA. Efficient multi-objective molecular optimization in a continuous latent space. *Chem Sci*. 2019;10(34):8016-8024.
70. Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs. *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17*. Curran Associates; 2017:1025-1035.
71. Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS. A comprehensive survey on graph neural networks. *IEEE Trans Neural Netw Learn Syst*. 2021;32(1):4-24.
72. Bronstein MM, Bruna J, Cohen T, Veličković P. *Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges*; 2021. arXiv preprint arXiv:2104.13478v2.
73. Atz K, Grisoni F, Schneider G. *Geometric Deep Learning on Molecular Representations*; 2021. arXiv preprint arXiv:2107.12375v4.
74. Kadurin A, Nikolenko S, Khrabrov K, Aliper A, Zhavoronkov A. DruGAN: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico. *Mol Pharm*. 2017;14(9):3098-3104.
75. Gómez-Bombarelli R, Wei JN, Duvenaud D, et al. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Cent Sci*. 2018;4(2):268-276.
76. Krenn M, Häse F, Nigam AK, Friederich P, Aspuru-Guzik A. Self-referencing embedded strings (SELFIES): a 100% robust molecular string representation. *Mach Learn Sci Technol*. 2020;1(4):045024.
77. Blaschke T, Olivecrona M, Engkvist O, Bajorath J, Chen H. Application of generative autoencoder in De novo molecular design. *Mol Inform*. 2018;37(1–2):1700123.
78. Lim J, Ryu S, Kim JW, Kim WY. Molecular generative model based on conditional variational autoencoder for de novo molecular design. *J Chem*. 2018;10(1):31.
79. Bjerrum EJ, Sattarov B. Improving chemical autoencoder latent space and molecular de novo generation diversity with heteroencoders. *Biomolecules*. 2018;8(4):1-17.
80. Prykhodko O, Johansson SV, Kotsias PC, et al. A de novo molecular generation method using latent vector based generative adversarial network. *J Chem*. 2019;11(1):1-13.
81. Griffiths RR, Hernández-Lobato JM. Constrained Bayesian optimization for automatic chemical design using variational autoencoders. *Chem Sci*. 2020;11(2):577-586.
82. Mercado R, Rastemo T, Lindelöf E, et al. Graph networks for molecular design. *Mach Learn Sci Technol*. 2021;2(2):025023.
83. Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE. Neural message passing for quantum chemistry. *Proceedings of the 34th International Conference on Machine Learning. Vol. 70 of Proceedings of Machine Learning Research*. PMLR; 2017:1263-1272.
84. Coley CW, Barzilay R, Green WH, Jaakkola TS, Jensen KF. Convolutional embedding of attributed molecular graphs for physical property prediction. *J Chem Inf Model*. 2017;57(8):1757-1772.
85. Velickovic P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. Paper presented at: *6th International Conference on Learning Representations, ICLR 2018*, Vancouver, BC, Canada, April 30–May 3, 2018, Conference Track Proceedings; 2018.
86. Zhou J, Cui G, Hu S, et al. Graph neural networks: a review of methods and applications. *AI Open*. 2020;1(1):57-81.
87. Zhang Z, Cui P, Zhu W. Deep learning on graphs: a survey. *IEEE Trans Knowl Data Eng*. 2022;34(1):249-270.
88. Xu K, Hu W, Leskovec J, Jegelka S. *How Powerful are Graph Neural Networks?* 2018. arXiv preprint arXiv:1810.00826v3.
89. Schütt KT, Sauceda HE, Kindermans PJ, Tkatchenko A, Müller KR. SchNet—a deep learning architecture for molecules and materials. *J Chem Phys*. 2018;148(24):1-11.
90. Unke OT, Meuwly M. PhysNet: a neural network for predicting energies, forces, dipole moments, and partial charges. *J Chem Theory Comput*. 2019;15(6):3678-3693.
91. Klicpera J, Groß J, Günnemann S. *Directional Message Passing for Molecular Graphs*; 2020. arXiv Preprint arXiv:2003.03123.
92. Zhang S, Liu Y, Xie L. Molecular Mechanics-Driven Graph Neural Network with Multiplex Graph for Molecular Structures; 2020. arXiv preprint arXiv:2011.07457.
93. Morris C, Ritzert M, Fey M, et al. Weisfeiler and leman go neural: higher-order graph neural networks. *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019, and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence*. Vol 33. EAAI; 2019:4602-4609.
94. Flam-Shepherd D, Wu TC, Friederich P, Aspuru-Guzik A. Neural message passing on high order paths. *Mach Learn Sci Technol*. 2021;2(4):045009.
95. Yang K, Swanson K, Jin W, et al. Analyzing learned molecular representations for property prediction. *J Chem Inf Model*. 2019;59(8):3370-3388.
96. Schweidtmann AM, Weber JM, Wende C, Netze L, Mitsos A. Obey validity limits of data-driven models through topological data analysis and one-class classification. *Optim Eng*. 2022;23(2):855-876.
97. Kusner MJ, Paige B, Hernández-Lobato JM. Grammar variational autoencoder. *Proceedings of the 34th International Conference on Machine Learning. Vol. 70 of Proceedings of Machine Learning Research*. PMLR; 2017:1945-1954.
98. Dai H, Tian Y, Dai B, Skiena S, Song L. *Syntax-Directed Variational Autoencoder for Structured Data*; 2018. arXiv preprint arXiv:1802.08786.
99. Breiman L. Bagging predictors. *Mach Learn*. 1996;24(2):123-140.
100. Breiman L. Stacked regressions. *Mach Learn*. 1996;24(1):49-64.
101. Caruana R. Multitask learning. *Mach Learn*. 1997;28(1):41-75.
102. Ruder S. *An Overview of Multi-Task Learning in Deep Neural Networks*; 2017. arXiv preprint arXiv:1706.05098.
103. vom Lehn F, Brosius B, Broda R, Cai L, Pitsch H. Using machine learning with target-specific feature sets for structure–property relationship modeling of octane numbers and octane sensitivity. *Fuel*. 2020;281:118772.
104. Shahriari B, Swersky K, Wang Z, Adams RP, de Freitas N. Taking the human out of the loop: a review of Bayesian optimization. *Proc IEEE*. 2016;104(1):148-175.
105. Schweidtmann AM, Clayton AD, Holmes N, Bradford E, Bourne RA, Lapkin AA. Machine learning meets continuous flow chemistry: automated optimization towards the Pareto front of multiple objectives. *Chem Eng J*. 2018;352(1–9):277-282.
106. Felton KC, Rittig JG, Lapkin AA. Summit: benchmarking machine learning methods for reaction optimisation. *Chem Methods*. 2021;1(2):116-122.
107. Häse F, Aldeghi M, Hickman RJ, et al. Olympus: a benchmarking framework for noisy optimization and experiment planning. *Mach Learn Sci Technol*. 2021;2(3):035021.
108. Holland JH. Genetic algorithms. *Sci Am*. 1992;267(1):66-73.

109. Whitley D. A genetic algorithm tutorial. *Stat Comput*. 1994;4(2):65-85.
110. Snoek J, Rippel O, Swersky K, et al. Scalable Bayesian optimization using deep neural networks. In: Bach F, Blei D, eds. *Proceedings of the 32nd International Conference on Machine Learning. Vol. 37 of Proceedings of Machine Learning Research*. PMLR; 2015:2171-2180.
111. Wang Z, Hutter F, Zoghi M, Matheson D, de Feitas N. Bayesian optimization in a billion dimensions via random embeddings. *J Artif Intel Res*. 2016;55:361-387.
112. Wang Z, Gehring C, Kohli P, Jegelka S. Batched large-scale Bayesian optimization in high-dimensional spaces. In: Storkey A, Perez-Cruz F, eds. *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics. Vol. 84 of Proceedings of Machine Learning Research*. PMLR; 2018:745-754.
113. Kirschner J, Mutny M, Hiller N, Ischebeck R, Krause A. Adaptive and safe Bayesian optimization in high dimensions via one-dimensional subspaces. In: Chaudhuri K, Salakhutdinov R, eds. *Proceedings of the 36th International Conference on Machine Learning. Vol. 97 of Proceedings of Machine Learning Research*. PMLR; 2019:3429-3438.
114. Tropsha A, Gramatica P, Gombar V. The importance of being Earnest: validation is the absolute essential for successful application and interpretation of QSPR models. *QSAR Comb Sci*. 2003;22(1):69-77.
115. Jaworska J, Nikolova-Jeliazkova N, Aldenberg T. QSAR applicabilty domain estimation by projection of the training set descriptor space: a review. *Altern Lab Anim*. 2005;33(5):445-459.
116. Gramatica P. Principles of QSAR models validation: internal and external. *QSAR Comb Sci*. 2007;26(5):694-701.
117. Weaver S, Gleeson MP. The importance of the domain of applicability in QSAR modeling. *J Mol Graph Model*. 2008;26(8):1315-1326.
118. Sheridan RP, Feuston BP, Maiorov VN, Kearsley SK. Similarity to molecules in the training set is a good discriminator for prediction accuracy in QSAR. *J Chem Inf Comput Sci*. 2004;44(6):1912-1928.
119. Hirschfeld L, Swanson K, Yang K, Barzilay R, Coley CW. Uncertainty quantification using neural networks for molecular property prediction. *J Chem Inf Model*. 2020;60(8):3770-3780.
120. Soleimany AP, Amini A, Goldman S, Rus D, Bhatia SN, Coley CW. Evidential deep learning for guided molecular property prediction and discovery. *ACS Cent Sci*. 2021;7(8):1356-1367.
121. Nigam A, Pollice R, Hurley MFD, et al. Assigning confidence to molecular property prediction. *Expert Opin Drug Discovery*. 2021;16(9):1009-1023.
122. Landrum G. *RDKit: Open-Source Cheminformatics Software*. Accessed September 8, 2021. http://www.rdkit.org
123. The Luigi Authors. *Luigi (Python Package)*; 2012. Accessed September 8, 2021. https://github.com/spotify/luigi
124. Ruddigkeit L, van Deursen R, Blum LC, Reymond JL. Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. *J Chem Inf Model*. 2012;52(11):2864-2875.
125. Ramakrishnan R, Dral PO, Rupp M, von Lilienfeld OA. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*. 2014;1(1):140022.
126. Fey M, Lenssen JE. *Fast Graph Representation Learning with PyTorch Geometric*; 2019. arXiv Preprint arXiv:1903.02428v3.
127. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: machine learning in Python. *J Mach Learn Res*. 2011;12:2825-2830.
128. The GPyOpt Authors. *GPyOpt: A Bayesian Optimization Framework in Python*; 2016. Accessed September 8, 2021. http://github.com/SheffieldML/GPyOpt
129. Schweidtmann AM, Mitsos A. Deterministic global optimization with artificial neural networks embedded. *J Optim Theory Appl*. 2019;180(3):925-948.
130. Schweidtmann AM, Bongartz D, Grothe D, et al. Deterministic global optimization with Gaussian processes embedded. *Math Program Comput*. 2021;13(3):553-581.
131. Liu DC, Nocedal J. On the limited memory BFGS method for large scale optimization. *Math Program*. 1989;45(1–3):503-528.
132. Solgi RM. *geneticalgorithm (Python package)*; 2020. Accessed September 8, 2021. https://github.com/rmsolgi/geneticalgorithm
133. Schweidtmann AM, Rittig JG, König A, Grohe M, Mitsos A, Dahmen M. Supporting information: graph neural networks for prediction of fuel ignition quality. *Energy Fuel*. 2020;34(9):11395-11407.
134. König A, Neidhardt L, Viell J, Mitsos A, Dahmen M. Integrated design of processes and products: optimal renewable fuels. *Comput Chem Eng*. 2020;134:106712.
135. Derfer JM, Boord CE, Burk FC, et al. *Knocking Characteristics of Pure Hydrocarbons*. ASTM International; 1958.
136. Merck KGaA. *Sigma-Aldrich [online]*. Accessed September 8, 2021. https://www.sigmaaldrich.com
137. Royal Society of Chemistry. *Chemspider* [online]. Accessed September 8, 2021. http://www.chemspider.com
138. Leppard WR. The autoignition chemistries of octane-enhancing ethers and cyclic ethers: a motored engine study. *SAE Trans*. 1991;100:589-604.
139. Kubic WL, Jenkins RW, Moore CM, Semelsberger TA, Sutton AD. Artificial neural network based group contribution method for estimating cetane and octane numbers of hydrocarbons and oxygenated organic compounds. *Ind Eng Chem Res*. 2017;56(42):12236-12245.
140. Chemspace US Inc. *Chemspace* [online]. Accessed September 8, 2021. https://chem-space.com
141. Enamine Ltd. *EnamineStore* [online]. Accessed September 8, 2021. https://www.enaminestore.com
142. SynQuest Laboratories, Inc. *SynQuest Labs* [online]. Accessed September 8, 2021. http://synquestlabs.com
143. Magnusson R, Nilsson C. The influence of oxygenated fuels on emissions of aldehydes and ketones from a two-stroke spark ignition engine. *Fuel*. 2011;90(3):1145-1154.
144. Hoppe F, Burke U, Thewes M, Heufer A, Kremer F, Pischinger S. Tailor-made fuels from biomass: potentials of 2-butanone and 2-methylfuran in direct injection spark ignition engines. *Fuel*. 2016;167(3):106-117.
145. Hechinger M. Model-Based Identification of Promising Biofuel Candidates for Spark-Ignited Engines [PhD thesis]; 2014. RWTH Aachen University, Germany.
146. Yanowitz J, Ratcliff MA, McCormick RL, Taylor JD, Murphy MJ. *Compendium of Experimental Cetane Numbers* (Technical Report NREL/TP-5400-67585); 2017. National Renewable Energy Laboratory (NREL), Golden, CO.
147. Perez PL, Boehman AL. Experimental investigation of the autoignition behavior of surrogate gasoline fuels in a constant-volume combustion bomb apparatus and its relevance to HCCI combustion. *Energy Fuel*. 2012;26(10):6106-6117.
148. vom Lehn F, Cai L, Tripathi R, Broda R, Pitsch H. A property database of fuel compounds with emphasis on spark-ignition engine applications. *Appl Energy Combust Sci*. 2021;5(12):100018.
149. Naegeli DW, Yost DM, Moulton DS, Owens EC, Chui GK. The measurement of octane numbers for methanol and reference fuels blends. *SAE Trans*. 1989;98:712-722.
150. Yanowitz J, Christensen E, McCormick RL. *Utilization of Renewable Oxygenates as Gasoline Blending Components* (Technical Report NREL/TP-5400-50791); 2011. National Renewable Energy Laboratory (NREL), Golden, CO.
151. Petroleum products—determination of knock characteristics of motor fuels—research method (ISO 5164:2014); 2014. German version.

152. Petroleum products—determination of knock characteristics of motor and aviation fuels—motor method (ISO 5163:2014); 2014. German version.
153. Lee C, Vranckx S, Heufer KA, et al. On the chemical kinetics of ethanol oxidation: shock tube, rapid compression machine and detailed modeling study. *Z Phys Chem*. 2012;226(1):1-28.
154. Ramalingam A, Zhang K, Dhongde A, et al. An RCM experimental and modeling study on CH_4 and CH_4/C_2H_6 oxidation at pressures up to 160 bar. *Fuel*. 2017;206:325-333.
155. Weisser GA. Modelling of Combustion and Nitric Oxide Formation for Medium-Speed DI Diesel Engines: A Comparative Evaluation of Zero- and Three-Dimensional Approaches [PhD thesis]; 2001. ETH Zurich, Switzerland.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Rittig JG, Ritzert M, Schweidtmann AM, et al. Graph machine learning for design of high-octane fuels. *AIChE J*. 2023;69(4):e17971. doi:10.1002/aic.17971