Graph Neural Networks for Prediction of Fuel Ignition Quality

Artur M. Schweidtmann,[†] Jan G. Rittig,[†] Andrea König,[†] Martin Grohe,[‡] Alexander Mitsos,[¶,†,§] and Manuel Dahmen[*,§]

[†]*RWTH Aachen University, Process Systems Engineering (AVT.SVT), Aachen 52074, Germany*

[‡]*RWTH Aachen University, Lehrstuhl Informatik 7, Aachen 52074, Germany*

[¶]*JARA-ENERGY, Aachen 52056, Germany*

[§]*Forschungszentrum Jülich GmbH, Institute for Energy and Climate Research IEK-10: Energy Systems Engineering, Jülich 52425, Germany*

E-mail: m.dahmen@fz-juelich.de

Abstract

Prediction of combustion-related properties of (oxygenated) hydrocarbons is an important and challenging task for which quantitative structure-property relationship (QSPR) models are frequently employed. Recently, a machine learning method, graph neural networks (GNNs), has shown promising results for the prediction of structure-property relationships. GNNs utilize a graph representation of molecules, where atoms correspond to nodes and bonds to edges containing information about the molecular structure. More specifically, GNNs learn physico-chemical properties as a function of the molecular graph in a supervised learning setup using a backpropagation algorithm. This end-to-end learning approach eliminates the need for selection of molecular descriptors or structural groups, as it learns optimal fingerprints through graph convolutions

and maps the fingerprints to the physico-chemical properties by deep learning. We develop GNN models for predicting three fuel ignition quality indicators, i.e., the derived cetane number (DCN), the research octane number (RON), and the motor octane number (MON), of oxygenated and non-oxygenated hydrocarbons. In light of limited experimental data in the order of hundreds, we propose a combination of multi-task learning, transfer learning, and ensemble learning. The results show competitive performance of the proposed GNN approach compared to state-of-the-art QSPR models making it a promising field for future research. The prediction tool is available via a web front-end at `www.avt.rwth-aachen.de/gnn`.

1 Introduction

The worldwide increase in CO_2 emissions and the depletion of fossil resources call for the development of renewable fuels. A wide range of non-oxygenated and oxygenated hydrocarbons derived from renewable resources such as biomass has been investigated as pure-component fuel or blend components for use in internal combustion engines.[1–7] To determine how suited a molecule is for a fuel application, combustion-related properties need to be evaluated. The cetane number (CN) or derived cetane number (DCN), the research octane number (RON), and the motor octane number (MON) are commonly employed to characterize the auto-ignition/knocking behavior of a particular fuel. Fuels with a high RON (MON) exhibit low knocking tendency and are therefore suitable for spark-ignition (SI) engines, whereas fuels with a high (D)CN (approx. above 40) exhibit a short ignition delay which is required in compression-ignition (CI) engines.[4,8,9] Experimental RON, MON, and (D)CN values are available for a range of different fuel molecules,[10–12] however, for many interesting molecules such data is not readily available. For these molecules predictive models are required that enable rapid estimation of fuel ignition quality.[4,13]

In the past decades, several models have been developed to predict (D)CN[12–28] and RON/MON[12,16,29–32] of (oxygenated) hydrocarbons by utilizing quantitative structure-property

relationship (QSPR) modeling. In QSPR, the modeling process can be broken down into two steps: First, QSPR models introduce molecular descriptors $\boldsymbol{D} = [d_1, d_2, ..., d_n]^T$ that depend on the structure of a molecule m. Second, a regression model $F(\boldsymbol{D}) : \boldsymbol{D} \mapsto \hat{p}$ is fitted that predicts a property $\hat{p}$ as a function of $\boldsymbol{D}$.[33] The regression model is either linear or nonlinear, depending on the QSPR. Group contribution methods[34–36] are a particular type of QSPR model where the molecular descriptors $\boldsymbol{D}$ are structural group counts, i.e., the number of occurrences of basic functional groups, e.g., methyl ($-CH_3-$) or methylene ($-CH_2-$), in a molecule m.

QSPR models for DCN, RON, and MON differ in the way they encode the molecular structure. Various descriptors have been used including structural group counts (e.g., in[12–14,21,29–32]), the number of aromatic bonds (e.g., in[13,27]), and topological indices, such as the Wiener Index[37] or branching indices (e.g., in[20,26,31]). Previous models have also used a variety of techniques for the regression step, e.g., linear or nonlinear regression,[13,14,21,26,29,30] or artificial neural networks (ANNs).[12,17–19,22,24,27,30–32] Development of QSPR models, however, highly depends on the choice of informative descriptors, a selection process that requires domain knowledge and intuition.

Deep learning allows to learn representations of data with multiple abstraction levels. This has shown remarkable success for end-to-end learning in various domains surpassing previously performed manual feature selection.[38] In particular, graph neural networks (GNNs)[39,40] have recently shown promising results for the prediction of structure-property relationships of molecules.[41–46] GNNs utilize graph representations of molecules, where atoms correspond to nodes and bonds correspond to edges. For each atom, its local environment is learned by graph convolutions. These atom environments are then combined into a molecular fingerprint by applying pooling functions.[47] Finally, an ANN maps the fingerprint to the molecular property of interest. Since the graph convolutions and pooling functions are differentiable, the full model can be trained with the backpropagation algorithm. In contrast to QSPRs, the processes of choosing molecular descriptors and performing property regres-

sion are thus merged into a simultaneous training step. This enables supervised end-to-end training from the molecular graph to the property. In particular, the molecular fingerprints adapt during training and learn molecular structure information that is important for the property of interest.

We propose the first GNN model for the prediction of DCN, RON, and MON. The model is trained on literature data and is applicable to a wide range of oxygenated and non-oxygenated hydrocarbons. The GNN architecture includes state-of-the-art higher-order molecular graph features[46] and is provided open-source.[48] Furthermore, we provide a web front-end that can be easily accessed online to make predictions. The web front-end takes SMILES strings as input and automatically predicts DCN, RON, and MON (`www.avt.rwth-aachen.de/gnn`).

One of the main challenges in GNN training in the context of fuel ignition quality is the limited availability of training data. To mitigate this issue, we propose three model extensions that reduce data requirements while achieving competitive prediction accuracy: First, we propose a multi-task learning approach where DCN, RON, and MON are trained jointly. This approach shares the graph convolutions and the molecular fingerprint among all prediction tasks and thus takes advantage of correlations between DCN, RON, and MON data sets. Second, we perform a transfer learning approach that utilizes a broader data set from different (D)CN measurement techniques for pre-training of our final model. Third, we perform ensemble learning averaging out random model variations.

The remainder of this paper is structured as follows: In Section 2, we provide a general background on graph representations of molecules and GNNs. Then, we briefly describe the databases of this work in Section 3. Afterwards, we propose the GNN architecture in Section 4. Furthermore, we briefly describe the considered learning methods: multi-task learning, transfer learning, and ensemble learning. In Section 5, we present the results, discuss the different learning methods, and compare our GNN model to state-of-the-art QSPR models. Finally, we conclude our findings and show potentials for future research (Section 6).

2 Fundamentals of graph neural networks

In this section, we present a brief background on graph representations for molecules and GNNs. Furthermore, we introduce the concept of higher-order GNNs that is fundamental to our modeling approach.

2.1 Molecular graphs

Any molecule can be represented as a molecular graph where nodes $w, v \in V$ correspond to atoms. Edges $e_{vw} \in E$ correspond to bonds between two atoms.[49,50] Furthermore, a feature vector is assigned to each node and each edge that includes information about atom types, e.g., C atom, and bond types, e.g., double bond. The node feature vectors $\boldsymbol{f}^V(v)$ can contain additional atom information such as orbital hybridization. To reduce the size of molecular graphs, hydrogen atoms can be implicitly included in the feature vectors of nodes of heavy atoms by using a hydrogen count, resulting in an H-depleted molecular graph.[51] Similarly, bond feature vectors $\boldsymbol{f}^E(e_{vw})$ can provide additional information, e.g., on ring structures. GNNs operate on graph structures, i.e., they take the molecular graph and its feature vectors as inputs.

2.2 Graph neural networks

As illustrated in Figure 1, GNNs have two main phases: (i) the message passing phase and (ii) the readout phase.[43] In the message-passing phase, graph convolutional layers are commonly applied with a large variety of layer structures existing.[41,43,47,52,53]

Figure 2 illustrates the basic concept of graph convolutional layers. The overall goal of the graph convolutional layer is to combine node information of a considered node (here #2 in red) with node information of its neighbors (here #1,3,4 in yellow) and bond information (green). To this end, node state vectors and edge state vectors are combined through message and update functions as explained in more detail in the following paragraph. The result of

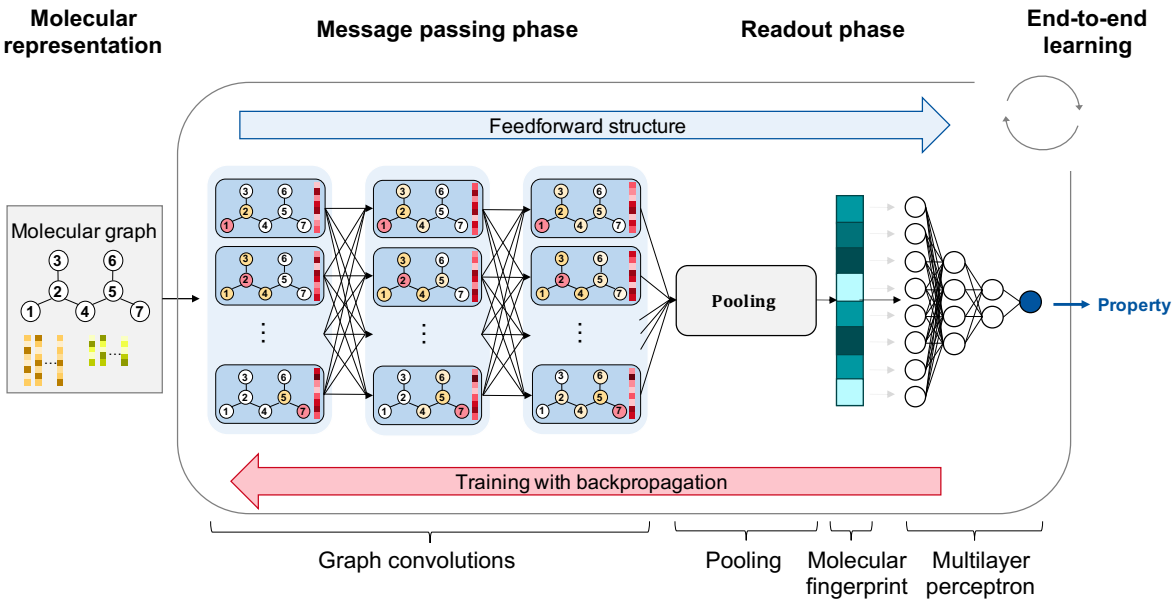


Figure 1: Overview of a graph neural network model for property prediction.

the graph convolutional layer is an updated node state vector of the considered node.

Within a graph convolutional layer, information about a node's neighborhood $N(v) = \{w \mid e_{vw} \in E, w \neq v\}$ is aggregated and passed to the respective node.[43] The message $\boldsymbol{m}_v^l$ is passed along edges to a node by applying a message function $\boldsymbol{M}_l$, i.e.,

$$\boldsymbol{m}_v^l = \sum_{w \in N(v)} \boldsymbol{M}_l \left(\boldsymbol{h}_w^{l-1}, \boldsymbol{f}^E(e_{vw}) \right) \ , \tag{1}$$

where $\boldsymbol{h}_w^{l-1}$ denotes the hidden state of a neighbor in layer $l-1$. The 0-th hidden state vector is initialized with the input feature vector of a node, i.e., $\boldsymbol{h}_v^0 = \boldsymbol{f}^V(v)$. The message function $\boldsymbol{M}_l$ depends on the previous hidden states of the neighbors $\boldsymbol{h}_w^{l-1}$ and the features of the respective edges $\boldsymbol{f}^E(e_{vw})$, with the dimension of the hidden state vector for layers $l > 0$ being a hyperparameter. The hidden state of the considered node $\boldsymbol{h}_v^{l-1}$ is then updated to $\boldsymbol{h}_v^l$ by an update function $\boldsymbol{U}_l$ that combines its hidden state from the previous layer with the received message containing information of its neighbors:

$$\boldsymbol{h}_v^l = \boldsymbol{U}_l \left(\boldsymbol{h}_v^{l-1}, \boldsymbol{m}_v^l \right) \tag{2}$$

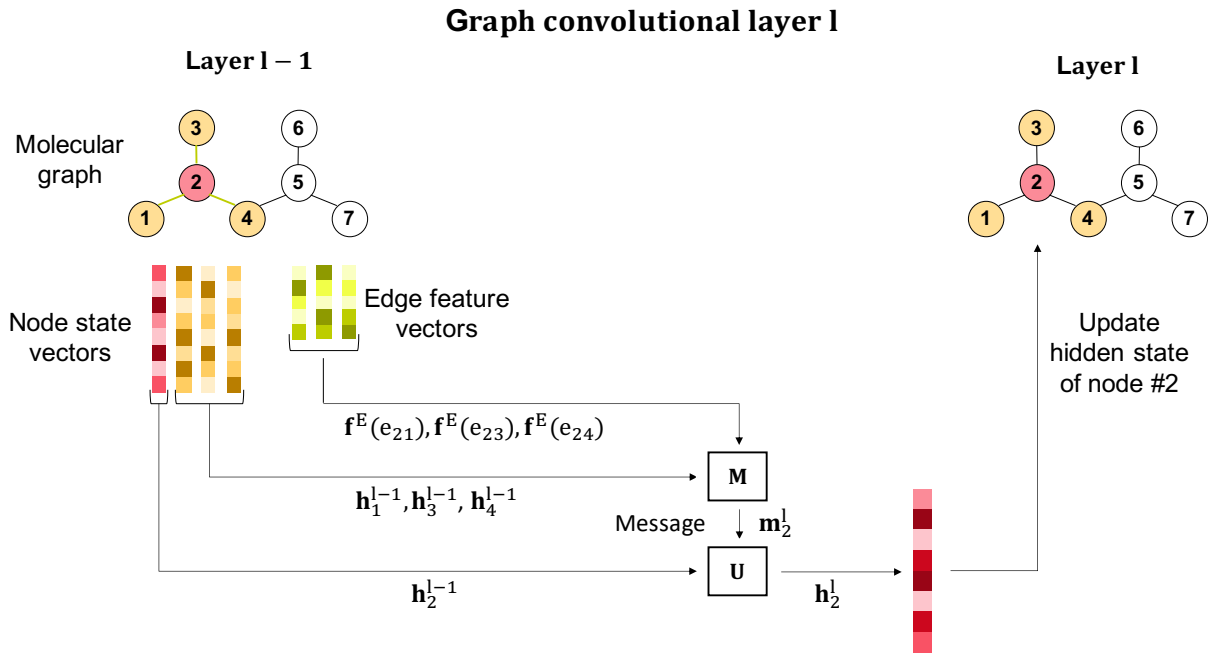


Figure 2: Illustration of a graph convolutional layer. We consider the update of the node state vector of the considered atom (#2 in red). The atom information is given as node state vectors of considered node and its neighbors (#1, 3, 4 in yellow). Furthermore, bond information is given as edge feature vectors (green). The message function $\boldsymbol{M}$ generates the message and the update function $\boldsymbol{U}$ computes the update for new state vector of the considered node.

The message passing and updating is repeated for a fixed number of iterations which results in multiple graph convolutional layers $l \in \{1, 2, ..., L\}$. After L graph convolutions, the hidden states of nodes $\boldsymbol{h}_v^L$ contain local information of environments with a radius of L nodes.

In the readout phase, the hidden node states of the last convolutional layer are combined into a graph representation vector $\boldsymbol{h}_G$, i.e., molecular fingerprint, by using a pooling function $\boldsymbol{h}_G = \boldsymbol{p}(\boldsymbol{h}_1^L, \boldsymbol{h}_2^L, ..., \boldsymbol{h}_{|v|}^L)$ where $\boldsymbol{p}$ is commonly chosen as the mean, sum, or max function.[52] Finally, the molecular fingerprint vector $\boldsymbol{h}_G$ is utilized for regression of molecular properties of interest, e.g., using a multilayer perceptron (MLP), i.e., $\hat{p} = \mathrm{MLP}(\boldsymbol{h}_G)$.

A strong advantage of this method is that all functions from the molecular graph to the property are explicit and differentiable allowing for supervised training using backpropagation. This enables end-to-end learning of GNNs, whereby the graph convolutions and the molecular fingerprint adapt during training to extract information of the molecular graph

that is relevant for the property to be predicted.[41,43,54]

2.3 Higher-order graph neural networks

Morris et al. have recently extended the message passing to higher-order graph features.[46] Here, the message passing step does not only apply to the initial molecular graph but also to modified higher-dimensional molecular graphs. For these higher-dimensional graphs, the nodes are k-dimensional subsets s of the nodes in the initial graph, $s = \{v_1, ..., v_k\} \in [V]^k = \{U \subseteq V \mid |U| = k\}$ for $k > 1$. Hence, any combination of k nodes from the original graph (atoms in the molecular graph) are combined in a separate node.

The neighborhood of a k-dimensional node s is defined as $N(s) = \{t \in [V]^k \mid |s \cap t| = k - 1\}$ for $k > 1$. This means that two k-dimensional nodes s and t with k atoms each are adjacent to each other if the cut set of the two nodes consists of $k - 1$ atoms. This concept of higher-order neighborhood is illustrated in Figure 3, where we consider a red node and its yellow neighbor in the initial molecular graph (1-GNN) as well as higher-order graphs.

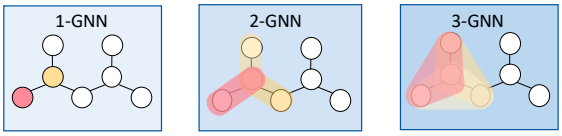


Figure 3: Illustration of the neighborhood in k-dimensional graphs. We highlight the yellow neighbors of a red node.

The message passing phase for the initial molecular graph is referred to as 1-GNN. The message passing phase for higher-dimensional graphs with nodes consisting of k nodes from the original graphs, the structure is called k-GNN.[46] When considering k-dimensional nodes, the initializations of the hidden node states cannot simply be atom types and features, but rather must be combinations of the respective individual atom types and features. At first, the initialization of the hidden node state of a k-dimensional node s contains the isomorphic type $\boldsymbol{f}_{\text{iso}}(s)$.[46] For example, the initial H-depleted molecular graph ($k = 1$) of oxygenated hydrocarbons, the isomorphic type of the node v corresponds to the atom type {{C},{O}}

and is included in the initial feature vector of a node. For the 2-GNN ($k = 2$) of such hydrocarbons, the isomorphic type of the set $s = \{v_1, v_2\}$ corresponds to the 2-set of atom types: {{C,C},{C,O},{O,O}}.

Furthermore, the k-GNN model usually works in a hierarchical manner such that the outputs of the (k-1)-th GNN serve as inputs for the k-th GNN.[46] Therefore, in addition to the isomorphic type, the respective hidden node states of the last graph convolutional layer L of the preceding GNN are combined by a pooling function, e.g., mean function, and used for initializing the hidden state of a k-dimensional node. Note that this does not refer to the pooling of the molecular fingerprints in the readout phase. Hence for the 2-GNN, the hidden node states of a subset $s = \{v_1, v_2\}$ are initialized with the concatenation ($\|$) of the isomorphic type and the averaged hidden node states of the two respective nodes from the 1-GNN, i.e., $\boldsymbol{h}_s^{2,0} = \boldsymbol{f}_{\text{iso}}(s) \,\|\, \text{mean}(\boldsymbol{h}_{v_1}^L, \boldsymbol{h}_{v_2}^L)$.

3 Databases for fuel ignition quality

We collect DCN, RON, and MON data of non-oxygenated and oxygenated hydrocarbons from different literature sources. The number of species per molecular class and fuel ignition quality indicator is shown in Table 1. Note that we provide our full data set in the Supporting Information (SI).

The cetane number (CN), an indicator for CI fuel quality, is determined in a cooperative fuel research (CFR) reference engine.[55] In comparison to the CFR engine, testing methods in a variety of constant-volume combustion chambers (CVCCs) require lower fuel quantities and shorter measurement times, but yield so-called derived cetane numbers (DCNs)[11] instead of true CFR CN. Our objective is to predict DCN values determined by a particular CVCC-based experimental setup, i.e., the ignition quality tester (IQT) which is standardized by the ASTM D6980[56] and widely used to assess diesel fuel ignition quality.[13,57] To this end, we consider IQT-DCN data from the Compendium of Experimental Cetane Numbers provided

by Yanowitz et al.[11] for 236 different species.

Although (D)CN values from non-IQT experiments cannot be expected to closely match IQT-DCN,[13] we utilize such data for a transfer learning approach. Specifically, we consider (D)CN values for 479 species from various measurement setups from the Compendium of Experimental Cetane Numbers.[11] We exclude the test set compounds, use the remaining 447 molecules for pre-training, and subsequently refine the resulting model based on IQT-only data. We provide the different data sets created for model development online.[48]

RON and MON are used to quantify the knocking tendency of a fuel and are suitable ignition quality indicators for SI engines. They are measured according to the ASTM D2699[58] and ASTM D2700[59] standards, respectively. As a database for our model, we take RON (MON) values for 335 (318) species from literature.[2,10,12,32,60–67] For all reported data (RON, MON, DCN), we take average values whenever multiple values have been reported for a single species.

4 Modeling approach

In this section, the GNN model development is described (cf. Figure 4). First, the workflow of the molecular representation is explained in Section 4.1. Then, the basic architecture of the GNN model is outlined in Section 4.2. Finally, the model extensions for multi-task learning (Section 4.3), transfer learning (Section 4.4), and ensemble learning (Section 4.5) are described.

We use PyTorch Geometric,[68] an open-source library for deep learning on graphs in Python. The implementation is adapted from our previous work on k-GNNs.[46] Our open-source code for training as well as the trained models can be retrieved on.[48] Additionally, for more convenient use, the model can be accessed freely via a web front-end (`www.avt.rwth-aachen.de/gnn`) to make predictions.

Table 1: Databases assembled for this work: Number of species per molecular class and fuel ignition quality indicator. Number of species in our test set is given in parentheses. Details can be found in the Supporting Information.

	IQT-DCN	**(D)CN**	**RON**	**MON**
n-alkanes	9 (0)	18 (0)	7 (0)	7 (0)
iso-alkanes	17 (2)	39 (2)	43 (6)	42(6)
cycloalkanes	20 (0)	34 (0)	74 (7)	65 (7)
alkenes	15 (3)	33 (3)	87 (16)	84 (16)
cycloalkenes	10 (0)	12 (0)	22 (2)	22 (2)
alkynes	1 (0)	1 (0)	8 (0)	4 (0)
aromatics	13 (4)	63 (4)	41 (12)	43 (12)
alcohols	21 (3)	38 (3)	14 (1)	13 (1)
cyclic alcohols	2 (0)	2 (0)	0 (0)	0 (0)
aldehydes	7 (1)	7 (1)	0 (0)	0 (0)
ketones	9 (3)	9 (3)	8 (1)	7 (1)
cyclic ketones	5 (1)	5 (1)	2 (0)	2 (0)
ethers	17 (3)	27 (3)	5 (0)	5 (0)
hydrofurans	7 (1)	7 (1)	3 (1)	3 (1)
other cyclic ethers	4 (0)	4 (0)	2 (0)	2 (0)
esters	39 (4)	133 (4)	12 (3)	12 (3)
lactones	4 (1)	4 (1)	1 (0)	1 (0)
furans	5 (2)	5 (2)	3 (2)	3 (2)
acetals	2 (0)	2 (0)	1 (0)	1 (0)
carboxylic acids	0 (0)	5 (0)	0 (0)	0 (0)
more than one type of oxygen functionality	29 (4)	31 (4)	2 (0)	2 (0)
Total	**236 (32)**	**479 (32)**	**335 (51)**	**318 (51)**

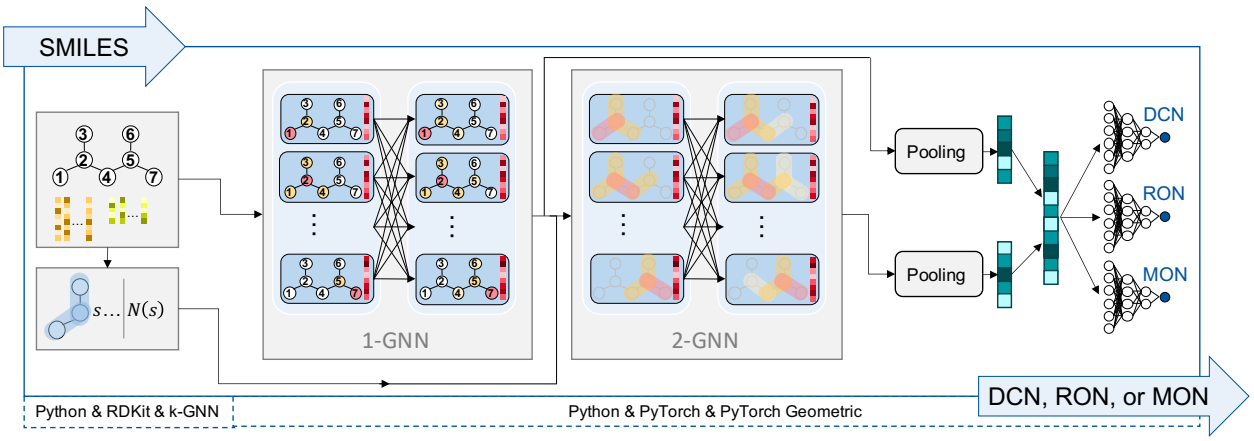


Figure 4: Workflow of the multi-task GNN model for predicting DCN, RON, and MON of hydrocarbons with SMILES strings as input. The model works in a hierarchical manner in which the outputs of the 1-GNN part are used as inputs for the 2-GNN. The outputs of both, the 1-GNN and 2-GNN part, are concatenated and represent the molecular fingerprint. This serves as the input for the three MLP channels predicting the target properties.

4.1 Molecular representation

For generating the representation of molecules that serve as an input to the GNN, SMILES strings[69] are transformed into molecular graphs. Each node and each edge is assigned a feature vector. Each feature is represented as a one-hot encoder with the size of the number of possible values for this feature and a single entry with value one at the index corresponding to the value of the feature. The features are selected according to previous literature[41,43,44] and are shown in Table 2 and 3 for nodes and edges, respectively. We use RDKit[70] for SMILES strings transformation and calculation of features. Note that the data set considered in this work does not include any atoms with sp3d or sp3d2 hybridization. Furthermore, we find that RDKit encodes stereochemistry exclusively by means of the more general E/Z notation instead of the cis/trans notation on our data set. The dimensionality of the hybridization and stereo feature vector could thus be reduced without loss of information.

Table 2: Atomic features used as initial node states, similar to.[41,43,44] All features are implemented as one-hot encoders.

Feature	Description	Dimension
atom type	type of atom (C, O)	2
is in ring	whether the atom is part of a ring	1
is aromatic	whether the atom is part of an aromatic system	1
hybridization	sp, sp2, sp3, sp3d, or sp3d2	5
# bonds	number of bonds the atom is involved in	6
# Hs	number of bonded hydrogen atoms	5

Table 3: Bond features used as edge features, similar to.[41,43,44] All features are implemented as one-hot encoders.

Feature	Description	Dimension
bond type	single, double, triple, or aromatic	4
conjugated	whether the bond is conjugated	1
is in ring	whether the bond is part of a ring	1
stereo	none, any, E/Z, or cis/trans	6

4.2 Model architecture

The proposed GNN model combines our higher-dimensional GNNs[46] with recurrent neural network architectures.[43,71,72] By using higher-dimensional GNNs, higher-order characteristics of a molecular graph can be extracted. The recurrent neural networks allow us to share parameters within the message passing phase of a GNN. We use gated recurrent units (GRUs) as recurrent neural networks, because GRUs avoid the vanishing gradient problem while having fewer parameters than long short-term memories (LSTMs) and thus have the potential to generalize faster on small data sets.[71] As illustrated in Figure 4, we apply two GNN structures in the message passing phase: (i) 1-GNN and (ii) 2-GNN. Thereby, atom environments in a molecule are first examined locally and then interactions between different atom environments are studied.

First, in the 1-GNN, an edge feature network and a GRU explore local atomic environments within the molecular graph. In particular, the updated hidden state in layer l, i.e., $\boldsymbol{h}_v^l$, is computed as

$$\boldsymbol{h}_v^l = \mathrm{GRU}\left(\boldsymbol{h}_v^{l-1}, \sigma\left(\theta_v \cdot \boldsymbol{h}_v^{l-1} + \boldsymbol{m}_v^l\right)\right) \ , \tag{3}$$

where the message $\boldsymbol{m}_v^l$ is given by

$$\boldsymbol{m}_v^l = \sum_{w \in N(v)} \mathrm{ANN}_{\theta_e}\left(\boldsymbol{f}^E(e_{vw})\right) \ \cdot \ \boldsymbol{h}_w^{l-1} \ . \tag{4}$$

Herein, the edge feature network is a feedforward ANN, i.e., ANN_{θ_e}, that maps edge features $\boldsymbol{f}^E$ to a parameter matrix θ_e. Then, the parameter matrix θ_e is multiplied with the hidden states of a node's v neighbors, $\boldsymbol{h}_w^{l-1}$ with $w \in N(v)$, to calculate the message $\boldsymbol{m}$. The message is added to the hidden state of the considered node $\boldsymbol{h}_v^{l-1}$ multiplied with a parameter matrix θ_v. This result is transformed with an activation function σ, here rectified linear unit (ReLU). By applying a GRU, the updated hidden state in layer l, i.e., $\boldsymbol{h}_v^l$, is finally computed. Note that in this work the hidden states are based on nodes, whereas Yang et al.[44] consider hidden states based on edges. The initial hidden states $\boldsymbol{h}_v^0 = \boldsymbol{f}^V(v)$ are mapped to the dimension of the following hidden states by a shallow ANN with ReLU activation.

Secondly, a higher-dimensional message passing process is applied to enable interactions between atom environments (cf. Section 2.3). By combining the final atom representations of the 1-GNN into higher-dimensional nodes on which another message passing phase is applied, long-range effects of atom groups within a molecule can be captured. In this work, we found a 1,2-GNN architecture to have a lower mean absolute error (MAE) compared to that of a 1,2,3-GNN or a simple 1-GNN, thus the 1,2-GNN architecture is used to learn higher-dimensional graph features. Accordingly, we call the hierarchical combination of the 1-GNN and 2-GNN structure 1,2-GNN in the remainder of this work. We update the hidden states in the 2-GNN message passing similarly to the previously described 1-GNN, except that the edge feature network is replaced by a simple parameter matrix θ_2^k as there are no features for edges of the higher-order graph.[46] As the 1-GNN is an input to the 2-GNN, the

1,2-GNN indirectly takes the edge features of the original molecular graph into account.

After the message passing process, the model employs sum pooling for aggregating the hidden node states of the 1-GNN and 2-GNN resulting in two graph representation vectors. Sum pooling is applied, since the nature of the contribution of atoms and bonds to DCN, RON, and MON is expected to be additive, similar to in group contribution models.[12,13] After pooling, the two graph representation vectors are concatenated to the molecular fingerprint, i.e., $\boldsymbol{h}_G = [\boldsymbol{h}_{G_{1-GNN}}^T, \boldsymbol{h}_{G_{2-GNN}}^T]^T$. Finally, the molecular fingerprint is fed into a deep MLP for the prediction of DCN, RON, and MON, $\hat{p} = \mathrm{MLP}(\boldsymbol{h}_G)$.

4.3 Single- and multi-task learning

Having several prediction tasks, machine learning models can be trained in single- or multi-task manner.[73–75] In single-task learning, individual models are trained for each task. In multi-task learning, some representation is shared among the different tasks. For ANNs, this means that weights and bias parameters of hidden layers are shared between multiple tasks, i.e., they have equal values. Besides the shared layers, further individual hidden layers are employed for each task. The shared representation captures general information that is relevant to all tasks.[74] In the individual layers, task-specific information is extracted. In this way, the model learns more general input representations in the first layers compared to single-task models and overfitting can be reduced.[74] This is particularly relevant when the data sets are considerably small. Furthermore, multi-task learning can enable knowledge transfer between different prediction tasks.[75] In previous literature, this has been shown to yield superior results to single-task models in multiple molecular applications.[41,76,77]

In our model, we utilize multi-task learning by sharing the graph convolutional layers to create a general molecular fingerprint on which three individual MLPs (also called channels) are used for predicting DCN, RON, and MON. As cetane and octane numbers are known to correlate negatively,[4,8,12,13,78,79] multi-task learning is particularly promising in this context.

4.4 Transfer learning

Another technique enabling knowledge transfer in machine learning is transfer learning.[80,81] In transfer learning, knowledge learned in one domain is transferred to another domain, i.e., to the target task.[81] One way to perform transfer learning concerns pre-training of ANNs on a (source) task related to the target task. Afterward, the parameters of the pre-trained model are used to initialize parameters of a model trained on the target task data. Thus, transfer learning is particularly relevant for problems where the target data basis is small.

Transfer learning has recently been applied in the context of molecular property prediction with GNNs. For example, Grambow et al. pre-trained GNNs for thermophysical property predictions on large data sets from quantum-mechanical calculations and retrained parts of the GNN on a smaller experimental data set.[82]

We aim to improve our IQT-DCN prediction by transferring information from additional (D)CN data, i.e., from measurement techniques other than IQT (cf. Section 3). Thus, we propose a transfer learning approach, where CN and DCN data from various measurement setups are utilized for pre-training and then models are retrained on IQT-only DCN data.

4.5 Ensemble learning

Ensemble learning is a technique in machine learning where multiple models are trained and utilized for a single prediction task.[83–85] In most applications, several individual models are trained independently on a randomly drawn subset of the training data. Then, the predictions of the individual models are averaged to receive a more accurate prediction. This way, prediction can be improved as random model errors are averaged out. Averaging single model predictions is also known as bootstrap aggregating or bagging.[83] This is particularly relevant for models with low bias and high variance which is the case for complex GNNs. Furthermore, small data sets can lead to high variance.

We train independent GNN models with randomly selected training and validation sets. To ensure an unbiased model comparison, all models share the same independent test set.

While some advanced ensembling techniques apply weights to models, e.g., boosting,[86] we use a standard bagging technique that applies the same weight to all models.

5 Results and discussion

In this section, we first briefly summarize the general training settings (Section 5.1) and hyperparameter selection (Section 5.2). Then, we analyze the prediction accuracy of the proposed model developments: multi-task learning (Section 5.3), transfer learning (Section 5.4), and ensemble learning (Section 5.5). Finally, we compare the proposed model to state-of-the-art QSPR models (Section 5.6).

5.1 General training settings

As described in Section 3, the data set of DCN values extracted from the Compendium of Experimental Cetane Numbers[11] includes DCN measurements of 236 different components measured with the IQT method. We use this high-quality DCN data set and the RON and MON data sets for the training of the single and multi-task models (cf. Section 5.3). As typically done in machine learning, the data sets are standardized to zero mean and standard deviation of one for each target property, i.e., DCN, RON, and MON. Then, the data sets are randomly split into a training (85%) and test (15%) set. The test set is separated from the rest of the data and not used until the final testing of the model. For training the model, an internal validation set (15% of the original data set) is separated randomly from the training data and used for early stopping.

For each data point, the molecular graphs are generated as described in Section 4.1. Then, the model is trained based on the training set. Here, the mean squared error is used as the loss function. During training, the model performance regarding the internal validation set is measured in each epoch. The learning rate is decreased by a factor of 0.8 after every 3 consecutive epochs in which the error on the internal validation set did not

decrease. Training is stopped either after a maximum number of 300 epochs was reached or if the internal validation error did not decrease in the 50 preceding epochs, according to early stopping. The error on the internal validation set is also used for comparison of the different model structures and the selection of hyperparameters. The training and random selection of the internal validation set are repeated 40 times for all models.

5.2 Hyperparameter selection

The proposed GNN model exhibits several hyperparameters that need to be chosen. To identify a suitable model architecture, the following hyperparameters are varied within the given ranges: initial learning rate $\in \{0.0005, 0.001, 0.005\}$, hidden states size $\in \{32, 64, 128\}$, number of graph convolutional layers $\in \{1, 2, 3, 4, 5\}$ for the 1-GNN part and number of graph convolutional layers $\in \{1, 2, 3\}$ for the 2-GNN part, and message passing function $\in$ {without GRU, with GRU}. To this end, we performed an extensive hyperparameter study on a preliminary data set with 40 repetitions for each hyperparameter setting. This preliminary data set is 99.7% identical to our final training data set. The only difference is that we have corrected two mistakes in SMILES strings that occurred in the automated data processing and added one new molecule based on new literature that we found. In the hyperparameter study, we considered the total MAE, i.e., RON, MON, and DCN together, on the validation set averaged over 40 runs to decide on the final architecture and parameters. Based on these results, we use message passing with GRU, two graph convolutional layers in the 1-GNN part, two graph convolutional layers in the 2-GNN part, a hidden states size of 64, and an initial learning rate of 0.001 for the final model. We note that the differences between different hyperparameter settings are often similar to the variance of the MAE. Therefore, the sensitivity of the model performance with respect to the hyperparameter selection is rather weak in our study.

The remaining hyperparameters are described in the following and selected based on literature and expert knowledge. Trial and error attempts to change these other hyperpa-

rameters did not lead to improved results. We use atom and bond features as described in Section 4.1. We apply an edge feature network with three layers and the following number of neurons: #1: 12 (i.e., number of edge features), #2: 128, #3: 4096 (i.e., number of hidden state squared). The MLPs constitute five layers with #1: 128, #2: 64, #3: 32, #4: 16, #5: 1 neurons.

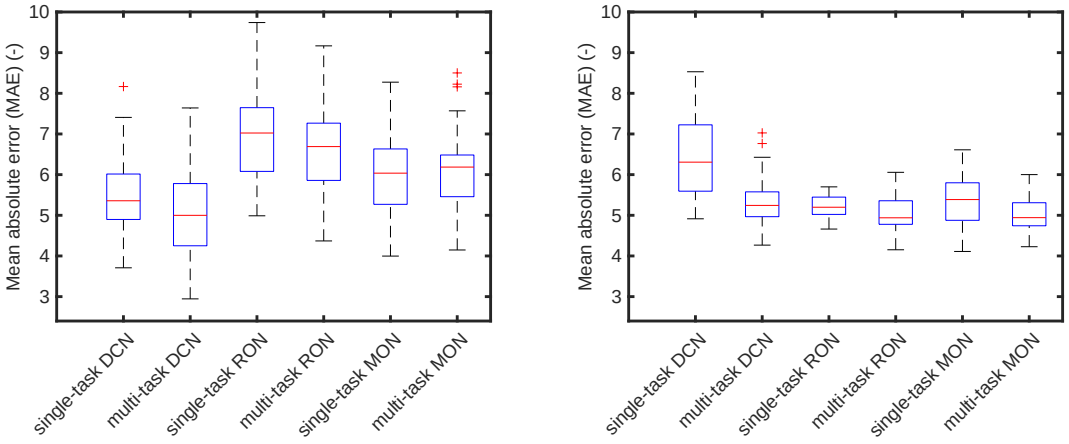
5.3 Single- and multi-task learning

The aforementioned model settings were used for single-task and multi-task learning. The mean absolute errors (MAEs) of the two approaches on their validation and test set are displayed in Figure 5. The respective box plots illustrate the distribution of MAEs over the 40 individual training runs for each model.

Figure 5 shows that the model performance exhibits a high variance. This is mainly caused by the small data size for training, validation, and testing. As the validation sets of the 40 independent model runs are selected randomly, they show a larger variance of the MAE. In contrast, all 40 independent models share the same test set. Thus, the MAE distribution on the test set is more narrow. One methodology against high model variance is bootstrap aggregation which is performed in Section 5.5.

Table 4 summarizes the MAEs on the training, internal validation, and independent test set averaged over the 40 training runs for comparison. The averaged results show that the multi-task training approach improves the prediction accuracy on all test sets and for all predicted properties. For instance, the MAE of the DCN on the test set is reduced by about 17% from 6.4 to 5.3.

The results indicate that the simultaneous learning of DCN, RON, and MON leads to a better generalization of the graph convolutional layers and thus molecular fingerprint. One reason for the synergies are believed to be the correlations between DCN, RON, and MON.[4,8,12,13,78,79]



(a) MAE on validation set

(b) MAE on test set

Figure 5: Comparison of the MAE of the single-task learning and the multi-task learning approach on the validation and test sets. The box-plots indicate the lowest and largest MAE (excluding outliers), the lower and upper quartile, and the median of the MAE over 40 independent model instances. Note that points that are more than 1.5 times the interquartile range away from the top or bottom of the box are marked as outliers.

5.4 Transfer learning

For the transfer learning approach, we pre-train the single-task DCN model on data from all different (D)CN measurement methods, i.e., we use (D)CN data of 447 components collected by Yanowitz et al.[11] (cf. Section 3). Then, the learned parameters are used to initialize the parameters in the graph convolutions and the MLP of the single-task DCN and also the multi-task model. For the latter, only the parameters of the MLP for predicting the DCN

Table 4: Mean absolute error (MAE) of training, validation, and test set averaged over 40 training runs. The table includes single-task learning (STL), multi-task learning (MTL), transfer learning (TL), and ensemble learning (EL). Lowest test set errors are highlighted in bold.

	DCN			RON			MON		
	Train.	Val.	Test	Train.	Val.	Test	Train.	Val.	Test
STL	2.7	5.5	6.4	3.7	7.0	5.2	3.1	6.0	5.4
MTL	1.8	5.1	5.3	2.8	6.7	5.0	2.3	6.1	5.0
STL & TL	2.2	4.6	6.0	–	–	–	–	–	–
MTL & TL	1.8	5.2	5.9	3.2	6.6	5.1	2.6	6.0	4.9
MTL & EL	1.8		**4.2**	2.8		**4.5**	2.3		**4.4**

are transferred from the pre-training since RON and MON values are not subject to transfer learning. Finally, we retrain the models by only considering IQT-DCN data.

The results of the transfer learning approach are summarized in Table 4. Transfer learning improves the MAE of the single-task model for predicting the DCN from 6.4 to 6.0. For the multi-task model, however, transfer learning does not improve prediction accuracy: Test MAEs for RON and MON are almost the same with and without transfer learning. Furthermore, test MAE for DCN even increases from 5.3 to 5.9 with transfer learning. One possible reason for the poor performance of transfer learning in the multi-task learning is that the pre-training is essentially a single-task problem because we use only (D)CN data for pre-training. Thus, the pre-trained model could be biased towards (D)CN which then could lead to poor generalization of the multi-task model. As a consequence, we do not use the transfer learning approach for our final model.

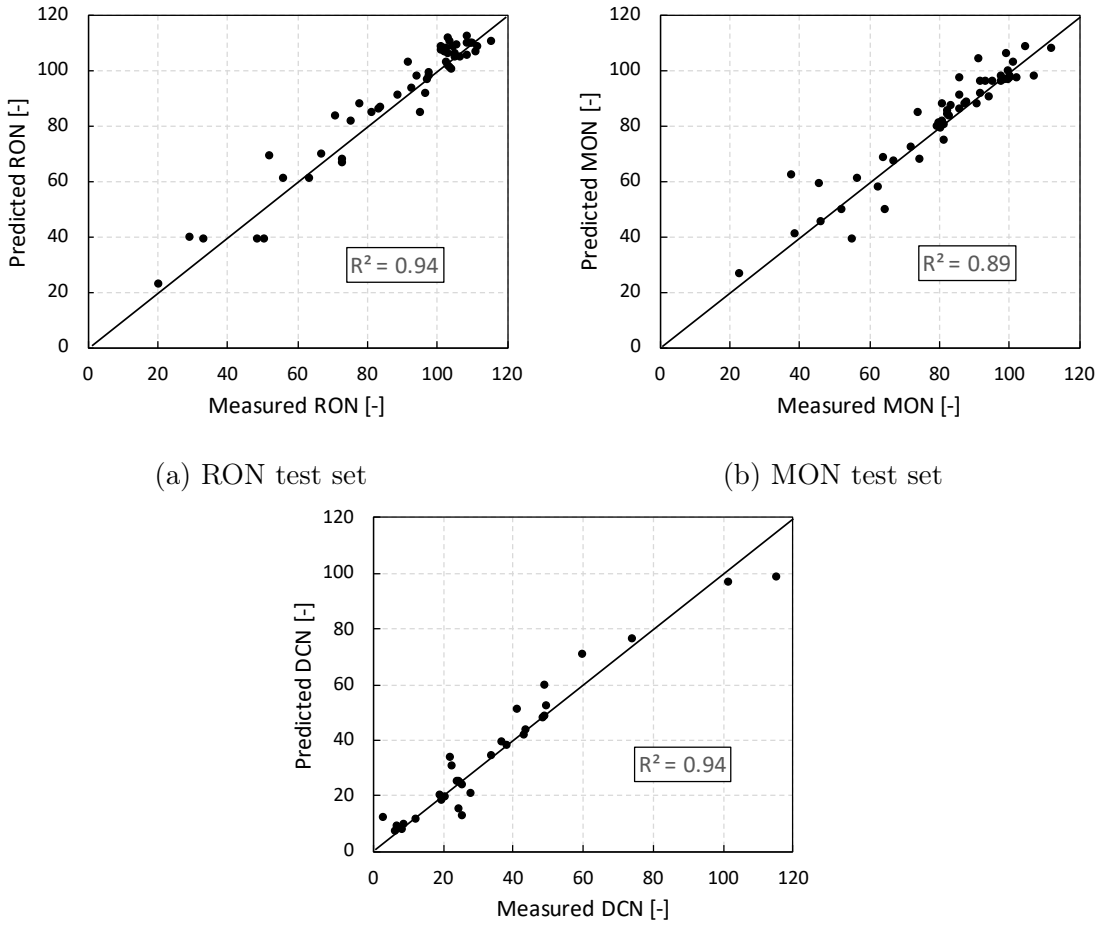
5.5 Ensemble learning

After developing a suitable model architecture, model ensembling is applied to address the observed high variation (cf. discussion in Section 5.3). As described in Section 4.5, ensemble learning averages the response of multiple models and mitigates random model variations. Herein, we utilize the previously trained 40 model instances. We perform the ensemble learning on the multi-task architecture without transfer learning.

The results are summarized in Table 4. They show the averaged MAE on the test set and the combined training and validation set. The error on a validation set is shown as part of the training set because the averaged 40 model instances have individual randomly selected validation sets. Ensemble learning reduces the MAE of the DCN, RON, and MON significantly from 5.3 to 4.2, from 5.0 to 4.5, and from 5.0 to 4.4, respectively. The bootstrap aggregation compensates for the previously identified large model variations.

Figure 6 illustrates the parity plots for the independent test set of the proposed ensemble model. Herein, every point represents the averaged prediction of 40 multi-task models for

a data point in the test set. The plots show high coefficients of determination for all three properties, i.e., $R^2_{\text{DCN}} = 0.94$, $R^2_{\text{RON}} = 0.94$, and $R^2_{\text{MON}} = 0.89$. For the MON, the higher number of outliers causes a slightly weaker coefficient of determination. Note that the parity plots show an uneven distribution of the data in the test set. For instance, there exist few data points with DCN numbers above 100 or RON numbers below 50.



(a) RON test set

(b) MON test set

(c) DCN test set

Figure 6: Multi-task GNN model ensembling: Parity plots for (a) RON, (b) MON, and (c) DCN test sets.

5.6 Comparison to recent QSPR models

We compare our GNN model to three recent literature QSPR models for fuel ignition indicator prediction. Our own previous model follows a group contribution and multivariate

nonlinear regression approach for predicting IQT-DCN.[13] The model by vom Lehn et al.[31] combines group contributions and a feed-forward artificial neural network (ANN) to predict RON. Finally, the model by Kubic et al.[12] combines group contributions and a multi-task feed-forward ANN for simultaneous regression of CN, RON, and MON values.

A fair comparison between the different models is difficult for multiple reasons. First, models have been developed from different training data sets, leading to different model applicability domains. This is most evident in case of the model proposed by vom Lehn et al.,[31] which is applicable to alkanes, alkenes, cyclic alkanes, and alcohols only, whereas molecular diversity of the training data is much higher for our GNN model and the other two models. Second, different performance evaluation methods, i.e., cross-validation or evaluation on an independent test set, and different performance metrics, i.e., mean absolute error (MAE) or coefficient of determination (R^2), have been employed, as can be seen from Table 5. Our procedure for GNN hyperparameter tuning (cf. Section 5.2) follows a cross-validation (CV) approach, i.e., we repeatedly split the dataset into training and validation sets and use the average validation set MAE to find the optimal hyperparameter settings. Since CV-MAE can be a biased estimator of the true prediction error of a model if model parameters are tuned outside of the CV loop,[87] we use an independent test set to estimate the true prediction error. Third, many molecules located in our test set have been used to train the models proposed by Dahmen & Marquardt,[13] Kubic et al.,[12] and vom Lehn et al.[31] Simply rerunning these models on our test set would give them an unfair advantage.

Still, a comparative true prediction performance test can be carried out by exclusively examining those compounds from our test set that were not included in the respective training set of a comparison model. This approach results in the four head-to-head comparisons shown in Tables 6 to 9.

Table 6 compares the DCN predictions made by the GNN model to the predictions made by our previous DCN model and the measurement values. While both models exhibit a decent performance, the MAE of the proposed GNN model is lower than the MAE of our

previous QSPR model (3.6 instead of 5.2). Table 7 shows the comparison between the GNN model and the RON model by vom Lehn et al.[31] Since the latter model is applicable to alkanes, alkenes, cycloalkanes, and alcohols only, this comparison is based on just four compounds. The resulting MAEs are roughly similar. Finally, Table 8 and 9 compare the GNN model to the multi-task ANN model by Kubic et al.,[12] where it can be noted that the MAEs of the GNN model are far lower. In summary, the different head-to-head comparisons suggest highly competitive performance of the new GNN model. We provide our model, the training scripts, and all data sets open-source[48] as well as a web front-end (`www.avt.rwth-aachen.de/gnn`) so that others can easily perform their own benchmarks.

Development of QSPR and GNN models differs significantly from each other also from a conceptual point of view. Most notably, QSPR modeling requires to choose a set of descriptors, e.g., structural group counts, as potential explanatory variables. This step may facilitate understanding of the prediction problem (the human learns through model development) and can encode physical understanding into a tailored model structure. However, this also means that QSPR models inherently rely on assumptions about the underlying phenomena, i.e., the descriptors or structural groups of potential value. In contrast, the presented GNN method is trained in an end-to-end learning approach, as it relies on only few atomic and bond features (cf. Tables 2 and 3), and thus provides a flexible model structure that can possibly learn a broad variety of properties. End-to-end learning with graph convolutions, however, comes at the cost of higher computational effort for training.

Applicability domain (AD) quantification in GNN-based property models is an open research question, which is closely linked with the question of how well GNNs can generalize. Traditional AD methods used in QSPR modeling, e.g., the Williams plot,[88,89] are not directly transferable to GNNs unless a suitable distance metric for molecular graphs is established that can be linked to GNN prediction performance. There are some recent works on autoencoders for molecular graphs[90–95] that might help to derive AD concepts based on a real-valued latent space identified by the autoencoders. When making predictions with

our model, we recommend to check from Table 1 whether property data was available during model development for the molecular class the compound of interest belongs to. For instance, from Table 1 it can be seen that no RON/MON data for aldehydes and cyclic alcohols were available for model training. Whereas it might prove a reasonable assumption that cyclic alcohols behave similar to acyclic alcohols, we abstain from using our model to predict RON/MON of aldehydes.

Table 5: Performance comparison of the proposed model to three recent QSPR models.

	GNN model		Dahmen & Marquardt[13]		Kubic et al.[12]		vom Lehn et al.[31]	
method	test set		test set		cross-validation		cross-validation	
metric	MAE	R^2	MAE	R^2	MAE	R^2	MAE	R^2
DCN	4.2	0.94	5.8	0.84	–	0.90	–	–
RON	4.5	0.94	–	–	–	0.93	4.0	0.92
MON	4.4	0.89	–	–	–	0.91	–	–

Table 6: Comparison with the IQT-DCN model by Dahmen & Marquardt[13] based on those molecules from our DCN test set that were not included in the training set used by Dahmen & Marquardt.[13]

	true value DCN	GNN model DCN	Dahmen & Marquardt[13] DCN
1,2-dimethylbenzene	8.3	7.4	7.9
furan	7.0	8.2	9.4
methyl erucate	74.2	75.9	87.6
2-heptanol	25.0	24.1	21.5
4-ethyl guaiacol	19.6	17.4	25.4
1,3,5-triisopropylbenzene	2.8	11.5	10.4
ocimene	28.0	20.1	14.9
1,4-dimethylbenzene	6.2	6.8	7.9
6-undecanone	49.0	59.2	51.8
4-nonanone	43.0	41.3	41.3
mean absolute error (MAE)		3.6	5.2

Table 7: Comparison with the ANN-based RON model by vom Lehn et al.[31] based on those molecules from our RON test set that were not included in the training set used by vom Lehn et al.[31] Since the model by vom Lehn et al. is applicable to alkanes, alkenes, cycloalkanes, and alcohols only, this comparison is limited to four compounds.

	true value RON	GNN model RON	vom Lehn et al.[31] RON
methylcyclopropane	102.5	107.1	106.0
2,2-dimethyloctane	49.0	38.6	31.7
1,5-hexadiene	71.1	82.9	86.7
2,4-hexadiene	97.1	91.0	98.6
mean absolute error (MAE)		8.2	9.5

Table 8: Comparison with the ANN model by Kubic et al.[12] based on those molecules from our DCN test set that were not included in the training set used by Kubic et al.[12]

	true value DCN	GNN model DCN	Kubic et al.[12] DCN
δ-undecalactone	48.6	47.1	38.6
2-heptanol	25.0	24.1	25.2
4-methoxybenzaldehyde	25.8	12.5	49.7
geraniol	19.3	19.4	22.5
4-ethyl guaiacol	19.6	17.4	5.1
ocimene	28.0	20.1	-2.0
dodecyl vinyl ether	101.7	96.0	126.2
propylene glycol monomethyl ether acetate	24.0	24.5	34.2
6-undecanone	49.0	59.2	58.5
mean absolute error (MAE)		4.7	14.0

Table 9: Comparison with the ANN model by Kubic et al.[12] based on those molecules from our RON & MON test set that were not included in the training set used by Kubic et al.[12]

	true value		GNN model		Kubic et al.[12]	
	RON	MON	RON	MON	RON	MON
methylcyclopropane	102.5	81.2	107.1	87.5	101.1	90.2
furan	108.6	91.6	111.6	103.3	98.9	97.4
aceticacid-2-methylpropylester	108.7	112.3	109.3	107.3	123.4	105.6
4-methylcyclohexene	84.1	67.0	85.8	66.3	59.2	54.2
tetrahydrofuran	72.9	64.8	66.3	49.1	92.8	87.9
2-octene	56.3	56.5	60.3	60.1	42.3	39.1
1-methylcyclohexene	89.2	72.1	90.1	71.6	64.3	58.6
2-phenylpentane	103.5	92.1	101.2	91.2	99.8	97.7
2-methylpropanoicacidmethylester	103.6	104.7	109.8	108.1	138.9	108.9
1,5-hexadiene	71.1	37.6	82.9	61.7	95.9	81.7
3-methyl-2-butanone	108.9	102.2	104.7	96.9	111.6	103.0
methyl-2-methylbutanoate	110.5	99.1	108.9	105.1	119.9	104.8
4-octene (trans)	73.3	74.3	67.5	67.2	89.1	88.7
2,4-hexadiene	97.1	80.7	91.0	78.4	103.3	88.4
allylcyclopentane	52.1	45.6	68.5	58.3	86.0	73.0
mean absolute error (MAE)			5.1	7.0	16.1	13.2

6 Conclusion

Predictive models for fuel ignition quality play a crucial role in the development of novel fuels. We propose a data-driven graph neural network (GNN) model for the prediction of three important fuel auto-ignition indicators, i.e., the derived cetane number (DCN), the research octane number (RON), and the motor octane number (MON). Our model is applicable to a wide spectrum of non-oxygenated and oxygenated hydrocarbons, shows competitive performance to state-of-the-art models, and can be easily accessed via a web interface.

From the methodological point of view, our GNN-based model offers the advantage that, in contrast to previous works based on QSPR modeling, no molecular descriptors or structural groups, have to be selected, because GNNs achieve end-to-end learning from the molecular structure to the properties of interest. While such a data-driven approach is often believed to require extensively large data sets, this work demonstrates that good model ac-

curacies can indeed be achieved for small data sets (order of hundreds) by using multi-task and ensemble learning. Given the expected future increase in measurement data available for training, we expect further potential for GNNs in fuel ignition quality prediction. We provide the corresponding training code and the final model open-source making it a viable tool for further development. Finally, this work may constitute a prototype for rapid, versatile property prediction beyond DCN, RON and MON and thus for property prediction in various disciplines.

Acknowledgement

We thank Sophia Rupprecht for collecting and preparing RON and MON data from literature and Florian vom Lehn for sharing his knowledge on converting RON and MON data for values above 100. Furthermore, we thank Lukas Breuer, Ibrahim Kasem, Jonas Völl, and Fabio Zuraszek for implementing the web interface for online model evaluation. We thank William L. Kubic, Jr. for sharing the Excel implementation of his RON/MON/DCN model. We thank Florian vom Lehn for providing RON predictions made by his model for our test set compounds. This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - Cluster of Excellence 2186 "The Fuel Science Center". Simulations were performed with computing resources granted by RWTH Aachen University under project "thes0682".

Supporting Information Available

The following files are available free of charge.

- RON, MON and (D)CN data (XLSX)

References

(1) Leitner, W.; Klankermayer, J.; Pischinger, S.; Pitsch, H.; Kohse-Höinghaus, K. Advanced biofuels and beyond: Chemistry solutions for propulsion and production. *Angewandte Chemie International Edition* **2017**, *56*, 5412–5452.

(2) Lange, J.-P.; Price, R.; Ayoub, P. M.; Louis, J.; Petrus, L.; Clarke, L.; Gosselink, H. Valeric biofuels: A platform of cellulosic transportation fuels. *Angewandte Chemie International Edition* **2010**, *49*, 4479–4483.

(3) Lange, J.-P.; Van Der Heide, E.; van Buijtenen, J.; Price, R. Furfural—A promising platform for lignocellulosic biofuels. *ChemSusChem* **2012**, *5*, 150–166.

(4) Dahmen, M.; Marquardt, W. Model-based design of tailor-made biofuels. *Energy & Fuels* **2016**, *30*, 1109–1134.

(5) Gschwend, D.; Soltic, P.; Wokaun, A.; Vogel, F. Review and performance evaluation of fifty alternative liquid fuels for spark-ignition engines. *Energy & Fuels* **2019**, *33*, 2186–2196.

(6) König, A.; Ulonska, K.; Mitsos, A.; Viell, J. Optimal applications and combinations of renewable fuel production from biomass and electricity. *Energy & Fuels* **2019**, *33*, 1659–1672.

(7) Regalbuto, J. R. Engineering. Cellulosic biofuels–got gasoline? *Science* **2009**, *325*, 822–824.

(8) Kalghatgi, G. T. *Auto-ignition quality of practical fuels and implications for fuel requirements of future SI and HCCI engines*; 2005; SAE Technical Paper 2005-01-0239.

(9) Kalghatgi, G. Developments in internal combustion engines and implications for combustion science and future transport fuels. *Proceedings of the Combustion Institute* **2015**, *35*, 101–115.

(10) Derfer, J. M.; Boord, C. E.; Burk, F. C.; Hess, R. E.; Lovell, W. G.; Randall, R. A.; Sabina, J. R. *Knocking Characteristics of Pure Hydrocarbons*; ASTM International: 100 Barr Harbor Drive, PO Box C700, West Conshohocken, PA 19428-2959, 1958.

(11) Yanowitz, J.; Ratcliff, M. A.; McCormick, R. L.; Taylor, J. D.; Murphy, M. J. *Compendium of Experimental Cetane Numbers (Technical Report NREL/TP-5400-67585)*; 2017; National Renewable Energy Laboratory (NREL), Golden, CO, United States.

(12) Kubic, W. L.; Jenkins, R. W.; Moore, C. M.; Semelsberger, T. A.; Sutton, A. D. Artificial neural network based group contribution method for estimating cetane and octane numbers of hydrocarbons and oxygenated organic compounds. *Industrial & Engineering Chemistry Research* **2017**, *56*, 12236–12245.

(13) Dahmen, M.; Marquardt, W. A novel group contribution method for the prediction of the derived cetane number of oxygenated hydrocarbons. *Energy & Fuels* **2015**, *29*, 5781–5801.

(14) DeFries, T. H.; Kastrup, R. V.; Indritz, D. Prediction of cetane number by group additivity and carbon-13 nuclear magnetic resonance. *Industrial & Engineering Chemistry Research* **1987**, *26*, 188–193.

(15) Yang, H.; Fairbridge, C.; Ring, Z. Neural network prediction of cetane numbers for isoparaffins and diesel fuel. *Petroleum Science and Technology* **2001**, *19*, 573–586.

(16) Lapidus, A.; Smolenskii, E.; Bavykin, V.; Myshenkova, T.; Kondrat'ev, L. Models for the calculation and prediction of the octane and cetane numbers of individual hydrocarbons. *Petroleum Chemistry* **2008**, *48*, 277–286.

(17) Santana, R. C.; Do, P. T.; Santikunaporn, M.; Alvarez, W. E.; Taylor, J. D.; Sughrue, E. L.; Resasco, D. E. Evaluation of different reaction strategies for the improvement of cetane number in diesel fuels. *Fuel* **2006**, *85*, 643–656.

(18) Sennott, T.; Gotianun, C.; Serres, R.; Ziabasharhagh, M.; Mack, J.; Dibble, R. Artificial neural network for predicting cetane number of biofuel candidates based on molecular structure. ASME 2013 Internal Combustion Engine Division Fall Technical Conference, 13.-16.10.2013, Dearborn, Michigan, USA. 2013.

(19) Kessler, T.; Sacia, E. R.; Bell, A. T.; Mack, J. H. Artificial neural network based predictions of cetane number for furanic biofuel additives. *Fuel* **2017**, *206*, 171–179.

(20) Guan, C.; Zhai, J.; Han, D. Cetane number prediction for hydrocarbons from molecular structural descriptors based on active subspace methodology. *Fuel* **2019**, *249*, 1–7.

(21) Ogawa, H.; Nishimoto, H.; Morita, A.; Shibata, G. Predicted diesel ignitability index based on the molecular structures of hydrocarbons. *International Journal of Engine Research* **2016**, *17*, 766–775.

(22) Baghban, A.; Kardani, M. N.; Mohammadi, A. H. Improved estimation of cetane number of fatty acid methyl esters (FAMEs) based biodiesels using TLBO-NN and PSO-NN models. *Fuel* **2018**, *232*, 620–631.

(23) Baghban, A.; Adelizadeh, M. On the determination of cetane number of hydrocarbons and oxygenates using adaptive neuro fuzzy inference system optimized with evolutionary algorithms. *Fuel* **2018**, *230*, 344–354.

(24) Miraboutalebi, S. M.; Kazemi, P.; Bahrami, P. Fatty acid methyl ester (FAME) composition used for estimation of biodiesel cetane number employing random forest and artificial neural networks: A new approach. *Fuel* **2016**, *166*, 143–151.

(25) Smolenskii, E.; Bavykin, V.; Ryzhov, A.; Slovokhotova, O.; Chuvaeva, I.; Lapidus, A. Cetane numbers of hydrocarbons: Calculations using optimal topological indices. *Russian Chemical Bulletin* **2008**, *57*, 461–467.

(26) Creton, B.; Dartiguelongue, C.; de Bruin, T.; Toulhoat, H. Prediction of the cetane number of diesel compounds using the quantitative structure property relationship. *Energy & Fuels* **2010**, *24*, 5396–5403.

(27) Guo, Z.; Lim, K. H.; Chen, M.; Thio, B. J. R.; Loo, B. L. W. Predicting cetane numbers of hydrocarbons and oxygenates from highly accessible descriptors by using artificial neural networks. *Fuel* **2017**, *207*, 344–351.

(28) Saldana, D. A.; Starck, L.; Mougin, P.; Rousseau, B.; Pidol, L.; Jeuland, N.; Creton, B. Flash point and cetane number predictions for fuel compounds using quantitative structure property relationship (QSPR) methods. *Energy & Fuels* **2011**, *25*, 3900–3908.

(29) Meusinger, R.; Moros, R. Determination of quantitative structure–octane rating relationships of hydrocarbons by genetic algorithms. *Chemometrics and Intelligent Laboratory Systems* **1999**, *46*, 67–78.

(30) Albahri, T. A. Structural group contribution method for predicting the octane number of pure hydrocarbon liquids. *Industrial & Engineering Chemistry Research* **2003**, *42*, 657–662.

(31) vom Lehn, F.; Brosius, B.; Brust, D.; Cai, L.; Pitsch, H. Using machine learning in model development for global fuel auto-ignition quantities. 29. Deutscher Flammentag, 17.-18.09.2019, Bochum, Germany. 2019.

(32) Abdul Jameel, A. G.; van Oudenhoven, V.; Emwas, A.-H.; Sarathy, S. M. Predicting octane number using nuclear magnetic resonance spectroscopy and artificial neural networks. *Energy & Fuels* **2018**, *32*, 6309–6329.

(33) Katritzky, A. R.; Kuanar, M.; Slavov, S.; Hall, C. D.; Karelson, M.; Kahn, I.; Dobchev, D. A. Quantitative correlation of physical and chemical properties with chemical structure: Utility for prediction. *Chemical Reviews* **2010**, *110*, 5714–5789.

(34) Benson, S. W.; Cruickshank, F. R.; Golden, D. M.; Haugen, G. R.; O'Neal, H. E.; Rodgers, A. S.; Shaw, R.; Walsh, R. Additivity rules for the estimation of thermochemical properties. *Chemical Reviews* **1969**, *69*, 279–324.

(35) Joback, K. G.; Reid, R. C. Estimation of pure-component properties from group-contributions. *Chemical Engineering Communications* **1987**, *57*, 233–243.

(36) Gani, R.; Nielsen, B.; Fredenslund, A. A group contribution approach to computer-aided molecular design. *AIChE Journal* **1991**, *37*, 1318–1332.

(37) Wiener, H. Structural determination of paraffin boiling points. *Journal of the American Chemical Society* **1947**, *69*, 17–20.

(38) LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444.

(39) Gori, M.; Monfardini, G.; Scarselli, F. A new model for learning in graph domains. Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN), 31.07.-04.08.2005. Montreal, Quebec, Canada, 2005; pp 729–734.

(40) Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; Monfardini, G. The graph neural network model. *IEEE Transactions on Neural Networks* **2009**, *20*, 61–80.

(41) Coley, C. W.; Barzilay, R.; Green, W. H.; Jaakkola, T. S.; Jensen, K. F. Convolutional embedding of attributed molecular graphs for physical property prediction. *Journal of Chemical Information and Modeling* **2017**, *57*, 1757–1772.

(42) Coley, C. W.; Jin, W.; Rogers, L.; Jamison, T. F.; Jaakkola, T. S.; Green, W. H.; Barzilay, R.; Jensen, K. F. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chemical Science* **2019**, *10*, 370–377.

(43) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212v2* **2017**, arXiv.

(44) Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M.; Palmer, A.; Settels, V.; Jaakkola, T.; Jensen, K.; Barzilay, R. Analyzing learned molecular representations for property prediction. *Journal of Chemical Information and Modeling* **2019**, *59*, 3370–3388.

(45) Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. Molecular graph convolutions: Moving beyond fingerprints. *Journal of Computer-Aided Molecular Design* **2016**, *30*, 595–608.

(46) Morris, C.; Ritzert, M.; Fey, M.; Hamilton, W.; Lenssen, J.; Rattan, G.; Grohe, M. Weisfeiler and Leman go neural: Higher-order graph neural networks. Proceedings of the 33rd AAAI Conference on Artificial Intelligence, 27.01.-01.02.2019, Honolulu, Hawaii, United States. 2019.

(47) Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional networks on graphs for learning molecular fingerprints. Advances in Neural Information Processing Systems 28 (NIPS 2015). 2015; pp 2224–2232.

(48) Schweidtmann, A. M.; Rittig, J. G.; M., G.; Mitsos, A. Open-source graph neural network for prediction of fuel ignition quality. `https://git.rwth-aachen.de/avt.svt/public/graph_neural_network_for_fuel_ignition_quality`, 2020; accessed on 05.04.2020.

(49) Bonchev, D.; Rouvray, D. *Chemical Graph Theory: Introduction and Fundamentals*; Gordon and Breach Science Publishers, New York, United States, 1991.

(50) Minkin, V. I. Glossary of terms used in theoretical organic chemistry. *Pure and Applied Chemistry* **1999**, *71*, 1919–1981.

(51) Todeschini, R.; Consonni, V. *Handbook of Molecular Descriptors*; Wiley-VCH Verlag GmbH: Weinheim, Germany, 2000.

(52) Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: A benchmark for molecular machine learning. *Chemical Science* **2018**, *9*, 513–530.

(53) Zhou, J.; Cui, G.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434v4* **2018**, arXiv.

(54) Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. Advances in Neural Information Processing Systems 30 (NIPS 2017). 2017; pp 1024–1034.

(55) American Society for Testing and Materials, ASTM D 613: Standard Test Method for Cetane Number of Diesel Fuel Oil. 2015.

(56) American Society for Testing, ASTM D 6890: Standard Test Method for Determination of Ignition Delay and Derived Cetane Number (DCN) of Diesel Fuel Oils by Combustion in a Constant Volume Chamber. 2011.

(57) Kalghatgi, G. T. The outlook for fuels for internal combustion engines. *International Journal of Engine Research* **2014**, *15*, 383–398.

(58) American Society for Testing and Materials, ASTM D 2699: Standard Test Method for Research Octane Number of Spark-Ignition Engine Fuel. 2018.

(59) American Society for Testing and Materials, ASTM D 2700: Standard Test Method for Motor Octane Number of Spark-Ignition Engine Fuel. 2019.

(60) Egloff, G.; Van Arsdell, P. Octane rating relationships of aliphatic, alicyclic, mononuclear aromatic hydrocarbons, alcohols, ethers, and ketones. *Journal of the Institute of Petroleum (London)* **1941**, *27*, 121–138.

(61) Yanowitz, J.; Christensen, E.; McCormick, R. L. *Utilization of renewable oxygenates as gasoline blending components (Technical Report NREL/TP-5400-50791)*; 2011; National Renewable Energy Laboratory (NREL), Golden, CO, United States.

(62) McCormick, R. L.; Fioroni, G.; Fouts, L.; Christensen, E.; Yanowitz, J.; Polikarpov, E.; Albrecht, K.; Gaspar, D. J.; Gladden, J.; George, A. Selection criteria and screening of potential biomass-derived streams as fuel blendstocks for advanced spark-ignition engines. *SAE International Journal of Fuels and Lubricants* **2017**, *10*, 442–460.

(63) Leppard, W. R. The autoignition chemistries of octane-enhancing ethers and cyclic ethers: A motored engine study. *SAE Transactions* **1991**, 589–604.

(64) Harvey, B. G.; Merriman, W. W.; Quintana, R. L. Renewable gasoline, solvents, and fuel additives from 2,3-butanediol. *ChemSusChem* **2016**, *9*, 1814–1819.

(65) Naegeli, D. W.; Yost, D. M.; Moulton, D. S.; Owens, E. C.; Chui, G. K. The measurement of octane numbers for methanol and reference fuels blends. *SAE Transactions* **1989**, 712–722.

(66) Szybist, J.; West, B. Update on co-optima light-duty spark-ignition research. 2017; `http://nresolutions.com/AAE_Files/WG_Mtg_10_12-13_17/10.13_Presentations/02-J.Szybist_B.West_Co-Optima.pdf`, accessed on 05.04.2020.

(67) Römpp Online, Octan-Zahl. 2002; `https://roempp.thieme.de/lexicon/RD-15-00161`, accessed on 05.04.2020.

(68) Fey, M.; Lenssen, J. E. Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428v3* **2019**, arXiv.

(69) Weininger, D. SMILES, A chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Modeling* **1988**, *28*, 31–36.

(70) Greg Landrum, RDKit: Open-Source Cheminformatics. `http://www.rdkit.org`, accessed on 05.04.2020.

(71) Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078v3* **2014**, arXiv.

(72) Li, Y.; Tarlow, D.; Brockschmidt, M.; Zemel, R. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493v4* **2015**, arXiv.

(73) Caruana, R. Multitask learning. *Machine Learning* **1997**, *28*, 41–75.

(74) Ruder, S. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* **2017**, arXiv.

(75) Zhang, Y.; Yang, Q. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114* **2017**, arXiv.

(76) Dahl, G. E.; Jaitly, N.; Salakhutdinov, R. Multi-task neural networks for QSAR predictions. *arXiv preprint arXiv:1406.1231* **2014**, arXiv.

(77) Ramsundar, B.; Kearnes, S.; Riley, P.; Webster, D.; Konerding, D.; Pande, V. Massively multitask networks for drug discovery. *arXiv preprint arXiv:1502.02072* **2015**,

(78) Ryan III, T. W.; Matheaus, A. C. Fuel requirements for HCCI engine operation. *SAE Transactions* **2003**, 1143–1152.

(79) Perez, P. L.; Boehman, A. L. Experimental investigation of the autoignition behavior of surrogate gasoline fuels in a constant-volume combustion bomb apparatus and its relevance to HCCI Combustion. *Energy & Fuels* **2012**, *26*, 6106–6117.

(80) Torrey, L.; Shavlik, J. *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*; IGI Global, 2010; pp 242–264.

(81) Pan, S. J.; Yang, Q. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* **2009**, *22*, 1345–1359.

(82) Grambow, C. A.; Li, Y.-P.; Green, W. H. Accurate thermochemistry with small data sets: A bond additivity correction and transfer learning approach. *The Journal of Physical Chemistry A* **2019**, *123*, 5826–5835.

(83) Breiman, L. Bagging predictors. *Machine Learning* **1996**, *24*, 123–140.

(84) Breiman, L. Stacked regressions. *Machine Learning* **1996**, *24*, 49–64.

(85) Dietterich, T. G. Ensemble methods in machine learning. Multiple Classifier Systems: First International Workshop (MCS 2000), Lecture Notes in Computer Science, 21.06.-23.06.2000. Cagliari, Italy, 2000; pp 1–15.

(86) Freund, Y.; Schapire, R. E. Experiments with a New Boosting Algorithm. Proceedings of the Thirteenth International Conference on Machine Learning, 03.07.-06.07.1996. Bari, Italy, 1996; pp 148–156.

(87) Varma, S.; Simon, R. Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics* **2006**, *7*, 91.

(88) Gramatica, P. Principles of QSAR models validation: internal and external. *QSAR & Combinatorial Science* **2007**, *26*, 694–701.

(89) Gramatica, P. *Computational Toxicology*; Springer, 2013; pp 499–526.

(90) De Cao, N.; Kipf, T. MolGAN: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973* **2018**, arXiv.

(91) Liu, Q.; Allamanis, M.; Brockschmidt, M.; Gaunt, A. Constrained graph variational autoencoders for molecule design. Advances in Neural Information Processing Systems 31 (NIPS 2018). 2018; pp 7795–7804.

(92) Simonovsky, M.; Komodakis, N. Graphvae: Towards generation of small graphs using variational autoencoders. 27th International Conference on Artificial Neural Networks (ICANN 2018), 04.-07.10.2018. Rhodes, Greece, 2018; pp 412–422.

(93) Jin, W.; Barzilay, R.; Jaakkola, T. Junction tree variational autoencoder for molecular graph generation. Proceedings of the 35th International Conference on Machine Learning (ICML 2018), 10.-15.06.2018. Stockholmsmässan, Stockholm, Sweden, 2018; pp 2323–2332.

(94) Krenn, M.; Häse, F.; Nigam, A.; Friederich, P.; Aspuru-Guzik, A. Self-Referencing Embedded Strings (SELFIES): A 100% robust molecular string representation. *arXiv preprint arXiv:1905.13741* **2019**, arXiv.

(95) Kajino, H. Molecular hypergraph grammar with its application to molecular optimization. Proceedings of the 36th International Conference on Machine Learning (ICML 2019), 9.-15.06.2019. Long Beach, California, USA, 2019; pp 3183–3191.

Graphical TOC Entry