

Physical Pooling Functions in Graph Neural Networks for Molecular Property Prediction

Artur M. Schweidtmann^{1,2}, Jan G. Rittig¹, Jana M. Weber³, Martin Grohe⁴, Manuel Dahmen⁵, Kai Leonhard⁶, Alexander Mitsos^{*,7,5,1}

¹ RWTH Aachen University, Process Systems Engineering (AVT.SVT), Forckenbeckstr. 51, 52074 Aachen, Germany

² Current address: Delft University of Technology, Department of Chemical Engineering, Van der Maasweg 9, Delft 2629 HZ, The Netherlands

³ Delft University of Technology, Delft Bioinformatics Lab, Intelligent Systems, TU Delft, 2628 XE Delft, The Netherlands

⁴ RWTH Aachen University, Lehrstuhl für Informatik 7, Ahornstr. 55, 52074 Aachen, Germany

⁵ Forschungszentrum Jülich GmbH, Institute of Energy and Climate Research IEK-10 – Energy Systems Engineering, Wilhelm-Johnen-Str., 52425 Jülich, Germany

⁶ RWTH Aachen University, Institute of Technical Thermodynamics, Schinkelstr. 8, 52062 Aachen, Germany

⁷ JARA Center for Simulation and Data Science (CSD), Aachen, Germany

* Corresponding author: amitsos@alum.mit.edu

Abstract: Graph neural networks (GNNs) are emerging in chemical engineering for the end-to-end learning of physicochemical properties based on molecular graphs. A key element of GNNs is the pooling function which combines atom feature vectors into molecular fingerprints. Most previous works use a standard pooling function to predict a variety of properties. However, unsuitable pooling functions can lead to unphysical GNNs that poorly generalize. We compare and select meaningful GNN pooling methods based on physical knowledge about the learned properties. The impact of physical pooling functions is demonstrated with molecular properties calculated from quantum mechanical computations. We also compare our results to the recent set2set pooling approach. We recommend using sum pooling for the prediction of properties that depend on molecular size and compare pooling functions for properties that are molecular size-independent. Overall, we show that the use of physical pooling functions significantly enhances generalization.

Keywords

graph convolutional neural networks, pooling function, physics-informed machine learning, property prediction

1 Introduction

Graph neural networks (GNNs) are emerging for end-to-end learning of molecular properties [8, 15, 19, 27] in a broad variety of applications including chemical engineering [22, 31, 32, 38], (quantum) chemistry [2, 12, 34, 45, 48] and the prediction of physical [7] and crystal properties [5, 46]. Although GNNs are flexible

models for end-to-end learning, we show that their pooling function needs to be carefully selected because wrong decisions can lead to unphysical GNNs that are more prone to overfitting. GNNs take molecular graphs as inputs and represent atoms by nodes and bonds by edges. In addition, atoms and nodes are characterized by corresponding feature vectors. Most commonly, GNN architectures are based on message passing neural networks (MPNNs) [12]. In MPNNs, the node feature vectors are updated through a series of message passing procedures of neighboring nodes. Each of these sequential message passings corresponds to the graph convolutional layers of the GNN. Then, the resulting node feature vectors are combined into a molecular fingerprint vector through pooling. This molecular fingerprint is finally mapped to molecular properties of interest by feedforward artificial neural networks (ANNs). We show that the selection of the pooling function, which combines the feature vectors of all atoms into the molecular fingerprint, is critical for the GNN’s performance.

The vast majority of previous works does not select pooling functions based on physical understanding. In the previous literature, common pooling functions are mean, sum, and max pooling [44]. Most previous works use sum pooling [7, 24, 47, 48], while a few use mean pooling [26, 39]. Moreover, typically the same pooling function is used for a range of different properties. We argue that this can lead to unphysical GNNs and result in unnecessary errors. An illustrative example is the molecular weight that is given by the sum of the atom weights. In this case, using mean pooling in a standard GNN would lead to an unphysical architecture that cannot learn the correct underlying physics because the molecular mass cannot be computed as an average of atom weights. In contrast, selecting the sum pooling function according to the underlying physics enables the GNN to learn a meaningful model.

Some researchers circumvent the issue of selecting pooling functions by introducing flexible models for pooling such as set2set [43], DiffPool [50], or SortPool [51]. For example, the set2set approach employs a long short-term memory (LSTM) architecture designed for unordered and size-variant input sets [43]. The authors of DiffPool propose a hierarchical GNN structure that progressively coarsens the input graph in each layer by aggregating clusters of nodes until a single graph representation is obtained [50]. SortPool arranges the learned node representation in a consistent ordered tensor which is then truncated or extended to a user-defined fixed size [51]. Similarly, GNNs with a large number of convolutional layers combine node information through convolutions and thus reduce the importance of pooling functions [26]. Advanced pooling methods have also been applied in molecular property prediction. For instance, Gilmer et al. (2017) [12] applied the set2set method for learning various molecular properties from the QM9 data set [30, 33] and achieved state-of-the-art accuracies on all target properties compared to other GNN models at the time of publication. However, the additional flexibility typically results in larger data requirements, higher model variance, and the risk of overfitting. In other words, the selection of physical pooling functions over flexible model architectures for pooling can be understood as enforcing a hybrid model structure, which is known to reduce the data demand [10, 29, 37].

A few recent studies emphasize the importance of the choice of pooling functions for property prediction. Xu et al. (2018) [47] examine sum, mean, and max pooling and conclude that sum pooling is more powerful than mean and max pooling since it can better distinguish different graph structures. Pronobis

et al. (2018) [28] state that decomposition of molecules into atom-wise contributions combined with a property-suitable pooling function works better for “extensive properties”. Other works use property-specific pooling functions, where mean/set2set or sum pooling is applied to “intensive” or “extensive” properties, respectively [14, 23, 35, 36, 49]. Overall, the physical selection of pooling functions is somewhat contradictory in the previous literature and the terms “intensive” or “extensive” have been used colloquially and not in their thermodynamic sense. Also, a comparison of different pooling functions on the prediction and generalization capabilities of GNNs against a physical background has not been conducted yet, hence there is no guide for selecting suitable pooling functions based on physical knowledge in the literature.

We evaluate GNN pooling functions against the underlying physical nature of the learned properties. We analyze the impact of physical pooling functions on molecular properties learned from the common QM9 data set [30, 33] and demonstrate their superior performance.

2 Materials and methods

We first describe the graph representation of molecules and then briefly introduce the general GNN architecture used for property prediction. Finally, we provide physical insight into the learned properties and use the insight to design physical GNN architectures.

2.1 Molecular graph

Molecules can be described as molecular graphs with nodes $v, w \in V$ representing atoms and edges $e_{vw} \in E$ representing bonds. Each atom is described by a feature vector $\mathbf{f}^V(v)$, containing atom information, e.g., atom mass or orbital hybridization. Similarly, each bond is described by a bond feature vector $\mathbf{f}^E(e_{vw})$ that contains information on the bond type, e.g., single or double bond. Commonly, for organic molecules the hydrogen (H) atoms are omitted and replaced by the hydrogen count as a node feature [41]; this results in reducing the complexity of molecular graphs and therefore reducing data demand.

2.2 Graph neural network

GNNs exhibit two phases [12] as shown in Figure 1: (i) message passing phase and (ii) readout phase.

To initialize the message passing phase, each node $v \in V$ is assigned a state vector $\mathbf{h}_v^{l=0}$ initialized by the respective node feature vector [12]. Then, the state vector of the nodes in layer l are updated with information from their neighboring nodes $w \in N(v)$ along edges e_{vw} :

$$\mathbf{h}_v^l = U_l \left(\mathbf{h}_v^{l-1}, \sum_{w \in N(v)} M_l \left(\mathbf{h}_v^{l-1}, \mathbf{h}_w^{l-1}, \mathbf{f}^E(e_{vw}) \right) \right),$$

where $U_l(\cdot)$ and $M_l(\cdot)$ respectively denote the state update function and the message function in layer l . This message passing procedure is repeated L times

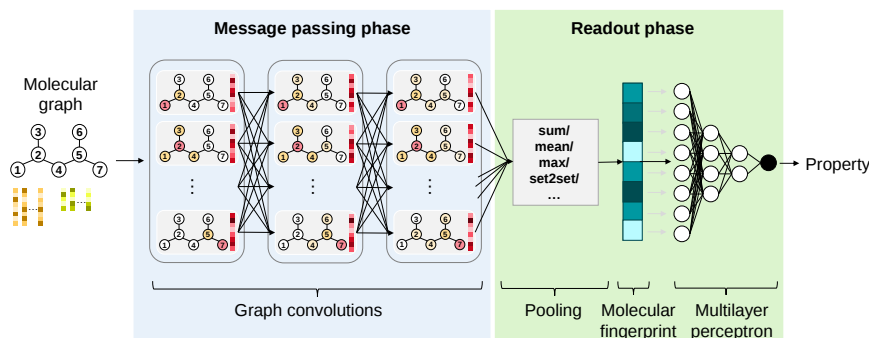


Fig. 1: Illustration of the GNN structure highlighting the message passing and readout phases.

until each node state vector $\mathbf{h}_v^L$ includes information about its local environment. This iterative message passing corresponds to the stacking of L graph convolutional layers.

We consider a standard GNN including edge features in the message passing phase, also known as 1-GNN [16, 26]. The 1-GNN uses the following message passing function:

$$\mathbf{h}_v^l = \sigma \left(\theta_v^l \cdot \mathbf{h}_v^{l-1} + \sum_{w \in N(v)} \text{ANN}_{\theta_e^l}(\mathbf{f}^E(e_{vw})) \cdot \mathbf{h}_w^{l-1} \right),$$

where σ indicates an activation function, θ_v^l denotes a parameter matrix, and $\text{ANN}_{\theta_e^l}$ denotes a feedforward ANN mapping the respective feature vectors $\mathbf{f}^E(e_{vw})$ of the edges e_{vw} connecting node v with its neighbors to a parameter matrix θ_e^l , referred to as edge feature network.

In the readout phase, the final state vectors of the nodes $\mathbf{h}_v^L$ are combined into a graph state vector $\mathbf{h}_G$ by a pooling function. The pooling function is necessary for molecular property prediction because the number of atoms usually differs between different molecules. This leads to a varying number of atom feature vectors that need to be combined to the molecular fingerprints. Thus, the pooling function combines a varying number of final state vectors for the nodes into a single graph state vector. In the context of molecular property prediction, the literature commonly refers to $\mathbf{h}_G$ as the molecular fingerprint. This molecular fingerprint $\mathbf{h}_G$ is finally fed into a feed-forward ANN for the prediction of molecular properties, $\hat{\mathbf{p}} = \text{MLP}(\mathbf{h}_G)$.

The molecular fingerprint is given by the pooling function $f_p(\cdot)$ that depends on the final state vectors of the nodes $\mathbf{h}_v^L$ with $v \in V$:

$$\mathbf{h}_G = f_p \left(\{\mathbf{h}_v^L \mid v \in V\} \right)$$

Common choices for f_p are the sum, mean, and max functions. An alternative pooling function is the set2set method [43] which can capture more complex relationships between different atomic contributions [12, 35] by employing a long short-term memory (LSTM) model [43]. After T steps of the following iterative computation, the molecular fingerprint is obtained by $\mathbf{h}_G = \mathbf{q}_{t=T}^*$ with

$$\begin{aligned}\mathbf{q}_t^* &= \mathbf{q}_t \parallel \mathbf{r}_t \\ \mathbf{q}_t &= \text{LSTM}(\mathbf{q}_{t-1}^*) \\ \mathbf{r}_t &= \sum_v \mathbf{a}_{v,t} \cdot \mathbf{h}_v^L \\ \mathbf{a}_{v,t} &= \text{softmax}(\mathbf{h}_v^L \cdot \mathbf{q}_t)\end{aligned}$$

where $\mathbf{q}_t$ is a query vector for iteration t providing information about the previous attention readout vector $\mathbf{r}_t$ from the memories, $\mathbf{a}_{v,t}$ is an attention vector resulting from averaging the attention of a node v by applying the softmax function, $\mathbf{r}_t$ is the attention readout, similar to the simple pooling method with sum, and $\mathbf{q}_t$ is a concatenation ($\parallel$) of the current query vector and the attention readout. The vector $\mathbf{q}_{t-1}^*$ is initialized at $t = 0$ by $\mathbf{q}_{-1}^* = \mathbf{0}$.

Recent GNNs incorporate physical knowledge into message passing. Over the last years, multiple MPNN architectures have been proposed that integrate physical knowledge to the message passing scheme, e.g., SchNet [35], PhysNet [42], DimeNet [21], MXMNet [52]. This includes the incorporation of directional information, such as interatomic distances and angles between atom pairs, into the message function $M_l(\cdot)$ modeling the interactions of atoms. We consider MXMNet that utilizes physical-driven message passing while preserving computational efficiency [52]. Within MXMNet, two message passing schemes are applied. In a global message passing scheme, information between atoms with a global cutoff distance d_g is exchanged. Further, a local message passing is applied to exchange information between atoms with a local cutoff distance d_l with $d_g > d_l$. This local cutoff distance represents the connectivity of atoms that are connected by chemical bonds. The architecture further enables to transfer of information between atom representations in the global and local message passing by including a cross layer mapping. In the readout step, the learned atom-wise representations are subsequently pooled by the sum operator for molecular property prediction.

2.3 Physical insight

The prediction of molecular properties by decomposing molecules into atomic contributions has a long history in chemical research. According to Bonchev [4], the first investigations into properties of molecules with additive characteristics in terms of atomic contributions were carried out in the 1850s. Later, quantitative structure-property relationship (QSPR) and group additivity methods were developed based on the additive character of atoms or functional groups within a molecule [3, 11, 17, 18]. Yet, not every molecular property exhibits purely additive effects.

In thermodynamics, macroscopic properties are categorized as intensive or extensive [6]. A system property is extensive if it scales linearly with the mass (and as such “extent”) of the system; examples are the mass or volume. In con-

trast, intensive properties do not change with the system mass. Note that sometimes thermodynamicists also distinguish between intensive (e.g., temperature and pressure) and specific (extensive quantity divided by volume, e.g., density) properties [40], but we will not. We transfer these concepts to molecules and distinguish between *molecular size-independent* and *molecular size-dependent* properties. Molecular size-dependent properties scale with the number of atoms in a molecule. For example, the molecular weight is determined by how many atoms of which type are present in a molecule. In contrast, there exist molecular size-independent properties that do not scale with the number of atoms in a molecule, e.g., the highest occupied molecular energy level (HOMO) [35]. Moreover, some properties of a substance, e.g., activity or toxicity, are mostly influenced by certain functional groups or structural fragments. Note that this dependency on molecular size does not necessarily correspond to the formal definition of intensive or extensive properties in a thermodynamic sense because the former is considered at a microscopic atom-based molecular level, not at a macroscopic mass-based system level. For example, the molar enthalpy with the unit J/mol is an intensive property. On a molecular level, the enthalpy of atomization $H_{298,atom}$ with the unit J/mol, is a molecular size-dependent property describing the amount of energy needed to break up a molecule into all of its single atoms at room temperature and fixed pressure [12].

Schütt et al. [35] consider the QM9 properties dipole moment (μ), isotropic polarizability (α), electronic spatial extent (R^2), zero point vibrational energy (ZPVE), heat capacity at 298.15K ($C_{v,298}$), atomization energy at 0K ($U_{0,atom}$), atomization energy at 298.15K ($U_{298,atom}$), enthalpy of atomization at 298.15K ($H_{298,atom}$), free energy of atomization at 298.15K ($G_{298,atom}$) as “extensive”. Other properties are the highest occupied molecular orbital (ϵ_{HOMO}), lowest unoccupied molecular orbital energy level (ϵ_{LUMO}), and HOMO-LUMO gap ($\Delta\epsilon$). For some of these properties, physical dependencies on molecular size are known. For example, Miller and Savchik [25] developed a semi-empirical approach for the prediction of isotropic polarizabilities as a sum of atomistic contributions that depend on their hybridization states based on theoretical calculations already in 1979. Each nonlinear molecule has $3N - 6$ vibrational degrees of freedom ($3N - 5$ for linear ones), N being its number of atoms. Each degree of freedom has a ZPVE proportional to its frequency $\nu \propto \sqrt{f/m}$. Here, m is the reduced mass of the parts of a molecule that vibrate with respect to each other and f is the force constant of this vibration. Hence, ZPVE is a molecular size-dependent property to first order but the frequencies of vibrations that include large fractions of a molecule decrease with increasing molecular size. This effect should be learned by the GNN. Similar relations apply to the heat capacity, enthalpy, and entropy contributions with the minor complication that terms dependent on the molecular mass (translation) and the moment of inertia (rotation) arise for some of the contributions [1]. Atomization energies and enthalpies include essentially sums of contributions of all bonds, which may be non-local in the case of conjugated bonds, and are hence molecular size-dependent properties as well. The electronic spatial extent is determined mainly by the shapes of the orbitals in very small molecules and is closely related to the radius of gyration for large molecules, which may even depend on the solvent for large molecules such as polymers and may scale with a fractal exponent in this case.

The dipole moment (μ) is a particularly interesting property. Even though formally the molecule size enters the dipole moment equation, in most molecules

local functional groups determine μ , and depending on orientation they can even weaken each other. In particular, one or a few strongly polar groups dominate the dipole moment that can then be written as the sum of the individual dipole moments vectors. Thus, we classify the dipole moment as molecular size independent. The prediction of dipole moment is expected to be challenging for conventional GNNs because long-range orientational relations between the polar groups may need to be learned by a model, e.g., for describing the difference between the polar ortho- and the unpolar para-benzoquinone.

Similarly, the relations are complex for energies of the HOMO, the LUMO, and their difference (i.e., the HOMO-LUMO-gap). Depending on the type of molecule, these orbitals may be quite localized to a certain functional group and thus independent of molecular size in some molecules. However, they may also be delocalized in other molecules and thus dependent on molecular size for small and medium-sized molecules but converge to a limit for large molecules as can be seen, e.g., from the Hückel model for conjugated double bonds [1]. Hence this property may be particularly challenging for a GNN model.

3 Results and discussion

In order to demonstrate the relevance of the pooling function in the readout phase, two case studies are conducted and discussed below. First, we consider the illustrative prediction of the molecular weight. Then, we consider the prediction of twelve quantum mechanical properties collected in the QM9 data set.

3.1 Hyperparameters and implementation

Our implementations are based on the models in PyTorch Geometric developed by Fey & Lenssen [9]. For our case study, we combine each mean, sum, and max as well as set2set pooling with the 1-GNN. The hyperparameters of the models are selected based on our experience from our previous work on predicting fuel properties [38]. The molecular graphs have the following features encoded as a one-hot vector: (node) atom type, is aromatic, is in ring, hybridization (e.g., sp , sp^2 , sp^3) hydrogen count, (edge) bond type, conjugated, and stereo. The 1-GNN comprises three graph convolutional layers with hidden dimension size of 64. To map the molecular fingerprint ($\mathbf{h}_G$) to the property ($\hat{p}$) of interest, we use multilayer perceptrons (MLPs), $\hat{p} = \text{MLP}(\mathbf{h}_G)$. The MLPs constitute four layers with #1: 64, #2: 32 #3: 16, #4: 1 neurons when mean, sum, or max pooling is used. When the set2set pooling is used, the MLP layers have #1: 128, #2: 64, #3: 32, #4: 1 neurons, because the output vector of the set2set method is twice its input size. For the set2set method, we set the number of processing steps T to 3.

We additionally test mean, sum, max pooling with MXMNet. We use the implementation and default hyperparameters with batch size 128 and global cutoff distance 5 as it was provided by the authors of MXMNet [52], cf. [53].

3.2 Illustrative case study: Molecular weight

To illustrate the importance of physical pooling functions on a simple example, we learn the molecular weight of alkanes. To compose the data set, we obtain

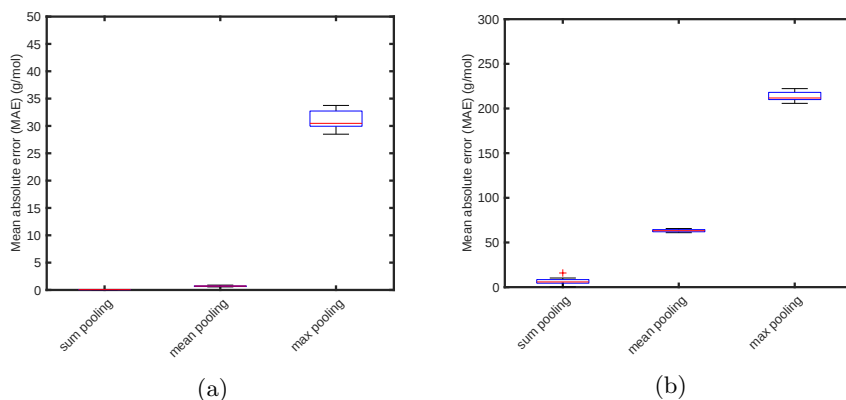


Fig. 2: Mean absolute error in g/mol for test of the 1-GNN with different pooling functions, namely: sum, mean, max. (a) test data set of alkanes with up to 30 C-atoms, (b) external data set of alkanes with 35 up to 60 C-atoms. Results are for ten independent training runs, each with 500 periods.

about 2,300 alkanes from C_1H_4 to $C_{60}H_{122}$ from the PubChem database [20] and compute their molecular weight using RDKit [13].

For illustration, we split this case study into two steps. Firstly, 1-GNNs with sum, mean, and max pooling functions are trained, validated, and tested on corresponding data sets with alkanes with up to 30 C-atoms. Secondly, the trained GNNs are tested against an external data set containing alkanes with more than 30 C-atoms. Thus, the generalization capabilities of the 1-GNNs are tested against extrapolated data.

The training of the 1-GNNs is repeated ten times for each pooling function with a maximum number of 500 epochs. The initial data set with up to 30 C-atoms is randomly split into 80% training, 10% validation, and 10% test sets for each run. Since we consider alkanes, we choose the attributes of the nodes in the molecular graph to include the hydrogen count only and do not use edge attributes, hence we replace the edge feature network in the message passing of the 1-GNN by a (learned) parameter matrix that is the same for all edges.

Figure 2 shows the test set performance of the 1-GNN with different pooling functions. Figure 2 (a) illustrates the test results for the data set of alkanes with up to 30 C-atoms. As expected, the sum pooling leads to the best performance on the test data set with an average mean absolute error of 0.06 g/mol as it captures the molecular size-dependent character of the molecular weights. In contrast, mean pooling leads to an average mean absolute error 0.7 g/mol. Max pooling even leads to an average absolute error in the order of 31 g/mol.

Figure 2 (b) shows the mean absolute error on the test data sets of alkanes with more than 30 C-atoms. The 1-GNN with sum pooling leads to an average absolute error of 6.5 g/mol. In contrast, the 1-GNN with mean pooling leads to an average error of 63 g/mol and the 1-GNN with max pooling leads to an average error of 213 g/mol.

The results clearly show that the sum pooling function, which was selected based on our physical insight, performs better than the unphysical mean and max pooling functions for the prediction of the molecular weight. In particular,

the sum pooling outperforms the unphysical pooling functions significantly when extrapolating the model. These results support our theoretical expectations that GNNs with unphysical pooling functions are more prone to overfitting. Notably, the mean absolute error of the GNN with mean pooling is much smaller on the test set with alkanes with less than 30 C-atoms compared to the test set with alkanes with more than 30 C-atoms. This result indicates that the selection of pooling functions based on the performance on a standard validation or test set could also be error-prone.

3.3 Physicochemical properties

We analyze the importance of physical pooling functions for a variety of relevant properties. In addition, we compare our results to a more complex set2set readout function and explore the MXMNet [52] architecture.

We use the QM9 data set to train our models [30, 33]. The experimental setup is twofold. First, the 1-GNNs with sum, mean, max, and set2set pooling functions are trained, validated, and tested on randomly selected subsets of the whole QM9 data set. This approach assesses the interpolation capabilities of the respective pooling functions. Second, we train, validate, and test the 1-GNN and MXMNet on the QM9 data excluding molecules with exactly 9 heavy atoms. These models are then tested against molecules with 9 heavy atoms from the QM9 data set. This approach is chosen to assess the generalization capabilities of the models in terms of extrapolation ability. For each training run, the data set is randomly split into 80% training, 10% validation, and 10% test sets. The training is stopped after 300 and 900 periods for the 1-GNN and MXMNet, respectively. The mean absolute error for the test set is reported based on the period with the lowest validation error.

3.3.1 Interpolation

The results for testing sum, mean, max, and set2set pooling function on the whole QM9 data set are summarized in Table 1. Overall, the results indicate that physically meaningful pooling functions lead to favorable performances on the QM9 data set for interpolation. It can be observed that for all molecular size-dependent properties, the 1-GNN with sum pooling significantly outperforms the 1-GNNs with mean and max pooling.

For the molecular size-independent properties, we do not observe a superior performance of one pooling function; all pooling functions result in similar accuracies.

3.3.2 Generalization with 1-GNN architecture

In order to analyze the generalization capability of the pooling functions on the QM9 data set, we train the 1-GNNs only on molecules with up to 8 heavy atoms, i.e., a maximum number of 8 C, N, O, F atoms. Then, we test the prediction accuracy on an internal test set, i.e., containing molecules with up to 8 heavy atoms, and also on a data set with the remaining QM9 molecules that have exactly 9 heavy atoms. The latter test set therefore tests the extrapolation capability of the derived GNNs. The results of the extrapolation are summarized in Table 2.

Table 1: Mean absolute errors averaged over three training runs for testing 1-GNN with different pooling functions, sum, mean, max, set2set against QM9 target properties. Properties are categorized into molecular size-independent (“m. size-ind.”) and molecular size-dependent (“m. size-dep.”) according to [35]. Errors of the best pooling function are bold type.

Target			Pooling			
			sum	mean	max	set2set
m. size-dep.	α	a_0^3	0.301	0.469	0.482	0.583
	R^2	a_0^2	21.5	25.0	24.1	24.1
	ZPVE	meV	9.39	24.16	26.99	21.96
	$C_{v,298}$	cal/mol K	0.149	0.203	0.197	0.197
	$U_{0,atom}$	eV	0.117	0.345	0.357	0.437
	$U_{298,atom}$	eV	0.117	0.363	0.366	0.386
	$H_{298,atom}$	eV	0.123	0.329	0.383	0.390
	$G_{298,atom}$	eV	0.112	0.293	0.328	0.313
m. size-ind.	μ	Debye	0.452	0.456	0.449	0.472
	ϵ_{HOMO}	meV	92.8	93.1	92.6	133.7
	ϵ_{LUMO}	meV	93.1	93.7	93.8	93.7
	$\Delta\epsilon$	eV	0.1322	0.1336	0.1283	0.1323

The interpolation performance of the models (indicated in black in Table 2) is similar to that of the previous models trained on the whole QM9 data set (cf. Table 1). Notably, the absolute errors increase for some properties, e.g., atomization energy, as the training set is much smaller. The QM9 data set contains about 108,000 molecules with 9 atoms and about 22,000 molecules with 1 to 8 heavy atoms.

For extrapolation (indicated in blue in Table 2), we find that sum pooling performs much better compared to mean, max, and set2set pooling on all tested molecular size-dependent properties. For the molecular size-independent properties we observe that the sum pooling does not outperform the other pooling functions anymore. Rather, mean and max pooling perform slightly better compared to sum pooling. Notably, the set2set method does not improve the accuracy compared to sum, mean, and max pooling for any property.

3.3.3 Generalization with MXMNet architecture

We also analyze the influence of the pooling function on the MXMNet GNN model [52]. The MXMNet model reached state-of-the-art performance on several prediction tasks in QM9 [52]. MXMNet includes directional information in its message passing process. For the readout step, sum pooling is applied in the original MXMNet model. We compare MXMNet performance with three different pooling functions: sum, mean, and max. Similar to the 1-GNN, we test MXMNet trained on molecules of QM9 with up to 8 heavy atoms against an internal test set and an external test set, i.e., extrapolating to molecules with 9 heavy atoms. The results are summarized in Table 3.

The interpolation performance of the MXMNet (indicated in black in Table 3) is highly favorably compared to the 1-GNN architecture, as expected.

Table 2: Mean absolute errors averaged over three independent training runs for testing 1-GNN with different pooling functions, sum, mean, max, and set2set, against: (black) QM9 data set excluding molecules with 9 heavy atoms, (blue) only molecules with 9 heavy atoms of the QM9 data set. The errors of the best pooling function are bold type. Units are equivalent to those in Table 1.

Target	1-GNN			
	sum	mean	max	set2set
α	0.302	0.816	0.727	1.164
	1.385	8.445	8.654	7.835
R^2	17.9	24.9	23.3	29.4
	83.6	236.0	229.5	227.0
ZPVE	0.0123	0.0561	0.0544	0.0547
	0.0288	0.3661	0.3127	0.4508
$C_{v,298}$	0.158	0.312	0.291	0.309
	0.650	3.509	3.360	3.129
$U_{0,atom}$	0.180	0.816	0.766	0.773
	1.365	8.082	8.100	9.613
$U_{298,atom}$	0.171	0.759	0.757	0.631
	1.239	8.368	8.117	8.220
$H_{298,atom}$	0.181	0.724	0.802	0.706
	1.197	8.275	8.125	10.424
$G_{298,atom}$	0.167	0.690	0.705	0.757
	1.142	7.671	7.363	8.751
μ	0.469	0.465	0.454	0.501
	0.588	0.572	0.569	0.611
ϵ_{HOMO}	0.110	0.109	0.112	0.127
	0.142	0.140	0.142	0.158
ϵ_{LUMO}	0.116	0.114	0.110	0.118
	0.179	0.171	0.167	0.173
$\Delta\epsilon$	0.161	0.156	0.150	0.156
	0.226	0.219	0.208	0.219

The MXMNet with sum pooling outperforms the other pooling approaches for all molecular size-dependent properties. This is in agreement with our expectations and previous observations on the 1-GNN architecture. For the molecular size-independent properties, we observe that sum, mean, and max pooling perform very similarly. Notably, the extrapolation performance of the MXMNet architecture (indicated in blue in Table 3) also significantly outperforms the 1-GNN architecture on all properties but R^2 .

The generalization results follow the same pattern as our previous observations. Again, we observe a significant advantage of sum pooling for all molecular size-dependent properties. For the molecular size-independent properties, we obtain similar performance of the pooling functions for LUMO. For μ , sum pooling performs only slightly better than mean and max pooling. For the HOMO and the HOMO-LUMO gap, however, mean and max pooling outperform sum pooling by a factor of more than 3. This demonstrates that also sum pooling can promote overfitting and thus prevent generalization in case of size-independent properties. Notably, the extrapolation error of the MXMNet is much larger for the unphysical pooling function compared to the extrapolation error of the simpler 1-GNN with the same pooling function. This indicates that the selection of physical pooling functions could be more important for more complex models.

4 Conclusion

GNNs have emerged as a promising deep learning technique for end-to-end molecular property prediction in chemical engineering. The selection of pooling functions in GNNs for property prediction should be based on physical knowledge because incorrect pooling functions can promote overfitting and weaken generalization. We identify the dependency of the learned property on the molecular size as key property for the selection of the pooling function: When a property is molecular size-dependent, the sum pooling function should be used. When a property is molecular size-independent, sum pooling can lead to poor generalization. We recommend to compare sum, mean, and max pooling functions for size-independent properties.

Our computational results support this hypothesis showing that physical GNN architectures generalize better than unphysical architectures. In future research, the physical selection of pooling functions should always be considered when predicting molecular properties with GNNs.

Acknowledgements

Supported by the German Research Foundation (DFG) within the framework of the Excellence Strategy of the Federal Government and the Länder - Cluster of Excellence 2186 "The Fuel Science Center" (ID390919832). Simulations were performed with computing resources granted by RWTH Aachen University under projects thes0682 and rwth0731. MD received funding from the Helmholtz Association of German Research Centers.

Table 3: Mean absolute errors averaged over three independent training runs for testing MXMNet with different pooling functions, sum, mean, max, against: (black) QM9 data set excluding molecules with 9 heavy atoms, (blue) only molecules with 9 heavy atoms of the QM9 data set. Error of best pooling function are bold type. Units are equivalent to those in Table 1.

Target	MXMNet		
	sum	mean	max
α	0.0781	0.1331	0.1347
	0.1887	1.2853	0.4569
R^2	1.78	2.55	2.77
	104.84	199.26	182.34
ZPVE	0.00144	0.00370	0.00278
	0.00226	0.04492	0.00807
$C_{v,298}$	0.0325	0.0503	0.0515
	0.0891	2.2484	0.6094
$U_{0,atom}$	0.0111	0.0728	0.0710
	0.0265	1.4095	0.5047
$U_{298,atom}$	0.0114	0.0729	0.0720
	0.0265	1.2072	0.5025
$H_{298,atom}$	0.0115	0.0723	0.0724
	0.0265	1.5855	0.5039
$G_{298,atom}$	0.0122	0.0662	0.0677
	0.0271	2.1216	0.5009
μ	0.0892	0.0981	0.1203
	0.1551	0.1708	0.1997
ϵ_{HOMO}	0.0516	0.0460	0.0550
	0.2273	0.0687	0.0743
ϵ_{LUMO}	0.0366	0.0384	0.0449
	0.0700	0.0707	0.0816
$\Delta\epsilon$	0.0766	0.0729	0.0783
	0.3896	0.1192	0.1263

Data and software availability

Our implementations are based on the models in PyTorch Geometric developed by Fey & Lenssen [9]. The model implementation is available at https://git.rwth-aachen.de/avt-svt/public/graph_neural_network_for_fuel_ignition_quality under Eclipse Public License 2.0 (cf. [38]). The MXMNet implementation is provided by the authors of MXMNet [52], cf. [53]. We use the QM9 data set to train our models [30, 33].

Authors contributions

Artur M. Schweidtmann: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Writing - Original Draft, Writing - Review & Editing, Visualization. **Jan G. Rittig:** Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization. **Jana M. Weber:** Methodology, Validation, Formal analysis, Writing - Review & Editing. **Martin Grohe:** Methodology, Validation, Formal analysis, Writing - Review & Editing. **Manuel Dahmen:** Methodology, Validation, Formal analysis, Writing - Review & Editing, Supervision. **Kai Leonhard:** Methodology, Validation, Formal analysis, Writing - Review & Editing. **Alexander Mitsos:** Methodology, Validation, Formal analysis, Resources, Writing - Review & Editing, Supervision, Project administration, Funding acquisition.

References

- [1] P. Atkins and R. Friedman. *Molecular Quantum Mechanics*. Oxford University Press, Oxford, 5th ed. edition, 2011.
- [2] S. Back, J. Yoon, N. Tian, W. Zhong, K. Tran, and Z. W. Ulissi. Convolutional neural network of atomic surface structures to predict binding energies for high-throughput screening of catalysts. *The journal of physical chemistry letters*, 10(15):4401–4408, 2019.
- [3] S. W. Benson, F. R. Cruickshank, D. M. Golden, G. R. Haugen, H. E. O’Neal, A. S. Rodgers, R. Shaw, and R. Walsh. Additivity rules for the estimation of thermochemical properties. *Chemical Reviews*, 69(3):279–324, 1969.
- [4] D. Bonchev and D. Rouvray. *Chemical Graph Theory: Introduction and Fundamentals*. Gordon and Breach Science Publishers, New York, United States, 1991.
- [5] C. Chen, W. Ye, Y. Zuo, C. Zheng, and S. P. Ong. Graph networks as a universal machine learning framework for molecules and crystals. *Chemistry of Materials*, 31(9):3564–3572, 2019.
- [6] E. R. Cohen, T. Cvitas, J. G. Frey, B. Holström, K. Kuchitsu, R. Marquardt, I. Mills, F. Pavese, M. Quack, J. Stohner, H. L. Strauss, M. Takami, and A. J. Thor, editors. *Quantities, Units and Symbols in Physical Chemistry*. The Royal Society of Chemistry, 2007.

- [7] C. W. Coley, R. Barzilay, W. H. Green, T. S. Jaakkola, and K. F. Jensen. Convolutional embedding of attributed molecular graphs for physical property prediction. *Journal of Chemical Information and Modeling*, 57(8):1757–1772, 2017.
- [8] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, pages 2224–2232. Curran Associates, Inc, 2015.
- [9] M. Fey and J. E. Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428v3*, 2019. arXiv.
- [10] B. Fiedler and A. Schuppert. Local identification of scalar hybrid models with tree structure. *IMA Journal of Applied Mathematics*, 73(3):449–476, 2008.
- [11] R. Gani, B. Nielsen, and A. Fredenslund. A group contribution approach to computer-aided molecular design. *AIChE Journal*, 37(9):1318–1332, 1991.
- [12] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212v2*, 2017. arXiv.
- [13] Greg Landrum. Rdkit: Open-source cheminformatics. accessed on 05.04.2020.
- [14] K. Gubaev, E. V. Podryabinkin, and A. V. Shapeev. Machine learning of molecular properties: locality and active learning. *The Journal of Chemical Physics*, 148(24):241727, 2018.
- [15] W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 1024–1034, 2017.
- [16] W. L. Hamilton, R. Ying, and J. Leskovec. Representation learning on graphs: Methods and applications.
- [17] K. G. Joback and R. C. Reid. Estimation of pure-component properties from group-contributions. *Chemical Engineering Communications*, 57(1-6):233–243, 1987.
- [18] A. R. Katritzky, V. S. Lobanov, and M. Karelson. Qspr: the correlation and quantitative prediction of chemical and physical properties from structure. *Chemical Society Reviews*, 24(4):279, 1995.
- [19] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, and P. Riley. Molecular graph convolutions: Moving beyond fingerprints. *Journal of Computer-Aided Molecular Design*, 30(8):595–608, 2016.

- [20] S. Kim, P. A. Thiessen, E. E. Bolton, J. Chen, G. Fu, A. Gindulyte, L. Han, J. He, S. He, B. A. Shoemaker, J. Wang, B. Yu, J. Zhang, and S. H. Bryant. Pubchem substance and compound databases. *Nucleic acids research*, 44(D1):D1202–13, 2016.
- [21] J. Klicpera, J. Groß, and S. Günnemann. Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123*, 2020.
- [22] Y. Li, P. Li, X. Yang, C.-Y. Hsieh, S. Zhang, X. Wang, R. Lu, H. Liu, and X. Yao. Introducing block design in graph neural networks for molecular properties prediction. *Chemical Engineering Journal*, 414:128817, 2021.
- [23] Z. Liu, L. Lin, Q. Jia, Z. Cheng, Y. Jiang, Y. Guo, and J. Ma. Transferable multilevel attention neural network for accurate prediction of quantum chemistry properties via multitask learning. *Journal of chemical information and modeling*, 61(3):1066–1082, 2021.
- [24] C. Lu, Q. Liu, C. Wang, Z. Huang, P. Lin, and L. He. Molecular property prediction: A multilevel quantum interactions modeling perspective. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:1052–1060, 2019.
- [25] K. J. Miller and J. A. Savchik. A new empirical method to calculate average molecular polarizabilities. *J. Am. Chem. Soc.*, 101:7206, 1979.
- [26] C. Morris, M. Ritzert, M. Fey, W. Hamilton, J. Lenssen, G. Rattan, and M. Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence, 27.01.-01.02.2019, Honolulu, Hawaii, United States*, volume 4602-4609. AAAI Press, 2019.
- [27] M. Niepert, M. Ahmed, and K. Kutzkov. Learning convolutional neural networks for graphs. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2014–2023, New York, New York, USA, 20–22 Jun 2016. PMLR.
- [28] W. Pronobis, K. T. Schütt, A. Tkatchenko, and K.-R. Müller. Capturing intensive and extensive dft/tddft molecular properties with machine learning. *The European Physical Journal B*, 91(8):338, 2018.
- [29] D. C. Psychogios and L. H. Ungar. A hybrid neural network-first principles approach to process modeling. *AIChE Journal*, 38(10):1499–1511, 1992.
- [30] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1:140022 EP –, 2014.
- [31] J. G. Rittig, K. B. Hicham, A. M. Schweidtmann, M. Dahmen, and A. Mitsos. Graph neural networks for temperature-dependent activity coefficient prediction of solutes in ionic liquids. *arXiv preprint arXiv:2206.11776*, 2022.

- [32] J. G. Rittig, M. Ritzert, A. M. Schweidtmann, S. Winkler, J. M. Weber, P. Morsch, K. A. Heufer, M. Grohe, A. Mitsos, and M. Dahmen. Graph machine learning for design of high-octane fuels. *arXiv preprint arXiv:2206.00619*, 2022.
- [33] L. Ruddigkeit, R. van Deursen, L. C. Blum, and J.-L. Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *Journal of chemical information and modeling*, 52(11):2864–2875, 2012.
- [34] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko. Quantum-chemical insights from deep tensor neural networks. *Nature Communications*, 8(1):13890, 2017.
- [35] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller. Schnet - a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24):241722, 2018.
- [36] K. T. Schütt, A. Tkatchenko, and K.-R. Müller. Learning representations of molecules and materials with atomistic neural networks.
- [37] A. M. Schweidtmann, E. Esche, A. Fischer, M. Kloft, J.-U. Repke, S. Sager, and A. Mitsos. Machine learning in chemical engineering: A perspective. *Chemie Ingenieur Technik*, 2021.
- [38] A. M. Schweidtmann, J. G. Rittig, A. König, M. Grohe, A. Mitsos, and M. Dahmen. Graph neural networks for prediction of fuel ignition quality. *Energy & Fuels*, 34(9):11395–11407, 2020.
- [39] H. Shindo and Y. Matsumoto. Gated graph recursive neural networks for molecular property prediction.
- [40] K. Stephan and F. Mayinger. *Thermodynamik. Grundlagen und technische Anwendungen: Band 1: Einstoffsysteme*, volume 1. Springer-Verlag, 2013.
- [41] R. Todeschini and V. Consonni. *Handbook of Molecular Descriptors*. Wiley-VCH Verlag GmbH, Weinheim, Germany, 2000.
- [42] O. T. Unke and M. Meuwly. Physnet: A neural network for predicting energies, forces, dipole moments, and partial charges. *Journal of Chemical Theory and Computation*, 15(6):3678–3693, 2019.
- [43] O. Vinyals, S. Bengio, and M. Kudlur. Order matters: Sequence to sequence for sets.
- [44] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2021.
- [45] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande. Moleculenet: A benchmark for molecular machine learning. *Chemical Science*, 9(2):513–530, 2018.

- [46] T. Xie and J. C. Grossman. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Physical review letters*, 120(14):145301, 2018.
- [47] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.
- [48] K. Yang, K. Swanson, W. Jin, C. Coley, P. Eiden, H. Gao, A. Guzman-Perez, T. Hopper, B. Kelley, M. Mathea, A. Palmer, V. Settels, T. Jaakkola, K. Jensen, and R. Barzilay. Analyzing learned molecular representations for property prediction. *Journal of Chemical Information and Modeling*, 59(8):3370–3388, 2019.
- [49] S. Ye, J. Liang, R. Liu, and X. Zhu. Symmetrical graph neural network for quantum chemistry, with dual r/k space.
- [50] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Advances in Neural Information Processing Systems*, pages 4800–4810, 2018.
- [51] M. Zhang, Z. Cui, M. Neumann, and Y. Chen. An end-to-end deep learning architecture for graph classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 2018.
- [52] S. Zhang, Y. Liu, and L. Xie. Molecular mechanics-driven graph neural network with multiplex graph for molecular structures. In *NeurIPS-W*, 2020.
- [53] S. Zhang, Y. Liu, and L. Xie. Molecular mechanics-driven graph neural network with multiplex graph for molecular structures. <https://github.com/zetayue/MXMNet>, 2020. accessed on 10.06.2021.