Modelling local and general quantum mechanical properties with attention-based pooling

David Buterez[1], Jon Paul Janet[2], Steven J. Kiddle[3], Dino Oglic[4], and Pietro Liò[1]

[1]Dept. of Computer Science and Technology, University of Cambridge, Cambridge, UK (db804 @ cam.ac.uk)
[2]Molecular AI, Discovery Sciences, R&D, AstraZeneca, Gothenburg, Sweden
[3]Data Science & Advanced Analytics, Data Science & AI, R&D, AstraZeneca, Cambridge, UK
[4]Center for AI, Data Science & AI, R&D, AstraZeneca, Cambridge, UK

Abstract

Atom-centred neural networks represent the state-of-the-art for approximating the quantum chemical properties of molecules, such as internal energies. While the design of machine learning architectures that respect chemical principles has continued to advance, the final atom pooling operation that is necessary to convert from atomic to molecular representations in most models remains relatively undeveloped. The most common choices, sum and average pooling, compute molecular representations that are naturally a good fit for many physical properties, while satisfying properties such as permutation invariance which are desirable from a geometric deep learning perspective. However, there are growing concerns that such simplistic functions might have limited representational power, while also being suboptimal for physical properties that are highly localised or intensive. Based on recent advances in graph representation learning, we investigate the use of a learnable pooling function that leverages an attention mechanism to model interactions between atom representations. The proposed pooling operation is a drop-in replacement requiring no changes to any of the other architectural components. Using SchNet and DimeNet++ as starting models, we demonstrate consistent uplifts in performance compared to sum pooling and a recent physics-aware pooling operation designed specifically for orbital energies, on several datasets, properties, and levels of theory, with up to 85% improvements depending on the specific task.

1 Introduction

Geometric deep learning (GDL) approaches are increasingly used across the life sciences, with remarkable potential and achievements in computational biology (analysing single-cell sequencing data [1, 2]), structural biology (prediction of protein structures [3] and protein sequence design [4]), drug discovery [5] and simulating rigid and fluid dynamics [6] being only a few examples. The simple but powerful formulation of GDL methods such as graph neural networks (GNN) motivated the investigation of long-standing problems from a new perspective, particularly in fields such as computational chemistry where the GDL abstractions can be naturally applied to objects like atoms and molecules (nodes and graphs), as well as their interactions (edges).

Approximating quantum mechanical properties using machine learning (ML) is of significant interest for applications in catalysis, material and drug design [7, 8]. However, traditional physics-based methods are severely limited by computational requirements that scale poorly with system size [9]. In the pursuit of accurate, scalable, and generalisable ML models, several different strategies have been proposed. One way to support this vision is by taking a purely data-driven approach and developing quantum machine learning (QML) models based on accurate physical methods such as density functional theory (DFT) combined with large and diverse collections of data (e.g. QM9 [10], QMugs [11], nablaDFT [12] and QM7-X [13]). Another approach is to devise transfer learning datasets and algorithms that can extract useful patterns from less accurate, but cheaper and more scalable simulations that ultimately benefit predictions at a higher fidelity level ([11, 8, 14]). At the same time, advances in GNN architectures and the ability to exploit specific features of quantum data such as atom positions and directional information such as bond angles and rotations are an active area of research that have produced state-of-the-art models ([15, 16, 17, 18]).

Most of the GNN advances have focused on more expressive ways of defining atom (node) representations and local interactions in increasingly large neighbourhoods centred on each of the nodes ('k-hops'). For example, SchNet [15] starts with atom embeddings based on the atom type and atom-wise layers

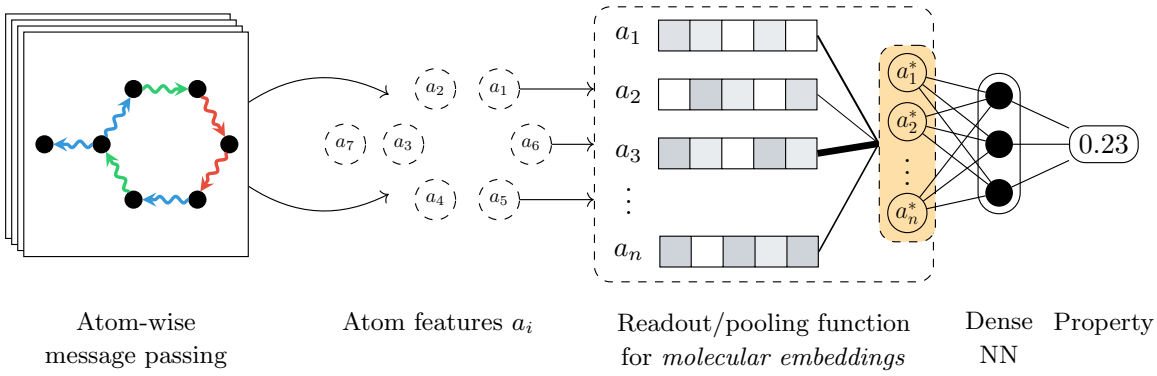

Figure 1: A common step in most atom-centred neural networks is the aggregation or pooling of learnt atom features into a molecule-level representation through a dedicated function (highlighted with a dashed box). Traditionally, simple functions that satisfy permutation invariance such as sum, mean, or maximum are used for this step. Alternatively, a more expressive molecular representation can be computed by neural networks, for example using attention to discover the most relevant atomic features.

implemented as linear transformations that are combined with convolutional layers satisfying rotational invariance. DimeNet [16] formulates the task as a message passing exercise, while also introducing directionality by considering the angles between atoms. Instead of atom embeddings, DimeNet computes directional embeddings between pairs of atoms j, i that incorporate atomic distances and angles by aggregating other embeddings directed towards the source atom j. The construction also guarantees invariance to rotations. Furthermore, instead of using raw angles, DimeNet represents distances and angles through a spherical 2D Fourier-Bessel basis, a physics-inspired decision that was also empirically found to be preferable. The original DimeNet architecture was subsequently updated to a faster and more accurate model denoted DimeNet++ by replacing costly operations with fast and expressive alternatives [17]. Recently, GDL architectures that are invariant to translations, rotations, and reflections such as E(n) GNNs have proven competitive in the prediction of quantum mechanical properties [18].

Even with the accelerated development of QML methods and the heterogeneity of recent approaches, a common element for most QML models is that they naturally operate at the level of atom representations, for example through message passing steps. However, many prediction targets of interest are formulated at the molecular level. e.g. total energy, dipole moment, highest occupied molecular orbital (HOMO) energy, (lowest unoccupied molecular orbital) LUMO energy, etc. Thus, an aggregation scheme must be used to combine the atom representations into a single molecule-level representation. This task is typically handled with simple fixed pooling functions like sum, average, or maximum. Despite their appealing simplicity, there are growing concerns regarding the representational power of this class of functions [19, 20]. In the following section, we also discuss the concurrently-developed Orbital Weighted Average (OWA), a physics-based method designed specifically for orbital properties and which also seeks to improve upon the standard pooling operators by exploiting the local and intensive character of the target property [20]. Buterez et al. also highlighted the lacklustre performance of standard pooling functions in a variety of settings, but particularly on challenging molecular properties [21]. As an alternative to standard pooling, the authors proposed replacing the fixed functions with learnable functions implemented as neural networks. When applied to conventional message passing architectures (GCN [22], GAT [23] and GATv2 [24], GIN [25], PNA [26]) that operate on the molecular graph with node features extracted from the SMILES [27] representation, neural pooling functions provided significant uplifts in performance and faster convergence times.

Apart from expressive power, the standard pooling functions are also widely used thanks to being permutation invariant with respect to the order of the atom representations that are being aggregated. Furthermore, these simple operations are also usually aligned with fundamental physical principles. For example, the total energy, a molecular property, can be obtained as the sum of the atom energies. In general, molecular properties that scale linearly with the number of atoms can be well approximated by fixed functions such as sum or average. However, it is not uncommon for the target property to behave non-linearly or be localised around a small subset of atoms which determine its value. Bioaffinity (the achieved level of inhibition or activation of a drug-like molecule against a protein target) is a property where we can reasonably expect that most of the effect comes from an active group of atoms [21]. In QML, a canonical example of a localised property is the HOMO energy.

In this work, we investigate the use of an attention-based pooling function on atomistic systems for the prediction of general and localised quantum properties with state-of-the-art 3D-coordinate aware models (the high-level workflow is illustrated in Figure 1). The chosen design satisfies a collection of desirable features, some of which were previously mentioned, namely **(i)** permutation invariance with respect to node (atom) order, **(ii)** increased representational power compared to standard pooling operators thanks to the underlying neural networks, **(iii)** the ability to model arbitrary, potentially long-range or localised relationships due to the attention mechanism, and **(iv)** generality and simplicity; the proposed method is applicable to any molecular property (including quantum properties), and can be used as drop-in replacement on any architecture that uses traditional pooling methods without any modifications to the model itself.

The current work represents an extension to previous work which considered only 2D molecular graphs, but demonstrated the potential of attention-based pooling for predicting properties such as the HOMO energy [21]. Here, we demonstrate consistent uplifts in performance (as measured by the Mean Absolute Error – MAE) compared to sum pooling, chosen as a representative of the established methods, on a selection of standard datasets of different sizes and simulated at different levels of theory, including QM7b (7,211 molecules) [28, 29, 30], QM8 (21,786 molecules) [31, 32], QM9 (130,831 molecules) [31, 10], QMugs (665K molecules) [11], and the CCSD and CCSD(T) datasets from MD17 (1,500 molecules for Aspirin, Benzene, Malonaldehyde, Toluene, and 2,000 for Ethanol) [33]. Sum pooling is the default choice in many implementations (e.g. SchNet, DimeNet), usually outperforming mean and maximum pooling [19] or matching them [21]. We also evaluate the proposed methods against OWA on the OE62 [34] dataset and conclude that attention-based pooling can match and outperform OWA depending on the configuration (e.g. number of attention heads). While our method introduces a large number of learnable parameters, this is normal for a standard attention implementation and does not significantly affect training times or introduce overfitting.

2 Methodology

2.1 Pooling functions

We start by assuming an atomistic model that operates on positional inputs (distances, angles, etc.) and which computes individual representations that require aggregation into a single molecule-level embedding. The specifics of the architecture or the implementation do not matter as long as the assumptions hold. For example, many message passing neural networks can be summarised into the following generic formulation that computes node-level features $\mathbf{h}_u$ [35]:

$$\mathbf{h}_a = \phi\left(\mathbf{x}_a, \bigoplus_{v \in \mathcal{N}_a} \psi(\mathbf{x}_a, \mathbf{x}_v)\right) \tag{1}$$

where a, v are nodes (atoms), $\mathbf{x}_i$ are atom representations, $\mathcal{N}_i$ is the 1-hop neighbourhood of atom i, $\oplus$ is a node-level aggregation function such as sum or average, and ϕ, ψ are learnable functions such as multi-layer perceptrons (MLPs). It should be noted that there are many variations and extensions of Equation (1), and this is only an example of a possible architecture. Importantly, once the message passing steps or equivalent updates are done, the atom-level representations are aggregated into a molecule-level representation $\mathbf{h}_m = \bigoplus_{i \in \mathcal{V}} (\mathbf{h}_i)$, where $\mathcal{V}$ is the collection of atoms in the molecule (note that this $\bigoplus$ can be different from the one in Equation (1)). This operation is often called an *aggregation*, *pooling*, or *readout* function.

2.2 Attention-based pooling

To design an expressive pooling function that considers the entire context of the molecule (i.e. all computed atom representations) and is self-contained (does not require any additional inputs), we leverage the existing Set Transformer framework introduced by Lee et al. for set modelling [36], and proposed by Buterez et al. for use in node aggregation [21]. In other words, the pooling operation is reframed as a set summarisation task, with an output that corresponds to the desired molecular embedding. This is achieved by assembling building blocks defined using a standard multi-head attention mechanism:

$$\text{Attention}(Q, K, V) = \omega(QK^\top)V \tag{2}$$

$$\text{MultiHeadAttention}(Q, K, V) = \text{Concatenate}(H_1, ..., H_m)W^O \tag{3}$$

$$\text{where } H_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{4}$$

The standalone attention module Attention$(\cdot,\cdot,\cdot)$ receives input *query*, *key*, and *value* vectors, of dimension d_k, d_k, and d_v respectively and gathered in matrices Q, K, V, respectively. With $\omega(\cdot) = \text{softmax}(\cdot/\sqrt{d_k})$, the attention operation computes a weighted sum of the values where a large query-key dot product assigns a larger weight to the corresponding value. In multi-head attention, Q, K, V are projected to new dimensions by learnt projections W_i^Q, W_i^K, W_i^V, respectively, for a total of m independent times. The results are processed by an attention module, with the concatenated output attention *heads* being projected with W^O.

Following the original Set Transformer implementation, we use the lower-level multihead and set attention blocks (MABs, respectively SABs) and pooling by multihead attention (PMA) to define an encoder-decoder architecture that embeds input atom vectors into a chosen dimension d, then learns to aggregate or compress the encoded representations into a single vector, the *molecule representation*. The encoder is defined as

$$\text{MAB}(X, Y) = H + \text{Linear}_\phi(H) \tag{5}$$
$$H = X + \text{MultiHeadAttention}(X, Y, Y) \tag{6}$$
$$\text{SAB}(X) = \text{MAB}\,(X, X) \tag{7}$$
$$\text{Encoder}(X) = \text{SAB}^n\,(X) \tag{8}$$

with a decoder:

$$\text{PMA}_k(Z) = \text{MAB}\,(S_k, \text{Linear}_\phi(Z)) \tag{9}$$
$$\text{Decoder}(Z) = \text{Linear}_\phi\,(\text{SAB}^n\,(\text{PMA}_k(Z))) \tag{10}$$

Here, Linear_ϕ denotes a linear layer followed by an activation function ϕ, $\text{SAB}^n(\cdot)$ represents n subsequent applications of a SAB, and S_k is a collection of learnable k seed vectors that are randomly initialised (PMA_k outputs k vectors). The resulting Set Transformer module can be used as a pooling function by encoding all the atomic representations into features Z that are transformed into a single-vector representation by the decoder. Here, we refer to this pooling function as attention-based pooling (ABP):

$$\text{ABP}(X) = \text{Decoder}(\text{Encoder}(X)) \tag{11}$$

2.3 Orbital Weighted Average pooling

Chen et al. have concurrently observed that the standard pooling functions (sum, average, maximum) might not accurately describe physical properties that are highly localised and intensive, such as orbital properties, and in particular the HOMO energy [20]. Instead, they discuss the importance of pooling functions that can attribute different weights or 'importance' for a subset of atomic representations. For example, the softmax function $\text{softmax}(\epsilon_1, ..., \epsilon_n) = \sum_{i=1}^{n} \frac{\exp(\epsilon_i)}{\sum_{j=1}^{n}\exp(\epsilon_j)}$, where ϵ_i are atomic representations of an n-atom system that in this case are assumed to be scalars. The general form is given by weighted average (WA) pooling:

$$f_{\text{WA}} = \sum_{i=1}^{n} w_i \epsilon_i \tag{12}$$

where additionally we assume that the learnable weights w_i are normalised by softmax to sum to 1.

From a physical perspective, the weights that the neural network will learn for HOMO energy prediction should tend towards the orbital coefficients l_i that describe the fraction of the orbital that is localised on a given atom i. To incorporate this idea into the pooling function, Chen et al. propose the following strategy:

1. Pre-compute the orbital coefficients for the dataset (offline)
2. Use a separate atomistic model to learn the weights for f_{WA}, which are forced to be close to the pre-computed coefficients by an updated loss function:

$$\mathcal{L}_{\text{OWA}} = \frac{1}{n_{\text{train}}}\left[\alpha \sum_{A=1}^{n_{\text{train}}}\left(E^A_{\text{HOMO}} - \sum_{i=1}^{n_A} w_{(A,i)}\epsilon_{(A,i)}\right)^2 + \beta \sum_{A=1}^{n_{\text{train}}}\sum_{i=1}^{n_A}(l_{(A,i)} - w_{(A,i)})^2\right] \tag{13}$$

where n_{train} denotes the number of training systems A in the dataset, n_A is the number of atoms in a system A, and E^A_{HOMO} is the target HOMO energy for a system A. The resulting pooling function with the learnt weights is denoted by f_{OWA} (orbital weighted average).

3 Design and implementation

As stated in *Methodology*, the proposed attention-based pooling function can be applied to a variety of atomistic modelling algorithms. Here, we chose to evaluate our methods using two architectures: SchNet and DimeNet++, which were briefly discussed in the introduction. Both models are widely-known and used, making them easily accessible in general purpose deep learning libraries such as PyTorch Geometric (used here) [37, 38]. Furthermore, DimeNet++ is a particularly competitive model which outperforms both contemporary and newer models (e.g. E(n) GNNs [18]).

For our evaluation, we chose sum pooling as a representative of the standard pooling methods. It is the default choice for SchNet and DimeNet++, and in our previous extensive evaluation of graph pooling functions we did not observe significantly better performance for any of the three functions [21]. Furthermore, from a physical perspective sum pooling can be considered a natural choice for approximating certain quantum properties.

Here, we have used the PyTorch Geometric implementations of SchNet and DimeNet++, modified to support attention-based pooling. As of PyTorch Geometric version 2.3.0 (available at the time of writing), the proposed pooling function is natively available as `SetTransformerAggregation` (based on our implementation). Unless otherwise noted, we use relatively deep models to ensure that the atom-level representations learnt before pooling are expressive enough. In particular, we use SchNet models with 256 filters, 256 hidden channels (hidden embedding size), and 8 interaction blocks, and otherwise default parameters (total parameter count before pooling: 2.3 million), and DimeNet++ models with 256 hidden channels (hidden embedding size), 6 interaction blocks, an embedding size of 64 in the interaction blocks, a basis embedding size of 8 in the interaction blocks, an embedding size of 256 for the output blocks, and otherwise default parameters (total parameter count before pooling: 5.1 million). The models chosen here are larger than the defaults in PyTorch Geometric and the original DimeNet study [16]. In addition, while it is common to output scalar representations for the atoms, we keep the same dimension for the atom representations as used inside the models (before output), i.e. 256. This ensures that the attention-based pooling can benefit from the full representation, although in principle it is possible to apply it to scalars. We follow the output of attention-based pooling with a small MLP, obtaining a scalar prediction.

We evaluate our methods on all the properties of the QM7b and QM8 datasets, on HOMO and LUMO energy prediction for QM9 and QMugs, as well as on energy prediction tasks from MD17, which provide a challenging setting due to the limited amount of data. We have also considered total energy prediction for QMugs as an example of a non-local property on the largest and most diverse of the available datasets. Results are provided for both SchNet and DimeNet++, with sum and attention-based pooling, i.e. 4 results per (property, dataset) pair. A batch size of 128 is used for all models. To ensure an accurate and self-contained comparison, we randomly generate 5 different train, validation, and test splits for each dataset using a ratio of 80%/10%/10%, and report the average MAE ± standard deviation. The MAE is used as it is widely used in the literature to evaluate atomistic models on quantum property prediction. Since ABP introduces several hyperparameters compared to standard functions, we evaluate a small set of common hyperparameter choices and select the best configurations according to the validation set.

We also compare attention-based pooling with OWA on the OE62 dataset used by Chen et al. on both HOMO and LUMO energy. For this comparison, we use the provided OWA source code as a starting point, and use the same underlying SchNet implementation provided by the `schnetpack` 1.0.1 library [39], including the same SchNet hyperparameters (embedding size of 128, 128 filters, 64 Gaussian functions, 6 interaction blocks, a cutoff radius of 5, followed by 4 atom-wise layers with an input of 128 features). A batch size of 40 was used, as larger models would run out of memory when using hardware equipped with 32GB of video memory. For attention-based pooling, we modified the atom-wise component to output representations of the same dimensionality as the inputs, as described above. We also generated 5 random splits using the same ratio as Chen et al. (32,000 molecules for train, 19,480 for validation, and 10,000 for test), and report both the MAE and the RMSE (root mean squared error).

4 Results

Our results indicate that attention-based pooling outperforms sum pooling on the majority of datasets and quantum properties, including properties computed at different levels of theory (Table 1). On QM7b, using ABP on top of SchNet results in an average decrease (across all tasks) in MAE of 50.5%, with the highest decrease on the 'Polarizability (self-consistent screening)' task (85.13%). The smallest decrease in

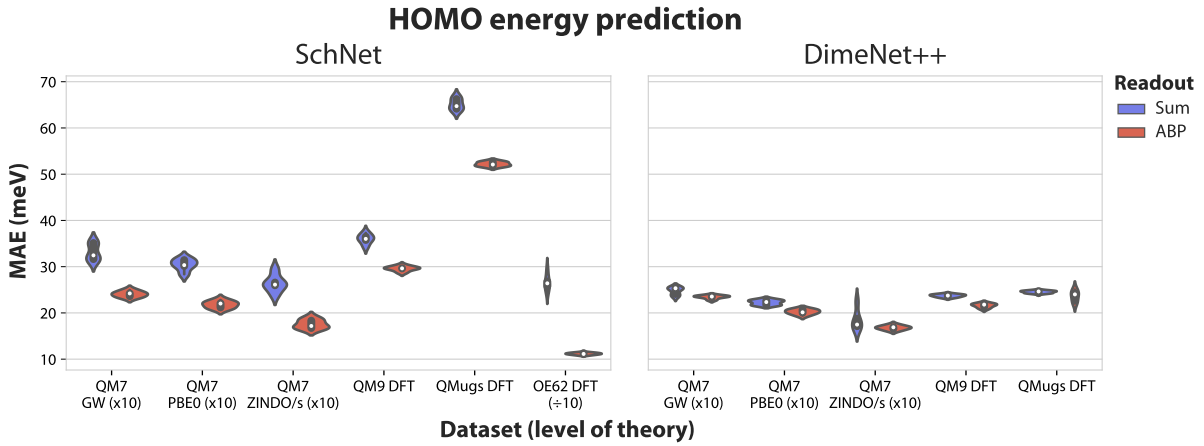

Figure 2: SchNet and DimeNet++ models evaluated on HOMO energy prediction on different datasets (QM7 with different levels of theory, QM9, and QMugs), with the mean absolute error reported on test sets corresponding to five different random splits for each dataset. The metrics of some datasets are scaled by 10 to ensure a similar scale for all datasets (indicated by '(x10)' or '(÷10)'). The exact metrics are reported in Tables 1 and 2.

MAE is observed for 'Atomization energy (DFT/PBE0)' (23.98%). When using DimeNet++, there is a more modest average decrease in MAE of 14.64%, with the most improved task being 'Atomization energy (DFT/PBE0)' (58.41%). The only QM7b task where ABP does not improve performance is 'Maximal absorption intensity (ZINDO)' (−0.37%) when using DimeNet++. The ABP-based DimeNet++ models generally matches the ABP-based SchNet models, suggesting that we are reaching the performance ceiling for these model configurations and tasks.

On QM8, there is an average decrease in MAE of 19.23% across all tasks for SchNet, and all tasks are improved when using ABP. The most improved task is 'E2-PBE0/def2SVP' (25.07%), and the least is 'f2-CC2' (13.15%). When using DimeNet++, the average decrease in MAE due to ABP is of 2.31%, with the most improved task being 'E1-CAM' (7.69%). We observed slightly worse performance when using ABP for only two tasks: 'f2-CAM' (−1.11%) and 'f2-PBE0/def2TZVP' (−0.48%). In general, for both SchNet and DimeNet++, the least improved tasks when using ABP involve the oscillator strength f_2.

For the larger quantum datasets (QM9 and QMugs), we train and evaluate models for the HOMO and LUMO energy prediction tasks. For HOMO, on QM9 we notice decreases in MAE of 21.67% and 9.89% for SchNet and DimeNet++, respectively. On QMugs, the decreases are of 24.77% and 3.92% for SchNet and DimeNet++, respectively. Similar uplifts are observed for LUMO. Interestingly, total energy prediction on QMugs is improved by a large amount (58.97%) on SchNet, and by a moderate amount on DimeNet++ (19.34%), despite not being a local property like the HOMO or LUMO energies.

For the small MD17 datasets, we observe a decrease in MAE for Aspirin of 15.25% and 8.38% for SchNet and DimeNet++, respectively, for Benzene of 36.37% and 5.01%, for Ethanol of −9.58% and 6.45%, for Malonaldehyde of −5.23% and −0.27%, and for Toluene of 37.21% and 2.45%. For these smaller datasets, we used SchNet with 128 filters, 128 hidden channels, and 4 interaction blocks, and DimeNet++ models with 128 hidden channels, 4 interaction blocks, and an embedding size of 128 for the output blocks. For the cases where using ABP did not improve the MAE (e.g. SchNet on Ethanol), we noticed that different underlying architectures can help improve upon the sum pooling result (usually more complex models with more interaction blocks).

To further validate the performance of ABP compared to sum pooling for each algorithm (i.e. SchNet and DimeNet++ in Table 1), we performed Wilcoxon signed-rank tests as the data is not normally distributed according to the `normaltest` function available in `scipy`: $p = 3.58 \times 10^{-11}$ (SchNet sum), $p = 2.4 \times 10^{-12}$ (SchNet ABP), $p = 5.73 \times 10^{-9}$ (DimeNet++ sum), $p = 8.54 \times 10^{-9}$ (DimeNet++ ABP). The Wilcoxon tests indicated statistical significance for SchNet ($p = 5.4 \times 10^{-6}$) and DimeNet++ ($p = 1.65 \times 10^{-5}$).

When compared to OWA pooling for HOMO energy prediction (Table 2), attention-based pooling matches or even slightly outperforms OWA depending on the ABP configuration, despite not leveraging pre-computed orbital coefficients. This can be observed both in terms of RMSE (the metric chosen by Chen

Table 1: Test MAE (mean ± standard deviation from 5 data random splits) for QM7b, QM8, QM9, QMugs, and MD17 (best MAEs in bold). Abbreviations: maximal absorption, MA; self-consistent screening, SCS; atomization, atom.; excitation, exc.; ionization, ion.; units, u.; malondialdehyde, MDA.

QM7b	**SchNet**		**DimeNet++**	
Task – level of theory (unit)	**Sum**	**ABP**	**Sum**	**ABP**
Atom. energy – ZINDO/s (meV)	3.859 ± 1.526	**3.113 ± 0.803**	4.268 ± 0.806	**2.694 ± 0.401**
Electron affinity – ZINDO (meV)	2.045 ± 0.240	**1.260 ± 0.176**	1.358 ± 0.074	**1.236 ± 0.057**
Exc. energy at MA – ZINDO (meV)	40.196 ± 2.018	**29.916 ± 1.118**	32.359 ± 0.681	**29.696 ± 0.658**
First exc. energy – ZINDO (meV)	2.429 ± 0.124	**1.493 ± 0.064**	1.571 ± 0.130	**1.422 ± 0.049**
HOMO – GW (meV)	3.311 ± 0.190	**2.405 ± 0.065**	2.475 ± 0.085	**2.342 ± 0.037**
HOMO – PBE0 (meV)	3.039 ± 0.131	**2.178 ± 0.078**	2.224 ± 0.056	**2.022 ± 0.055**
HOMO – ZINDO/s (meV)	2.639 ± 0.191	**1.752 ± 0.098**	1.849 ± 0.236	**1.678 ± 0.046**
Ion. potential – ZINDO/s (meV)	4.145 ± 0.326	**2.697 ± 0.183**	3.131 ± 0.206	**2.647 ± 0.152**
LUMO – GW (meV)	3.198 ± 0.313	**2.201 ± 0.180**	2.267 ± 0.179	**2.190 ± 0.144**
LUMO – PBE0 (meV)	2.042 ± 0.153	**1.499 ± 0.052**	1.610 ± 0.136	**1.484 ± 0.092**
LUMO – ZINDO/s (meV)	1.783 ± 0.164	**1.008 ± 0.091**	1.022 ± 0.083	**0.947 ± 0.061**
MA intensity – ZINDO (arbitrary u.)	0.062 ± 0.004	**0.050 ± 0.004**	**0.050 ± 0.003**	0.051 ± 0.002
Polarizability – DFT/PBE0 ($Å^3$)	0.055 ± 0.013	**0.031 ± 0.002**	0.041 ± 0.007	**0.034 ± 0.006**
Polarizability – SCS ($Å^3$)	0.040 ± 0.010	**0.022 ± 0.002**	0.038 ± 0.007	**0.029 ± 0.003**
QM8	**Sum**	**ABP**	**Sum**	**ABP**
E1-CAM (meV)	2.528 ± 0.105	**2.066 ± 0.054**	2.017 ± 0.137	**1.873 ± 0.042**
E1-CC2 (meV)	2.961 ± 0.089	**2.486 ± 0.058**	2.356 ± 0.087	**2.268 ± 0.045**
E1-PBE0/def2SVP (meV)	2.744 ± 0.098	**2.282 ± 0.033**	2.213 ± 0.092	**2.111 ± 0.068**
E1-PBE0/def2TZVP (meV)	2.811 ± 0.122	**2.328 ± 0.048**	2.198 ± 0.060	**2.121 ± 0.041**
E2-CAM (meV)	3.675 ± 0.066	**2.982 ± 0.066**	2.976 ± 0.129	**2.906 ± 0.063**
E2-CC2 (meV)	4.775 ± 0.339	**3.924 ± 0.077**	3.866 ± 0.153	**3.859 ± 0.107**
E2-PBE0/def2SVP (meV)	3.992 ± 0.176	**3.192 ± 0.038**	3.275 ± 0.110	**3.137 ± 0.077**
E2-PBE0/def2TZVP (meV)	3.873 ± 0.121	**3.190 ± 0.076**	3.173 ± 0.067	**3.134 ± 0.063**
f1-CAM (meV)	8.887 ± 0.747	**7.548 ± 0.435**	6.782 ± 0.378	**6.721 ± 0.486**
f1-CC2 (meV)	10.116 ± 0.237	**8.514 ± 0.612**	7.787 ± 0.258	**7.773 ± 0.397**
f1-PBE0/def2SVP (meV)	8.500 ± 0.585	**7.227 ± 0.390**	6.612 ± 0.560	**6.473 ± 0.453**
f1-PBE0/def2TZVP (meV)	8.375 ± 0.542	**7.309 ± 0.594**	6.919 ± 0.284	**6.547 ± 0.461**
f2-CAM (meV)	21.171 ± 0.813	**17.784 ± 0.610**	**16.108 ± 0.799**	16.289 ± 0.546
f2-CC2 (meV)	25.029 ± 0.825	**22.120 ± 0.601**	20.698 ± 1.039	**20.424 ± 1.136**
f2-PBE0/def2SVP (meV)	19.163 ± 1.141	**16.329 ± 1.101**	14.930 ± 0.762	**14.923 ± 0.693**
f2-PBE0/def2TZVP (meV)	19.679 ± 0.490	**17.049 ± 1.080**	**15.455 ± 0.709**	15.529 ± 0.781
QM9	**Sum**	**ABP**	**Sum**	**ABP**
HOMO – DFT (meV)	35.985 ± 1.071	**29.577 ± 0.517**	23.711 ± 0.303	**21.577 ± 0.459**
LUMO – DFT (meV)	33.505 ± 0.885	**26.960 ± 0.873**	20.832 ± 0.667	**20.488 ± 0.522**
QMugs	**Sum**	**ABP**	**Sum**	**ABP**
HOMO – DFT (meV)	65.094 ± 1.243	**52.172 ± 0.475**	24.536 ± 0.260	**23.610 ± 1.212**
LUMO – DFT (meV)	62.022 ± 0.830	**47.371 ± 0.916**	21.492 ± 0.558	**20.953 ± 0.376**
Total Energy – DFT (E_h)	9.051 ± 2.848	**3.714 ± 1.358**	3.402 ± 1.810	**2.744 ± 1.694**
MD17 (energies)	**Sum**	**ABP**	**Sum**	**ABP**
Aspirin – CCSD (kcal/mol)	3.935 ± 0.128	**3.414 ± 0.058**	2.537 ± 0.061	**2.341 ± 0.076**
Benzene – CCSD(T) (kcal/mol)	0.486 ± 0.295	**0.357 ± 0.080**	0.290 ± 0.034	**0.276 ± 0.070**
Ethanol – CCSD(T) (kcal/mol)	**0.676 ± 0.035**	0.748 ± 0.024	0.617 ± 0.063	**0.580 ± 0.034**
MDA – CCSD(T) (kcal/mol)	**0.937 ± 0.082**	0.989 ± 0.050	**1.058 ± 0.026**	1.061 ± 0.051
Toluene – CCSD(T) (kcal/mol)	1.529 ± 0.494	**1.114 ± 0.252**	1.159 ± 0.055	**1.132 ± 0.097**

Table 2: Test MAE and RMSE (mean ± standard deviation from 5 data random splits) for SchNet-based HOMO energy prediction on the OE62 dataset, including the number of learnable parameters for the attention-based pooling (ABP). The ABP configuration is reported as 'ABP(*embedding size*, *number of attention heads*, *number of SABs*)'. The number of learnable parameters for the underlying SchNet model (not including the readout) is 480,002. The smallest MAE/RMSE values are highlighted in bold. The unit used for energy is eV, as used by Chen et al.

Readout	**MAE**	**RMSE**	**# ABP parameters**
Sum	0.2656 ± 0.0177	0.4032 ± 0.0168	N/A
Average	0.1437 ± 0.0016	0.2043 ± 0.0009	N/A
OWA	0.1135 ± 0.0019	0.1670 ± 0.0021	N/A
ABP(64, 4, 2)	0.1158 ± 0.0020	0.1697 ± 0.0021	995,456
ABP(64, 8, 2)	0.1130 ± 0.0019	0.1660 ± 0.0028	3,694,720
ABP(64, 16, 2)	**0.1119 ± 0.0022**	0.1655 ± 0.0045	14,205,056
ABP(64, 16, 3)	0.1124 ± 0.0006	**0.1648 ± 0.0015**	18,403,456

Table 3: Test MAE and RMSE (mean ± standard deviation from 5 data random splits) for SchNet-based LUMO energy prediction on the OE62 dataset. The smallest MAE/RMSE values are highlighted in bold. The unit used for energy is eV, as used by Chen et al. The naming conventions and numbers of parameters are reported in Table 2.

Readout	**MAE**	**RMSE**
Sum	0.1654 ± 0.0083	0.2374 ± 0.0097
Average	0.1393 ± 0.0059	0.2037 ± 0.0097
OWA	0.1281 ± 0.0030	0.1858 ± 0.0056
ABP(64, 8, 2)	0.1050 ± 0.0024	0.1630 ± 0.0047
ABP(64, 16, 2)	**0.1010 ± 0.0011**	**0.1580 ± 0.0033**

Table 4: Test MAE for the QM7b dataset (HOMO energies) including WA pooling, which does not use pre-computed orbital coefficients (best MAEs in bold). The unit used for the energies is meV.

QM7b	**SchNet**		
Task (level of theory)	**Sum**	**WA**	**ABP**
HOMO (GW)	3.311 ± 0.190	2.951 ± 0.112	**2.342 ± 0.037**
HOMO (PBE0)	3.039 ± 0.131	2.748 ± 0.174	**2.022 ± 0.055**
HOMO (ZINDO/s)	2.639 ± 0.191	2.353 ± 0.163	**1.678 ± 0.046**

et al.) and MAE (used throughout the rest of the paper). Here, we also study LUMO energy prediction which is not considered by Chen et al., but is available in the OE62 dataset. We find that OWA offers a smaller improvement with respect to average pooling on LUMO energy prediction (8.04%) compared to HOMO energy (21.02%), as given by the MAE, with similar trends for the RMSE (Table 3). Furthermore, whereas for HOMO the attention-based pooling offered a small but noticeable improvement compared to OWA, for LUMO we observe a more significant improvement of 21.16% for ABP.

When not using orbital coefficients for a dataset such as QM7b (Table 4) where they are not readily available, we find that weighted average pooling still outperforms sum pooling by a noticeable amount (around 10%); however, ABP improves even further, with decreases in MAE between 41% and 57%.

5 Discussion

The presented results suggest that attention-based pooling is preferable to sum-based pooling for intensive, localised properties such as HOMO and LUMO energy prediction. Perhaps less expected is the uplift in performance on other properties that are not as localised as the HOMO and LUMO energies, for example total energy prediction on QMugs or energy prediction on the small MD17 datasets. In this latter case, it is also remarkable that a data-driven method like ABP is able to often outperform standard pooling when training with around 1,000 data points. Apart from the physical motivation behind improving localised property prediction, it should be noted that the attention mechanism adds an additional layer of expressivity to the network, enabling better approximation of general-purpose properties. Moreover, although we have presented the 'main' network such as SchNet or DimeNet++ and the pooling function as separate components, they work synergistically, especially for highly-expressive learnable and differentiable pooling functions. It is not unlikely that the patterns learnt at the pooling-level can propagate to the main model and lead to an improved holistic model behaviour.

When compared to OWA on the diverse OE62 dataset for HOMO energy prediction (Table 2), attention-based pooling can match and even outperform it depending on the attention configuration (i.e. number of attention heads and hidden dimensions). Under the same conditions, we observe a more than 20% improvement for ABP on LUMO energy prediction (Table 3). This is an interesting conclusion as OWA requires the pre-computation of orbital coefficients and their explicit incorporation in the loss function. Since this additional information is not required by our method, it suggests that most of the information that is required to reach this level of performance is already available in the network, but it is not fully exploited. Interestingly, Chen et al. noticed that WA can occasionally outperform OWA with the actual orbital coefficients, most likely due to the increased network flexibility. Thus, models that deviate from or even omit physical references can sometimes be preferable.

Although OWA is an innovative and physically-based approach, its scalability and applicability might limit its full potential. These methods requires the pre-computation of orbital coefficients, which are not generally available for most published datasets. Furthermore, the OWA weights are learnt by a second atomistic model which is inherently not scalable as it imposes a doubling of the model's requirements and an additional model must be added for each new property to be predicted in a multi-task scenario. Perhaps most limiting, the OWA approach is engineered specifically for orbital properties, with no straightforward analogue for properties that do not have a well-defined orbital basis.

We illustrate this last point by considering HOMO energy prediction tasks on the QM7b dataset without pre-computing the orbital coefficients (Table 4). The OWA method thus takes the more general, non-orbital specific WA form of Equation (12). The results indicate that attention-based pooling outperforms WA by about the same margin as WA pooling outperforms sum pooling. We also take the opportunity to highlight the similarity between the (O)WA methods and the well-known Deep Sets framework that considers sum-decomposable functions, where individual items are processed by simple neural networks such as MLPs before being summed [40]. Although Deep Sets offers a theoretically sound construction, our work has previously suggested and exemplified that Deep Sets-style pooling does not match the performance of attention-based pooling [21].

Overall, the replacement of simple pooling functions with an attention-based pooling function (here, the Set Transformer) has empirically proven to be the optimal choice in the majority of evaluated settings. Attention is particularly well-suited for tasks involving non-linear or localised patterns, although it is often effective for properties of different natures. In theory, an expressive and permutation-invariant module such as the Set Transformer can also learn to represent functions like sum, average, or maximum if necessary, although the amount and quality of data also becomes a consideration in such a scenario. Practically, the proposed pooling function acts as a drop-in replacement for existing pooling operations and does not require any pre-computations or modifications to the underlying network. Although the standard attention mechanism that we used here has quadratic time and memory scaling, we did not observe significantly larger training times or prohibitive increases in consumed memory for any of the shown experiments. We also did not notice overfitting or divergence due to the large number of parameters. Although a more efficient attention implementation is beyond the scope of this work, such alternatives already exist, including for the Set Transformer in the form of induced set attention blocks [36].

6 Code availability

The source code that enables all experiments to be reproduced is hosted on GitHub: https://github.com/davidbuterez/attention-based-pooling-for-quantum-properties.

7 Data availability

All the datasets used throughout the paper are publicly available through different hosting services, as indicated in the main text. For ease of use, we provide pre-processed versions of certain datasets which are accessible by following the instructions included in the source code.

References

[1] Juexin Wang, Anjun Ma, Yuzhou Chang, Jianting Gong, Yuexu Jiang, Ren Qi, Cankun Wang, Hongjun Fu, Qin Ma and Dong Xu. 'scGNN is a novel graph neural network framework for single-cell RNA-Seq analyses'. In: *Nature Communications* 12.1 (Mar. 2021), p. 1882. ISSN: 2041-1723. DOI: 10.1038/s41467-021-22197-x. URL: https://doi.org/10.1038/s41467-021-22197-x.

[2] David Buterez, Ioana Bica, Ifrah Tariq, Helena Andrés-Terré and Pietro Liò. 'CellVGAE: an unsupervised scRNA-seq analysis workflow with graph attention networks'. In: *Bioinformatics* 38.5 (Dec. 2021), pp. 1277–1286. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btab804. eprint: https://academic.oup.com/bioinformatics/article-pdf/38/5/1277/49009403/btab804.pdf. URL: https://doi.org/10.1093/bioinformatics/btab804.

[3] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N. Kinch, R. Dustin Schaeffer, Claudia Millán, Hahnbeom Park, Carson Adams, Caleb R. Glassman, Andy DeGiovanni, Jose H. Pereira, Andria V. Rodrigues, Alberdina A. van Dijk, Ana C. Ebrecht, Diederik J. Opperman, Theo Sagmeister, Christoph Buhlheller, Tea Pavkov-Keller, Manoj K. Rathinaswamy, Udit Dalwadi, Calvin K. Yip, John E. Burke, K. Christopher Garcia, Nick V. Grishin, Paul D. Adams, Randy J. Read and David Baker. 'Accurate prediction of protein structures and interactions using a three-track neural network'. In: *Science* 373.6557 (2021), pp. 871–876. DOI: 10.1126/science.abj8754. eprint: https://www.science.org/doi/pdf/10.1126/science.abj8754. URL: https://www.science.org/doi/abs/10.1126/science.abj8754.

[4] J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King and D. Baker. 'Robust deep learning–based protein sequence design using ProteinMPNN'. In: *Science* 378.6615 (2022), pp. 49–56. DOI: 10.1126/science.add2187. eprint: https://www.science.org/doi/pdf/10.1126/science.add2187. URL: https://www.science.org/doi/abs/10.1126/science.add2187.

[5] Jonathan M. Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M. Donghia, Craig R. MacNair, Shawn French, Lindsey A. Carfrae, Zohar Bloom-Ackermann, Victoria M. Tran, Anush Chiappino-Pepe, Ahmed H. Badran, Ian W. Andrews, Emma J. Chory, George M. Church, Eric D. Brown, Tommi S. Jaakkola, Regina Barzilay and James J. Collins. 'A Deep Learning Approach to Antibiotic Discovery'. In: *Cell* 180.4 (2020), 688–702.e13. ISSN: 0092-8674. DOI: https://doi.org/10.1016/j.cell.2020.01.021. URL: https://www.sciencedirect.com/science/article/pii/S0092867420301021.

[6] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec and Peter W. Battaglia. 'Learning to Simulate Complex Physics with Graph Networks'. In: *Proceedings of the 37th International Conference on Machine Learning*. ICML'20. JMLR.org, 2020.

[7] Adeesh Kolluru, Muhammed Shuaibi, Aini Palizhati, Nima Shoghi, Abhishek Das, Brandon Wood, C. Lawrence Zitnick, John R. Kitchin and Zachary W. Ulissi. 'Open Challenges in Developing Generalizable Large-Scale Machine-Learning Models for Catalyst Discovery'. In: *ACS Catalysis* 12.14 (2022), pp. 8572–8581. DOI: 10.1021/acscatal.2c02291. eprint: https://doi.org/10.1021/acscatal.2c02291. URL: https://doi.org/10.1021/acscatal.2c02291.

[8] Justin S. Smith, Benjamin T. Nebgen, Roman Zubatyuk, Nicholas Lubbers, Christian Devereux, Kipton Barros, Sergei Tretiak, Olexandr Isayev and Adrian E. Roitberg. 'Approaching coupled cluster accuracy with a general-purpose neural network potential through transfer learning'. In: *Nature Communications* 10.1 (July 2019), p. 2903. ISSN: 2041-1723. DOI: 10.1038/s41467-019-10827-4. URL: https://doi.org/10.1038/s41467-019-10827-4.

[9] Peikun Zheng, Roman Zubatyuk, Wei Wu, Olexandr Isayev and Pavlo O. Dral. 'Artificial intelligence-enhanced quantum chemical method with broad applicability'. In: *Nature Communications* 12.1 (Dec. 2021), p. 7022. ISSN: 2041-1723. DOI: 10.1038/s41467-021-27340-2. URL: https://doi.org/10.1038/s41467-021-27340-2.

[10] Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp and O. Anatole von Lilienfeld. 'Quantum chemistry structures and properties of 134 kilo molecules'. In: *Scientific Data* 1.1 (Aug. 2014), p. 140022. ISSN: 2052-4463. DOI: 10.1038/sdata.2014.22. URL: https://doi.org/10.1038/sdata.2014.22.

[11] Clemens Isert, Kenneth Atz, José Jiménez-Luna and Gisbert Schneider. 'QMugs, quantum mechanical properties of drug-like molecules'. In: *Scientific Data* 9.1 (June 2022), p. 273. ISSN: 2052-4463. DOI: 10.1038/s41597-022-01390-7. URL: https://doi.org/10.1038/s41597-022-01390-7.

[12] Kuzma Khrabrov, Ilya Shenbin, Alexander Ryabov, Artem Tsypin, Alexander Telepov, Anton Alekseev, Alexander Grishin, Pavel Strashnov, Petr Zhilyaev, Sergey Nikolenko and Artur Kadurin. 'nablaDFT: Large-Scale Conformational Energy and Hamiltonian Prediction benchmark and dataset'. In: *Phys. Chem. Chem. Phys.* 24 (42 2022), pp. 25853–25863. DOI: 10.1039/D2CP03966D. URL: http://dx.doi.org/10.1039/D2CP03966D.

[13] Johannes Hoja, Leonardo Medrano Sandonas, Brian G. Ernst, Alvaro Vazquez-Mayagoitia, Robert A. DiStasio and Alexandre Tkatchenko. 'QM7-X, a comprehensive dataset of quantum-mechanical properties spanning the chemical space of small organic molecules'. In: *Scientific Data* 8.1 (Feb. 2021). DOI: 10.1038/s41597-021-00812-2. URL: https://doi.org/10.1038/s41597-021-00812-2.

[14] David Buterez, Jon Paul Janet, Steven Kiddle and Pietro Liò. 'Multi-fidelity machine learning models for improved high-throughput screening predictions'. In: *ChemRxiv* (2022). DOI: 10.26434/chemrxiv-2022-dsbm5-v2.

[15] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko and K.-R. Müller. 'SchNet – A deep learning architecture for molecules and materials'. In: *The Journal of Chemical Physics* 148.24 (2018), p. 241722. DOI: 10.1063/1.5019779. eprint: https://doi.org/10.1063/1.5019779. URL: https://doi.org/10.1063/1.5019779.

[16] Johannes Gasteiger, Janek Groß and Stephan Günnemann. 'Directional Message Passing for Molecular Graphs'. In: *International Conference on Learning Representations (ICLR)*. 2020.

[17] Johannes Gasteiger, Shankari Giri, Johannes T. Margraf and Stephan Günnemann. 'Fast and Uncertainty-Aware Directional Message Passing for Non-Equilibrium Molecules'. In: *Machine Learning for Molecules Workshop, NeurIPS*. 2020.

[18] Victor Garcia Satorras, Emiel Hoogeboom and Max Welling. 'E(n) Equivariant Graph Neural Networks'. In: *CoRR* abs/2102.09844 (2021). arXiv: 2102.09844. URL: https://arxiv.org/abs/2102.09844.

[19] Artur M. Schweidtmann, Jan G. Rittig, Jana M. Weber, Martin Grohe, Manuel Dahmen, Kai Leonhard and Alexander Mitsos. 'Physical pooling functions in graph neural networks for molecular property prediction'. In: *Computers & Chemical Engineering* 172 (2023), p. 108202. ISSN: 0098-1354. DOI: https://doi.org/10.1016/j.compchemeng.2023.108202. URL: https://www.sciencedirect.com/science/article/pii/S0098135423000716.

[20] Ke Chen, Christian Kunkel, Bingqing Cheng, Karsten Reuter and Johannes T. Margraf. 'Physics-Inspired Machine Learning of Localized Intensive Properties'. In: *ChemRxiv* (2023). DOI: 10.26434/chemrxiv-2023-h9qdj.

[21] David Buterez, Jon Paul Janet, Steven J Kiddle, Dino Oglic and Pietro Liò. 'Graph Neural Networks with Adaptive Readouts'. In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh, Alekh Agarwal, Danielle Belgrave and Kyunghyun Cho. 2022. URL: https://openreview.net/forum?id=yts7fLpWY9G.

[22] Thomas N. Kipf and Max Welling. 'Semi-Supervised Classification with Graph Convolutional Networks'. In: *International Conference on Learning Representations (ICLR)*. 2017.

[23] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò and Yoshua Bengio. 'Graph Attention Networks'. In: *International Conference on Learning Representations*. 2018.

[24] Shaked Brody, Uri Alon and Eran Yahav. 'How Attentive are Graph Attention Networks?' In: *International Conference on Learning Representations*. 2022.

[25] Keyulu Xu, Weihua Hu, Jure Leskovec and Stefanie Jegelka. 'How Powerful are Graph Neural Networks?' In: *International Conference on Learning Representations*. 2019.

[26] Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò and Petar Veličković. 'Principal Neighbourhood Aggregation for Graph Nets'. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 13260–13271.

[27] David Weininger. 'SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules'. In: *Journal of Chemical Information and Computer Sciences* 28.1 (1988), pp. 31–36. DOI: 10.1021/ci00057a005. eprint: https://doi.org/10.1021/ci00057a005.

[28] L. C. Blum and J.-L. Reymond. '970 Million Druglike Small Molecules for Virtual Screening in the Chemical Universe Database GDB-13'. In: *J. Am. Chem. Soc.* 131 (2009), p. 8732.

[29] M. Rupp, A. Tkatchenko, K.-R. Müller and O. A. von Lilienfeld. 'Fast and accurate modeling of molecular atomization energies with machine learning'. In: *Physical Review Letters* 108 (2012), p. 058301.

[30] Grégoire Montavon, Matthias Rupp, Vivekanand Gobre, Alvaro Vazquez-Mayagoitia, Katja Hansen, Alexandre Tkatchenko, Klaus-Robert Müller and O Anatole von Lilienfeld. 'Machine learning of molecular electronic properties in chemical compound space'. In: *New Journal of Physics* 15.9 (2013), p. 095003. URL: http://stacks.iop.org/1367-2630/15/i=9/a=095003.

[31] Lars Ruddigkeit, Ruud van Deursen, Lorenz C. Blum and Jean-Louis Reymond. 'Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17'. In: *Journal of Chemical Information and Modeling* 52.11 (2012). PMID: 23088335, pp. 2864–2875. DOI: 10.1021/ci300415d. eprint: https://doi.org/10.1021/ci300415d. URL: https://doi.org/10.1021/ci300415d.

[32] Raghunathan Ramakrishnan, Mia Hartmann, Enrico Tapavicza and O. Anatole von Lilienfeld. 'Electronic spectra from TDDFT and machine learning in chemical space'. In: *The Journal of Chemical Physics* 143.8 (2015), p. 084111. DOI: 10.1063/1.4928757. eprint: https://doi.org/10.1063/1.4928757. URL: https://doi.org/10.1063/1.4928757.

[33] Stefan Chmiela, Huziel E. Sauceda, Klaus-Robert Müller and Alexandre Tkatchenko. 'Towards exact molecular dynamics simulations with machine-learned force fields'. In: *Nature Communications* 9.1 (Sept. 2018), p. 3887. ISSN: 2041-1723. DOI: 10.1038/s41467-018-06169-2. URL: https://doi.org/10.1038/s41467-018-06169-2.

[34] Annika Stuke, Christian Kunkel, Dorothea Golze, Milica Todorović, Johannes T. Margraf, Karsten Reuter, Patrick Rinke and Harald Oberhofer. 'Atomic structures and orbital energies of 61,489 crystal-forming organic molecules'. In: *Scientific Data* 7.1 (Feb. 2020), p. 58. ISSN: 2052-4463. DOI: 10.1038/s41597-020-0385-y. URL: https://doi.org/10.1038/s41597-020-0385-y.

[35] Michael M. Bronstein, Joan Bruna, Taco Cohen and Petar Velickovic. 'Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges'. In: *CoRR* abs/2104.13478 (2021). arXiv: 2104.13478. URL: https://arxiv.org/abs/2104.13478.

[36] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi and Yee Whye Teh. 'Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks'. In: *Proceedings of the 36th International Conference on Machine Learning*. 2019, pp. 3744–3753.

[37] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai and Soumith Chintala. 'PyTorch: An Imperative Style, High-Performance Deep Learning Library'. In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[38] Matthias Fey and Jan E. Lenssen. 'Fast Graph Representation Learning with PyTorch Geometric'. In: *ICLR Workshop on Representation Learning on Graphs and Manifolds*. 2019.

[39] K. T. Schütt, P. Kessel, M. Gastegger, K. A. Nicoli, A. Tkatchenko and K.-R. Müller. 'SchNetPack: A Deep Learning Toolbox For Atomistic Systems'. In: *Journal of Chemical Theory and Computation* 15.1 (2019), pp. 448–455. DOI: 10.1021/acs.jctc.8b00908. eprint: https://doi.org/10.1021/acs.jctc.8b00908. URL: https://doi.org/10.1021/acs.jctc.8b00908.

[40] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov and Alexander J. Smola. 'Deep Sets'. In: *CoRR* abs/1703.06114 (2017). arXiv: 1703.06114. URL: http://arxiv.org/abs/1703.06114.