Around Frame Averaging Equivariant Graph Neural Networks for Materials Modeling

Geometric Data Analysis

Ali Ramlaoui
Théo Saulus
Basile Terver
Master MVA
France

ABSTRACT

This report explores the application of Graph Neural Networks (GNNs) for modeling geometric structures, applied to catalyst discovery. We investigate FAENet, an architecture that incorporates a data augmentation technique called Frame-Averaging to endow the model with an $E(3)$-equivariance property while maintaining very good efficiency and competitive precision. We also propose enhancements to improve FAENet's precision and depth, including the use of Learnt Canonicalization Function, Noisy Nodes regularization, denoising pre-training, and Ewald-based long-range message passing. Experimental evaluations on the datasets OC20 and OC20-Dense (which is not present on the original paper) validate the effectiveness of some of these improvements, when computationally tractable for us.

1 INTRODUCTION

For many Machine Learning tasks, it is important to train models that preserve the symmetry of the problem to be solved [2]. The most common example is the image classification task where the output should not change when the image is rotated. A model that is insensitive to such a transformation is called invariant. For image segmentation the result should also be rotated with the image using a similar transformation leading to an equivariant model. GNNs have proven to be successful on multiple tasks that rely on modelling geometric structures that can be discretized. Chemistry and materials science are a good example where it is possible to predict the property of molecules by understanding the molecular topology and geometry [25]. This is done using for example three-dimensional point clouds structures that can be linked to form a graph structure depending on the distance between the atomic elements considered or known bonds.

Machine Learning for catalysts discovery. The release of high quality public datasets for Machine Learning (ML) applied to materials science in recent years has raised the interest of the geometric graphs community and allowed for the publications of many interesting architectures designed for the tasks of the datasets. Most of these datasets, such as *OC20* [4] which paved the way for the development of larger scale GNNs or *QM7-X* [11], focus on catalyst discovery which can help solve many societal challenges in the future. More specifically, finding relevant electrocatalysts can have major impacts for solving one of the biggest challenges for renewable energy storage which is accelerating *electrolysis* [27]. These datasets comprise millions ($1,281,040$ for OC20) of Density Functional Theory (DFT) relaxations which consist in single adsorbates, which are small molecules, physically binding at the surfaces of catalysts. Many tasks can then be considered from such elements of the dataset, mainly:

- *S2EF* (Structure to Energy and Forces). The positions of the atoms are given to a model which has to predict the energy and the forces for each atom as they will be computed by DFT.
- *IS2RS* (Initial State to Relaxed Structure). The inputs are the initial structures and the goal is to predict the atomic positions in their final relaxed state. DFT simulates this process by computing the forces on every atom from the interactions and then updating the positions until convergence. This can also be simulated using multiple steps of S2EF after training a model for the previous task.
- *IS2RE* (Initial State to Relaxed Energy). The goal of this task is to predict the relaxation energy after the binding of the adsorbate on the catalyst for the given initial configuration.

The ground truth of the tasks of interest is therefore the DFT. While DFT does not allow to find the global minimal binding energy because it depends on the initial configuration, it is a very accurate simulation of the process. The reason why ML models are interesting to approximate the simulations is that the complexity of DFT is $O(N^3)$ in the number of atomic elements. This complexity does not allow to sample from the extremely large space of catalysts and adsorbates couples (organic and inorganic) [27].

2 SYMMETRY-PRESERVING DATA AUGMENTATION

Notations. Formally, for any finite-dimensional vector space $\mathcal{X}$ and group G, a group action of G on $\mathcal{X}$ is a mapping

$$G \times \mathcal{X} \to \mathcal{X}$$
$$(g, x) \mapsto g \cdot x$$

such that for $g, h \in G$, $x \in \mathcal{X}$, we have $(gh) \cdot x \mapsto g \cdot (h \cdot x)$. In the case of materials design, we will usually have $\mathcal{X} = \mathbb{R}^{n \times 3}$, where n is the number of nodes (typically atoms) and $G = E(3)$, the group of orthogonal rotations, translations and reflections on the space.

Invariance and Equivariance. A real representation of the group G is a function $\rho : G \to \mathbb{R}^{n \times n}$ such that for $g, h \in G$, we have $\rho(gh) = \rho(g)\rho(h)$. The representation allows to translate the action of the group to an operation that can be expressed with elements

from different sets such as matrix multiplications. A function $f : \mathcal{X} \to \mathcal{Y}$ is said to be:

- G-invariant if for $g \in G$, $x \in \mathcal{X}$, $f(\rho(g)x) = f(x)$.
- G-equivariant if we have $f(\rho_1(g)x) = \rho_2(g)f(x)$, where ρ_1 and ρ_2 are respectively representations of the group G on the vector spaces $\mathcal{X}$ and $\mathcal{Y}$.

2.1 Frame-Averaging

Equivariance in GNNs. For GNNs, equivariance can be enforced in the architecture of the model during the message-passing steps by using equivariant features of the input's representation. In materials modeling, most models follow this paradigm with different families of models, often at the cost of expensive features computations [7]. A more comprehensive comparison of architectures is available in section 3.

Frame averaging. Another approach consists in using data augmentation and, for example, rotating the training samples randomly with the hope that the model learns to preserve symmetry without any theoretical guarantees [12]. However, the improvements in accuracy and speed from such models were not enough to constitute a true Pareto optimal improvement. One of the main contributions of the FAENet architecture is its ability to learn equivariance without having any symmetry preserving constraints on the GNN, by leveraging a data augmentation strategy that preserves equivariance guarantees. Duval et al. [7] make use of the Frame Averaging framework, introduced by Puny et al. [19], in which $\mathcal{X}$ and $\mathcal{Y}$ denote normed linear spaces with respective representations ρ_1 and ρ_2 of a group G. In our case, the group of interest is $E(3)$. This choice is justified since the aim is to predict properties of adsorbates that bind on a surface with a repeated structure A *frame* is defined as a function $\mathcal{F} : \mathcal{X} \to 2^G$ taking values in a non-empty subset of the group G such that:

(1) $\mathcal{F}$ is G-equivariant if for all $x \in \mathcal{X}$, and $g \in G$, we have $\mathcal{F}(\rho_1(g)x) = g\mathcal{F}(x) = \{gh, h \in \mathcal{F}(x)\}$.
(2) $\mathcal{F}$ is bounded over a subset K if there exists $C > 0$ such that $||\rho_2(g)|| \leq C$, the operator norm over $\mathcal{Y}$, for all g in all frames composed from K.

Under such conditions, and for every $\Phi : \mathcal{X} \to \mathcal{Y}$, the function $\langle\Phi\rangle_{\mathcal{F}} : \mathcal{X} \to \mathcal{Y}$, called *average over the frame* $\mathcal{F}$, and defined as

$$\langle\Phi\rangle_{\mathcal{F}} : x \mapsto \frac{1}{|\mathcal{F}(x)|} \sum_{g \in \mathcal{F}(x)} \rho_2(g)\Phi(\rho_1(g)^{-1}x), \tag{1}$$

is G-equivariant. This allows to create an arbitrary neural network, and guarantee the symmetry by averaging the outputs over a well chosen frame.

Choosing the frame. In the case of FAENet, the neural network is defined as

$$\Phi : \mathbb{R}^{n\times 3} \times \mathbb{N}^n \to \mathcal{Y}$$
$$(X, Z) \mapsto y$$

where X is the position of the atoms, Z is their atomic numbers, $\mathcal{Y} = \mathbb{R}^{n\times 3}$ in the case of force predictions, and $\mathcal{Y} = \mathbb{R}$ in the case of energy prediction. The group $E(3)$ only acts on X, and not Z. The authors then define the frame for every molecule represented by positions X. A Principal Component Analysis (PCA) on the atomic structure allows to decompose the covariance matrix of the points cloud $\Sigma = U^T \Lambda U$ derived from the centroid of the positions $\mathbf{t} = \frac{1}{n} X^T \mathbf{1}$. U is an orthogonal matrix and Λ is the diagonal matrix containing the three eigenvalues $\lambda_1 > \lambda_2 > \lambda_3$ (assumed distinct because we consider non planar structures). The columns of U form an orthogonal basis that identifies the three principal components of spatial variance of the molecule. The frame can then be taken as

$$\mathcal{F}(X) = \{(U, t) \mid U = [\pm u_1, \pm u_2, \pm u_3]\} \tag{2}$$

which is a subset of $E(3)$. Every element of the frame (U, t) acts on (X, Z) using representations as follows:

- $\rho_1(g)(X, Z) = (XU^T + 1\mathbf{t}^T, Z)$, which provides a rotation of the elements, and then translation to the centroid on $\mathbb{R}^{n\times 3}$
- $\rho_2(g)X = XU^T$ for equivariant predictions (forces are rotated)
- $\rho_2(g)X = X$ for invariant predictions (energy does not change).

The authors prove that the frame $\mathcal{F}$ defined as such is G-equivariant and bounded.

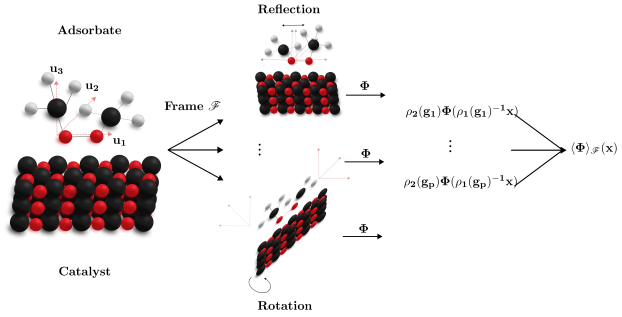

Figure 1: Frame-averaging on an adsorbate-catalyst system. PCA gives the three main components of the system to create the frame which then allows to make different predictions and average them out to get an equivariant Neural Network.

2.2 Stochastic Frame-Averaging

As defined in section 2.1, the frame requires to average the predictions of the neural network over $|\mathcal{F}(X)| = 2^3 = 8$ elements of the frame. In order to make the computations faster, the authors decide to randomly sample one (or a few) element from the frame to apply to the model during training. Given enough epochs and training, the model learns the symmetries through an empirical process thanks to the canonicalization during pre-processing. Although Stochastic Frame-Averaging (SFA) does not have theoretical guarantees, it is based upon a symmetry preserving framework and it has been experimentally shown to learn almost perfect invariance or equivariance, while accelerating the procedure with an expected $\times 8$ speedup [7].

Expressive power of the approach. The *expressive power* of regular GNNs is their ability to distinguish between two *non-isomorphic* graphs. Two graphs are said to be non-isormorphic if there does not exist a bijection of the nodes leading to the same graph. This

question has been studied in depth using using the Weisfeiler-Leman test, which cannot be applied to Geometric GNNs because the positional vectors of the nodes lead to physical symmetries that are not covered by the test. Work has been done to propose a geometric version of the Weisfeiler-Leman test (GWL): Joshi and Mathis [13] propose two main experiments to measure the expressive power of models in order to check whether they are able to differentiate between geometric graph structures that might lead to different chemical properties:

- **k-chain geometric graphs.** These are chains of $k-2$ nodes linked in 1D, with two nodes on each end that bend differently to form a 2D plane. Physically, the molecules are different and the models should be able to distinguish between these two graphs after at least $\lfloor \frac{k}{2} \rfloor$ iterations because of the local nature of message passing.
- **Rotationally symmetric structures.** Similarly to the previous test, two structures (carbon cycles for example) are appended to each end of an L-fold 1D structure of length L. The goal is to check whether the models can distinguish between variants with different rotations of the structure.

The authors of FAENet experimentally show that the model passes these tests, even with just one layer, for regular Frame-Averaging and for Stochastic Frame-Averaging thanks to the data-enforced symmetry through the canonical representation with PCA.

2.3 Improvement: Learning Canonicalization Functions

As an alternative to the previous methods, a more flexible, yet provably geometry-preserving, method has been proposed by Kaba et al. [14]. Instead of using the (S)FA heuristic, they learn the canonicalization function ζ, and prove that a shallow function is enough for this matter, allowing the prediction function Φ to be as expressive as in FAENet. They prove the following results:

Partial canonicalization. One can impose part of the geometric constraint on the prediction network Φ, and use the canonical network ζ for additional ones. Formally, if for $g, x \in G \times \mathcal{X}$, $\exists k \in K \leq G$ such that $\zeta(\rho_1(g)x) = \rho_2(g)\zeta(x)\rho_2(k)$ and Φ is K-equivariant, then $\tilde{\Phi} : x \mapsto \zeta'(x)\Phi(\zeta(x)^{-1}x)$ is G-equivariant, where $\zeta(x)^{-1}$ is the inverse of the representation matrix, and $\zeta'(x) = \rho_2(\rho_1^{-1}(\zeta(x)))$ is the counterpart of $\zeta(x)$ on the output.

Universality result. A function $\tilde{\Phi}$ is said to be a universal approximator of G-equivariant functions if for all G-equivariant function Ψ, for all compact set $\mathcal{K} \subseteq \mathcal{X}$, and $\varepsilon > 0$, there exists parameters of the function $\tilde{\Phi}$ such that for all $x \in \mathcal{K}$, $\|\tilde{\Phi}(x) - \Psi(x)\| < \varepsilon$. The weaker universality result they prove is the most relevant to us: if $\tilde{\Phi}(x) = \zeta'(x)\Phi(\zeta(x)^{-1}x)$, where ζ is G-equivariant and continuous, and Φ is a multi-layer perceptron, then $\tilde{\Phi}$ is a universal approximation of G-equivariant functions. In other words, one can choose a simple ζ function, and an expressive Φ function, and still maintain geometric guarantees.

Design of Φ and ζ. In practice, the authors experiment and propose some choices of networks. In particular, they test the idea of using shallow ζ functions, and prove their superiority against other methods, including Frame-Averaging, because the learned ζ functions are now task-oriented. Specifically, for tasks that involve predicting n-body physical properties, they propose to use Vector Neurons [5], a class of $SO(3)$-equivariant models, further orthonormalized with a Gram-Schmidt process to canonicalize the representation in $O(3)$. Implementing this strategy to our network implies to partly modify the training process, in order to also train the canonicalization function.

3 GEOMETRIC GNNS AND FAENET ARCHITECTURE

As summarized by Duval et al. [6], Geometric GNNs' interaction blocks update scalar and vector features from layer l to $l+1$ via learnable aggregate and update functions AGG and UPD, respectively:

$$\mathbf{s}_i^{(l+1)}, \vec{\mathbf{v}}_i^{(l+1)} = UPD\left(\mathbf{s}_i^{(l)}, \vec{\mathbf{v}}_i^{(l)}, AGG\left(\{(\mathbf{s}_i^{(l)}, \mathbf{s}_j^{(l)}, \vec{\mathbf{v}}_i^{(l)}, \vec{\mathbf{v}}_j^{(l)}, \vec{x}_{ij}) \mid j \in \mathcal{N}_i\}\right)\right) \tag{3}$$

where $\vec{x}_{ij} = \vec{x}_i - \vec{x}_j$ denote relative position vectors, $\mathcal{N}_i$ denote the neighbor nodes of node i. We will see in this section that $E(3)$-invariant GNN layers and non-symmetry preserving methods do not update vector features and only aggregate scalar quantities from local neighbourhoods:

$$\mathbf{s}_i^{(l+1)} = UPD\left(\mathbf{s}_i^{(l)}, AGG\Big(\{(\mathbf{s}_i^{(l)}, \mathbf{s}_j^{(l)}, \vec{\mathbf{v}}_i, \vec{\mathbf{v}}_j, \vec{x}_{ij}) | j \in \mathcal{N}_i\}\Big)\right). \tag{4}$$

3.1 FAENet architecture

FAENet was designed to make use of the geometric guarantees from the FA framework. Thus, it does not impose constraints through its architecture (contrary to other geometric GNNs, as explained in subsection 3.2). This allows for a more expressive network, while keeping theoretical guarantees of FA. The architecture devises into sequential blocks described as follows:

Graph creation. The input data is first transformed with ρ_1 in order to map it to a canonical representation, thanks to the frame $\mathcal{F}(D)$. In particular, when using SFA, only one canonical element is randomly picked. The adjacency matrix A is then computed based on a given distance cutoff c, such that

$$A_{ij} = \begin{cases} 1 & \text{if } \|x_i - x_j + O_{ij*} \cdot C\| < c \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where the term $O_{ij*} \cdot C$ allows to take *periodic boundary conditions* into account in the case of crystals modeling, as discussed in more depth in [4].

Embedding block. In this block, node embeddings $(h_1, \ldots, h_n) \in \mathbb{R}^{h \times n}$ and edge embeddings e_{ij}, for all pairs of nodes (i, j), are initialized. For nodes (i.e., atoms of the molecule), the atomic number, atomic group, and some selected physical properties (such as atomic radius and polarizability, Van-der-Walls radius) are passed in a two-layers perceptron. For edges, relative positions $\vec{x_{ij}} := \vec{x_i} - \vec{x_j}$ and radial basis functions $\|\vec{x_{ij}}\|$ are concatenated and fed into a two-layers perceptron. Duval et al. [7] prove through ablation studies that both of these components are important for performance, despite the apparent redundancy.

Interaction block. Here, information is propagated between each atom i and its neighbors $j \in \mathcal{N}_i$, using a continuous convolution of their embeddings h_j from the preceding block with geometric information:

$$h_i^{(l+1)} = h_i^{(l)} + MLP\left(\sum_{j \in \mathcal{N}_i} h_j^{(l)} \odot f_{ij}^{(l)}\right) \tag{6}$$

where $f_{ij}^{(l)} = \sigma(MLP(e_{ij} \| h_i^{(l)} \| h_j^{(l)}))$, and σ is the swish activation function. After each interaction block, a GraphNorm [3] batch normalization is used to prevent gradient issues. Additionnally, jumping connections link the interaction blocks and the output layer block (in the form of a concatenation or a sum), which proved to help generalization capabilities of the network.

Output block. Finally, the node embeddings $(h_1, \dots, h_n)$ are fed into a two-layers perceptron with 4 outputs for each atom. The three first are used to build the equivariant prediction $\tilde{\Phi}^{equiv}$, and the last one for the invariant prediction $\tilde{\Phi}^{invar}$. Let's name this (almost final) representation $(y_1, \dots, y_n)$.

- For $\tilde{\Phi}^{equiv}$, a (learnt) weighted average α of each y_j is computed:

$$\tilde{\Phi}^{equiv} = \sum_{j=1}^{n} \alpha\left(h_j^{(L)}\right) \cdot y_j \tag{7}$$

- For $\tilde{\Phi}^{equiv}$, y_i is multiplied by the inverse transformation $U^\top$ as explained in subsection 2.1. In the case of SFA, this is the inverse of the randomly picked transformation:

$$\tilde{\Phi}^{equiv} = hU^\top \tag{8}$$

In order to avoid non-physical predictions, Duval et al. [7] introduce an energy-conservation loss on top of the initial loss, designed to compare predicted forces with the gradient of the energy with respect to the atoms' positions:

$$\text{EnergConservLoss} = \sum_{i=1}^{n} \|\tilde{\Phi}^{forces}(D) - \nabla\tilde{\Phi}^{energy}(D)\|_2 \tag{9}$$

3.2 Comparison with other Geometric GNNs

We now provide an overview of the various types of Geometric GNNs, taking inspiration from Duval et al. [6].

Invariant GNNs. The first idea to enforce symmetry-invariance in GNNs was to extract and aggregate $E(3)$-invariant features, by scalarising the geometric information in the local neighborhood. For instance, computing relative distances between atoms is a scalarisation of the geometric information given by the positions of the atoms $\vec{x}$, and is invariant to translations, rotations and reflections. These scalar features are updated from layer l to $l+1$ according to (4).

SchNet [21] uses relative distances $\|\vec{x}_{ij}\|$ between pairs of nodes, encoded by a learnable radial basis function (rbf) ψ to encode local geometric information. Indeed, SchNet introduces a continuous convolution to combine the encoded distance information with neighboring atom representations via element wise multiplication $\odot$, which yields the following message passing equation:

$$\mathbf{s}_i^{(l+1)} = \mathbf{s}_i^{(l)} + \sum_{j \in \mathcal{N}_i} \mathbf{s}_j^{(l)} \odot \psi(\|\vec{x}_{ij}\|). \tag{10}$$

DimeNet [9] goes further by using continuous convolutions with a filter that combines pairwise distances $\|\vec{x}_{ij}\|$ and bond angles $\measuredangle ijk := \measuredangle(\vec{x}_{ij}, \vec{x}_{ik})$ between triplets of atoms, which yields the following equation with learnable f_1 and f_2 (typically a MLP):

$$\mathbf{s}_i^{(l+1)} = \sum_{j \in \mathcal{N}_i} f_1\left(\mathbf{s}_i^{(l)}, \mathbf{s}_j^{(l)}, \|\vec{x}_{ij}\|, \sum_{k \in \mathcal{N}_i \setminus \{j\}} f_2\left(\mathbf{s}_j^{(l)}, \mathbf{s}_k^{(l)}, \|\vec{x}_{ij}\|, \measuredangle ijk\right)\right). \tag{11}$$

Invariant GNNs learn scalar representations from equivariant geometric data, such as relative distances and bond angles, which is a computationnally efficent procedure to equivariant GNNs. They can also be used to learn equivariant properties, such as atomic forces, by taking the gradient of the energy, which is an invariant property, with respect to atomic positions.

Equivariant GNNs. Equivariant GNNs do not pre-compute local invariants but perform message passing so that the features at each layer are equivariant to symmetric transformations of the input. For example, a rotation of the input molecule would entail the hidden states at each layer to be rotated in the same way, whereas they would be unchanged in an invariant GNN. To ensure the equivariance of features, one must be careful about the type of the features, that is the linear representation of the symmetry group the feature vector transforms in, how to add, multiply and perform non-linear operations on various types.

Let us first consider the two familiar types of scalars and vectors. The multiplication rules on these types are the dot product of vectors, the product of a scalar and a vector and the product of two scalars. Given these three multiplication rules, we can build various message passing rules. To focus on the Aggregation step of (3), we denote

$$\mathbf{m}_i^{(l)}, \vec{\mathbf{m}}_i^{(l)} = AGG\left(\{(\mathbf{s}_i^{(l)}, \mathbf{s}_j^{(l)}, \vec{\mathbf{v}}_i^{(l)}, \vec{\mathbf{v}}_j^{(l)}, \vec{x}_{ij}) \mid j \in \mathcal{N}_i\}\right) \tag{12}$$

Schütt et al. [22] introduce PaiNN, which aggregates scalar and vector features learning filters f_1, f_2, f_3 depending on the pairwise distances:

$$\mathbf{m}_i^{(l)} = \mathbf{s}_i^{(l)} + \sum_{j \in \mathcal{N}_i} f_1(\mathbf{s}_j^{(l)}, \|\vec{x}_{ij}\|) \tag{13}$$

$$\vec{\mathbf{m}}_i^{(l)} = \vec{\mathbf{v}}_i^{(l)} + \sum_{j \in \mathcal{N}_i} f_2(\mathbf{s}_j^{(l)}, \|\vec{x}_{ij}\|) \odot \vec{\mathbf{v}}_j^{(l)} + \sum_{j \in \mathcal{N}_i} f_3(\mathbf{s}_j^{(l)}, \|\vec{x}_{ij}\|) \odot \vec{x}_{ij} \tag{14}$$

Scalar-vector equivariant GNNs do not involve expensive operations and achieve good performance. Many of the models in the literature are scalar-vector type GNNs. They themselves are a special type of a broader class of equivariant GNNs. Indeed, there exists a theory to support higher order tensor-type GNNs. In addition to scalar and vector features, such GNNs consider tensors of second order (classical matrices) or higher order to be part of the node's features.

Another type of equivariant GNN is with spherical tensors. In these models, each atom's embedding $\mathbf{h}_i$ is representend by the spherical harmonics coefficients $Y_{mc}^{(l)}$ of degree $0 \leq l \leq L$, order $-l \leq m \leq l$ and number of channels c.

Equiformer. The most performant equivariant GNNs are of the spherical tensor type and based on transformers blocks for message

passing. Liao and Smidt [17] introduce Equiformer and then propose Equiformer v2 Liao et al. [18] the same year, which increases the maximum degree of the tensors of the equivariant representations. Equiformer v2 outperforms previous state-of-the-art methods on the large-scale OC20 dataset by up to 12% on forces, 4% on energies and offers better speed-accuracy trade-offs.

Non-symmetry preserving GNNs. As discussed in Section 3.1, FAENet does not impose any constraints on the architecture of the GNN because it leverages a form of symmetry-preserving data augmentation technique to ensure equivariance of the model. FAENet's message passing equation can be compared to other types of Geometric GNNs by rewriting equation (6) as

$$\mathbf{s}_i^{(l+1)} = f_1\left(\mathbf{s}_i^{(l)} + \sum_{j \in \mathcal{N}_i} \mathbf{s}_j^{(l)} \odot f_2\left(\vec{x}_{ij}, \mathbf{s}_i^{(l)}, \mathbf{s}_j^{(l)}\right)\right), \tag{15}$$

where the features are only scalar and the aggregation function makes use of geometric information through the relative positions $\vec{x_{ij}}$ and a learnable convolution filter f_2.

3.3 General comments on FAENet

As discussed in 3.1, FAENet's main advantage is to enforce equivariance through SFA, allowing to avoid design constraints on the GNN and therefore enabling a computationally efficient design. Moreover, SFA also allows great expressivity (as discussed in 2.2).

Yet, the performances of FAENet stagnate when increasing the number of interaction blocks beyond about five blocks. This encourages to think of improvements of the model in order to leverage the benefits of having a deeper architecture. Also, the performances of FAENet are competitive but not the best at predicting the relaxed energy on the benchmarks. A good use for the current version of FAENet would be to leverage its speed to accelerate DFT by positioning the molecule in the state predicted by FAENet and then perform DFT, as recommended by Lan* et al. [16].

4 PROPOSED IMPROVEMENTS FOR FAENET

4.1 Improvement: Noisy Nodes

Oversmoothing. FAENet is not currently scalable as it cannot leverage the benefits of a deep GNN. This is due to a very well-known phenomen in learning with GNNs called *oversmoothing*. Oversmoothing occurs when a GNN's latent node representations become increasingly indistinguishable over successive steps of message passing. Once we have reached oversmoothing, the relational structure of the latent representation is lost for further message passing layers, which therefore cannot improve the expressivity of the GNN.

Noisy Nodes. To address this issue, Godwin et al. [10] propose to use noisy regularisation, introducing their method called Noisy Nodes. As illustrated in Figure 2, the core idea is that Noisy Nodes perturbs the input node positions with a noise σ, and requires a decoding head running in parallel to the original model to reconstruct the unperturbed information from the GNN's latent representations. In addition to the noising of the nodes, we can also ask the model to denoise the perturbed edge features (for example bond-types between atoms). In practice, this requires to add an auxiliary

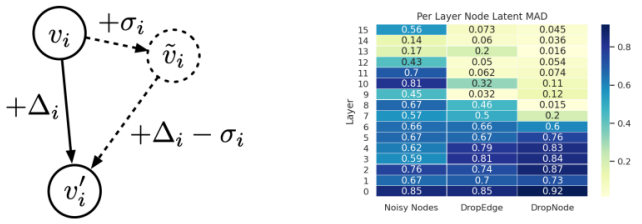

Figure 2: Left: Noisy Nodes mechanics during training. Right: per layer node latent diversity, measured by MAD on a 16 layer MPNN, we can see Noisy Nodes maintains a higher latent diversity throughout the layers than competing methods. Reprinted from Godwin et al. [10].

denoising autoencoder loss to the original training loss. Thus, the final loss of a GNN model we train with Noisy Nodes is

$$\text{Loss} = \lambda \cdot \text{NoisyNodesLoss}(\hat{G}', V') + \text{PrimaryLoss}(\hat{G}', V'), \tag{16}$$

where λ is the weight we assign to the auxiliary denoising task, $\hat{G}' = \tilde{\Phi}(\tilde{G})$ is the output of the model $\tilde{\Phi}$, $\tilde{G}$ is the noised graph, and V' can either be the target nodes features (e.g. atom positions at equilibrium, that is the IS2RS task) or the initial nodes positions (see next subsection on denoising pre-training). For example, Addanki et al. [1] propose to use as "Noisy Nodes loss" the cross-entropy on reconstructing atom and bond types or the MAE on regressing denoised displacements. The performances of Noisy Nodes on very deep GNNs is decisive, and allowed Addanki et al. [1] to leverage the power of a 50-layer inductive graph regressor, achieving a top-3 performance on the Open Graph Benchmark Large-Scale Challenge (OGB-LSC).

The authors suggest two main explanations for the success of Noisy Nodes. First, it adresses oversmoothing because successful denoising requires diversity amongst the nodes features in order to match the diversity in the noise targets σ_i. Second, denoising autoencoders [23] have a long history in pre-training of neural networks for representation and manifold learning. In the case of Noisy Nodes, the auxiliary autoencoder denoising task is done during training but the same principles apply: denoising encourages the network to learn a representation that is robust to partial corruption of the input pattern.

4.2 Improvement: Pre-training as denoising autoencoder

Building up on Noisy Nodes, Zaidi et al. [26] propose a pre-training of GNNs based on denoising. The empirical performances of Noisy Nodes suggest that denoising allows to learn robust and meaningful latent representations of the molecules. Since Noisy Nodes performs denoising as an auxiliary task during training, the representation learning benefits of denoising are limited to the downstream dataset on which the model is trained. Zaidi et al. [26] propose to rather perform denoising as a pre-training objective on a large, unlabelled dataset of atomic structures.

Starting with an input molecule $(X, Z) \in \mathbb{R}^{n\times 3} \times \mathbb{N}^n$, we create its noisy version

$$(\tilde{X}, Z) = \{(\tilde{x}_1, z_1), \ldots, (\tilde{x}_n, z_n)\}, \text{ where } \tilde{x}_i = x_i + \sigma\epsilon_i, \epsilon_i \sim \mathcal{N}(0, I_3). \tag{17}$$

The model is trained minimizing the following loss with respect to the parameters of the GNN-based model $\tilde{\Phi}$:

$$\mathbb{E}_{p(\tilde{X},X)} \left[\|\Phi(\tilde{S}) - (\epsilon_1, \ldots, \epsilon_n)\|_2^2 \right], \tag{18}$$

where the distribution $p(\tilde{X}, X)$ corresponds to sampling a molecule (X, Z) from the pre-training dataset and adding noise to it according to (17). It is important to note that, contrary to Godwin et al. [10], the model of Zaidi et al. [26] predicts the noise, not the original or target coordinates. It is also interesting to note that the noise is here added to the atom's positions before the creation of the graph (the graph is just created using the cutoff distance in FAENet), whereas it is added after creation in the original Noisy Nodes paper, as outlined in (16).

The authors provide an insightful physical interpretation, showing that the denoising objective corresponds to learning a molecular force field. They compared a common GNN, Graph Net Simulator [20] with Noisy Nodes trained from scratch versus using pre-trained parameters. The pre-training technique enabled to utilize existing large datasets of 3D structures to improve performance on various downstream molecular property prediction tasks, setting a new state-of-the-art in some cases such as the QM9 dataset. Interestingly, Liao et al. [18], authors of Equiformer v2 do not mention Noisy Nodes whereas they mention having tested it in Equiformer v1. Yet, they mention the pre-training via denoising as a potential improvement of their model.

4.3 Improvement: Ewald-based Long-Range Message Passing

The problem of oversmoothing in GNNs forces typical architectures to be very shallow compared to other Neural Network architectures. In the case of FAENet, going further than 5 layers does not improve the performances of the models. One of the issues is that the message passing mechanism in GNNs allows to see through k-nodes neighbours after exactly k iterations. This is therefore not efficient to model long-term interactions which are proven to be important to capture the reality of molecular dynamics [24]. An approach has been proposed as an adaption to the *Ewald summation* method but specifically designed for message passing called *Ewald message passing* [15]. The main idea behind Ewald summation is to decompose the electrostatic interaction potential with a given charge into a short-range interaction and a long-range interaction term. The short-range contribution can be computed with real spatial features and the long-range contribution is computed using a Fourier transform. This principle is illustrated in Figure 3. This allows for computational methods in electrostatics to converge faster and with a higher accuracy because the long-range interaction becomes more tractable.

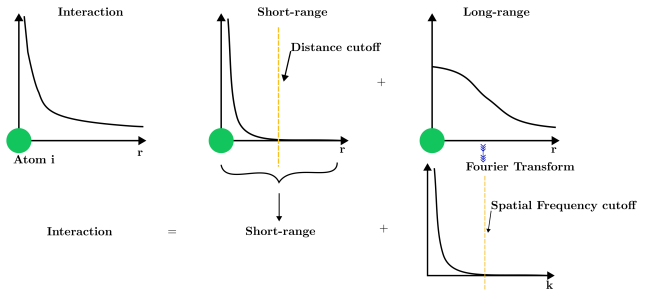

Figure 3: Ewald Summation interactions. The interaction term (left) is the result of the short-range interaction (middle) and the long-term interaction (right) which are both computed using cutoffs on respectively the real and the Fourier space.

In the case of GNNs, the short-range interaction is already computed by the currently implemented interaction blocks using a distance cutoff between the atoms which omits the negligible parts of this interaction for further atoms. However, the heavy-tail of long-range interactions is then reported to a new term in real space, which doesn't diverge anymore for closer atoms. It can then be computed using the same cutoff idea but in the Fourier space where a *spatial frequency cutoff* is used to make it tractable.

Periodic case. In the case where there exists a spatial periodic tiling of materials (OC20 for example), it is possible to define the set of periodic cells localization $\Lambda = \{\lambda_1\mathbf{v_1}, \lambda_2\mathbf{v_2}, \lambda_3\mathbf{v_3} \mid (\lambda_1, \lambda_2, \lambda_3) \in \mathbb{Z}^3\}$, where $\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}$ define the periodic cell lattice, similarly to the periodic interval in the 1D case. In the real space, the long-range interaction component would be written as a sum over all of the elements on the infinite tiling, which can be decomposed as a Fourier series expansion using the reciprocal lattice Λ'. This reciprocal lattice would be similar to the 2π in the 1D case of the Fourier transform. It is the periodic space of all the wavevectors of the Fourier series. This results in the proposed expansion for Ewald message passing:

$$M^{lr}(x_i) = \sum_{\mathbf{k}\in\Lambda'} \exp\left(i\mathbf{k}^T x_i\right) \cdot \sum_{j\in\mathcal{S}} h_j \exp\left(-i\mathbf{k}^T x_j\right) \cdot \hat{\Phi}^{lr}(||\mathbf{k}||) \tag{19}$$

where $M^{lr}(x_i)$ corresponds to the long-range message computed at node i from all of the nodes in the system S, and $\hat{\Phi}^{lr}$ is a learned Fourier coefficient of a radial basis function representing the interaction. The cutoff in the Fourier basis c_k is then set to make the sum finite over the set $\{k \in \Lambda', ||k|| \leq c_k\}$. Since the number of wavevectors used for the computation is finite, $\hat{\Phi}(\|\mathbf{k}\|)$ is learned for every $\mathbf{k}$. Since Ewald summation applies on periodic structures, the authors of Ewald message passing Kosmala et al. [15] propose tricks to deal with the aperiodic case by assuming an infinite tiling.

Results on FAENet. After comparing the training curves for FAENet with and without this module, we notice that the behaviour is almost strictly similar while the number of parameters of the model increases by 50%. One assumption is that, since FAENet does not incorporate equivariant interaction layers in the architecture, the graph representation vectors h_j do not have the same restrictions

Model	Throughput (sample/s)	MAE (eV)	MSE (eV)
SchNet	3070	0.66	0.99
DimeNet++	251	0.57	0.93
FAENet SFA	2944	0.55	0.79
FAENet SFA+Ewald MP	2476	0.56	0.81

Table 1: Comparison of the models SchNet [21], DimeNet++ [8], FAENet with Stochastic Frame-Averaging (SFA) and with Ewald Message Passing. Metrics used are the throughput (number of samples per second), the Mean Absolute Error (MAE) and the Mean Squared Error (MSE) evaluated on the OC20 IS2RE validation ID dataset.

as in Kosmala et al. [15] models, which leads to the same learning pattern whether we use Ewald message passing or not.

4.4 Recap: schema of proposed architecture modifications

A recap of the proposed modifications can be found in Figure 4, directly adapted from Duval et al. [7].

5 EXPERIMENTS

5.1 OC20

We try to run a few experiments to test the different model improvements we proposed earlier, while validating the results obtained for FAENet and other competing models. Table 1 presents the obtained metrics for all the models when trained on the OC20 [4] train split and evaluated on the In Domain (ID) split for the IS2RE task. Note that there also exist three other splits for the validation dataset of OC20: Out-of-domain for the adsorbate (OOD-ADS), the catalyst (OOD-CAT) and for both (OOD-BOTH) which are used in section 5.2. One of the main result is that the throughput (number of processed samples per second) of FAENet is much better than DimeNet++ [8]. This illustrates the main advantage of FAENet: it allows for high throughput while achieving a performance competing with state-of-the-art models. We also show that Ewald message slows down the model but does not improve the MAE.

5.2 Improvement: OC20-dense

The main goal of the evaluation on the previous datasets was to test how much the models were able to learn the different DFT objectives. Similarly to the behaviour of DFT, the predictions of the relaxed state will vary depending on the relative position of the adsorbate with respect to the crystal structure. These computations lead to the local relaxed state from the given initial state as illustrated in Figure 5. However, the main interest behind material discovery is to find the global minimal energy of the binding between the adsorbate and the catalyst.

OC20-Dense. An evaluation framework specifically designed for that purpose has been proposed by Lan* et al. [16]. In the OC-20 dataset, the data sampling consisted in checking many different molecules and crystals combinations. The goal of the OC-20 dense dataset is to provide samples, for the same adsorbate-catalyst system, of various initial configurations using both DFT heuristics and random positions. The hope is that, with such a dense sampling of the space of initial configurations, the relaxed energy will be closer to the global minimal ground truth energy. The authors introduce a new metric to compare the models, called success rate. The prediction of the model on an adsorbate-catalyst couple is considered to be successful if the smallest relaxed energy prediction is both valid and within a success threshold from the minimal known energy from DFT on the dataset or smaller. The predicted relaxed structure is said to be valid when it is within a defined threshold (we take similarly to the authors 0.1 eV) of the relaxed energy predicted by one iteration of DFT. One of the bottlenecks of this benchmark is that when initial configurations that are not in the dataset are explored, it is necessary to be able to compute one step of DFT to validate the prediction. In our case, we will have to limit ourselves to the dense sampling proposed in the dataset to benchmark FAENet.

Results. We extract the validation dataset provided by OC20-Dense and use it to test different models such as DimeNet++ [8] or SchNet [21] pre-trained on OC20 for the IS2RE task. To do so, we first need to split the validation dataset into ID, OOD-ADS, OOD-CAT, OOD-BOTH splits. We then train all models on the training set of OC20, and evaluate them for the different validation splits of OC20-Dense . Table 2 shows the results obtained both in terms of success rate and Mean Absolute Error (MAE) in eV. We also fine-tuned FAENet (*FAENet-ft*) on the ID distribution of the dense dataset. This allows to check whether having a dense sampling of the same systems would improve the model's ability to learn about the underlying chemical process or to find the global minimum. Although the fine-tuned model seems to perform similarly to the original one, the success rates indicates that seeing different initial configurations might help *stabilize* the predictions in a sense. Indeed, our interpretation of this would be that since information about different initial configurations at fixed properties were fed to it, it was able to generalize a little more on some configurations that were not correctly captured by the model.

Main takeaways. While it is interesting to benchmark these models for this new task, the main conclusions can be drawn by looking at the low success rate for all the models. Indeed, all of the models seem to perform very poorly in the task of predicting the global minimal energy for a given system and the variability might just be noise. Fine-tuning has shown that the architectures must be adapted for this specific task and that stronger prior about the impact of the position of the adsorbate should be incorporated. It could be interesting to develop tests of sensitivity with respect this parameter similarly to the Geometric GNNs expressivity tests.

6 CONCLUSION

In this report, we have discussed the main ideas of FAENet, highlighting some of its limitations. We have identified areas where FAENet could benefit from enhancements, to propose a series of modifications aimed at optimizing its performance. These proposed changes are inspired by recent research publications, and were selected to address the specific shortcomings we identified. We

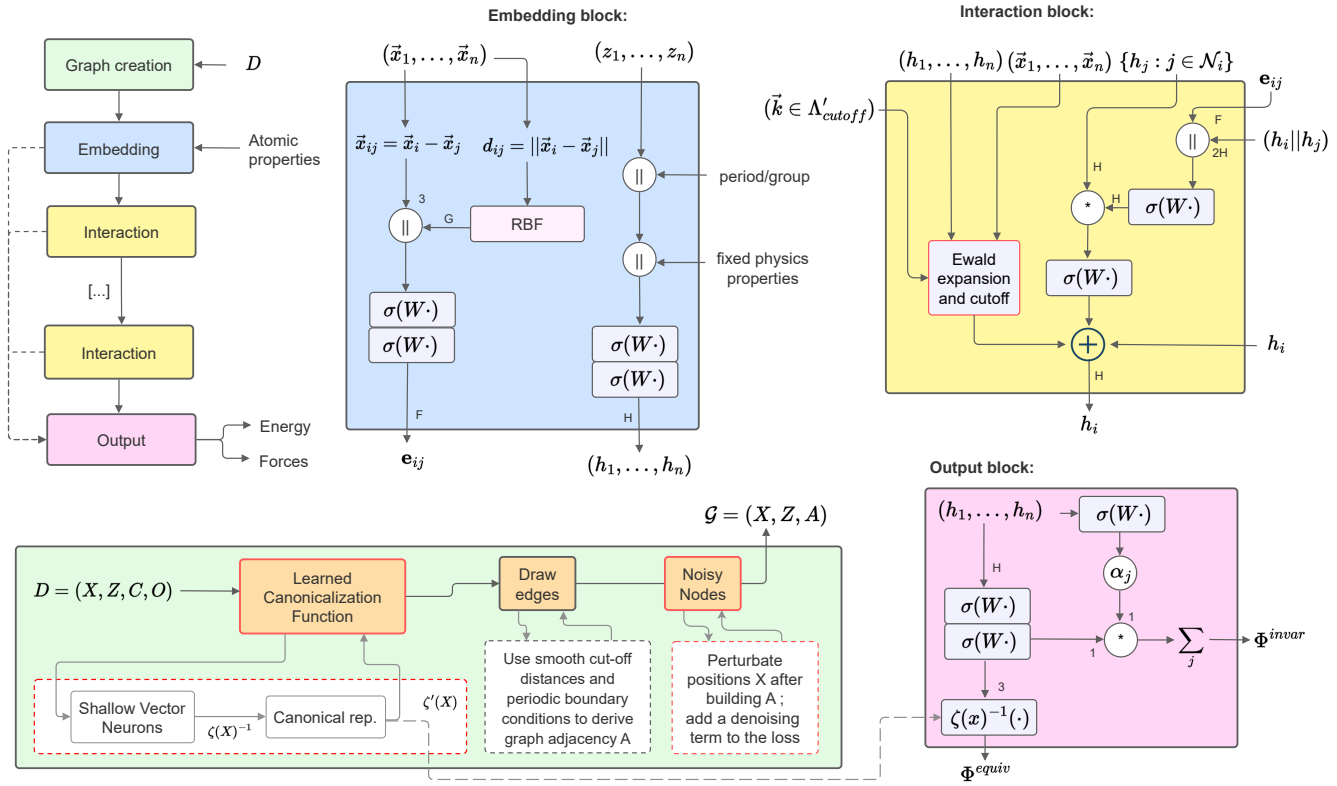

Figure 4: Overview of the proposed architecture, directly adapted from FAENet architecture [7], with modifications highlighted in red. The proposed model takes as input an input sample D, and outputs invariant (resp. equivariant) predictions $\tilde{\Phi}^{invar}$ (resp. $\tilde{\Phi}^{equiv}$), by passing through the steps described in the Overall pipeline. Notations are the same as in [7]: $*$ is the continuous convolution filter described in (6), $\|$ is the concatenation operation, $\sigma(W\cdot)$ is a one-layer perceptron with swish activation function. In the Interaction layer, superscripts indicating that inputs come from the previous layer (e.g., $h_j^{(l-1)}$) are not displayed for readability. Note that when applying the Noisy Nodes operation, the loss must be adapted as written in equation (16). A Noisy Pre-training may be used, in which case weights of all components are first trained following instructions in subsection 4.2.

	ID		OOD-ADS		OOD-CAT		OOD-BOTH	
Model	SR (%)	MAE (eV)	SR (%)	MAE (eV)	SR (%)	MAE (eV)	SR (%)	MAE (eV)
SchNet	7.79	0.71	5.30	0.72	7.22	0.73	6.01	0.74
DimeNet++	8.20	0.49	5.90	0.52	7.42	0.51	4.89	0.55
FAENet	7.79	0.47	4.89	0.51	8.04	0.50	7.29	0.52
FAENet-ft	(Train)	(Train)	7.94	0.51	8.66	0.48	8.91	0.52

Table 2: Comparison of models across different dense datasets: ID, OOD-ADS, OOD-CAT, OOD-BOTH.

believe that these modifications could enhance the efficiency of FAENet, and we hope to see them tested in the future.

REFERENCES

[1] Ravichandra Addanki, Peter W. Battaglia, David Budden, Andreea Deac, Jonathan Godwin, Thomas Keck, Wai Lok Sibon Li, Alvaro Sanchez-Gonzalez, Jacklynn Stott, Shantanu Thakoor, and Petar Veličković. 2021. Large-scale graph representation learning with very deep GNNs and self-supervision. arXiv:2107.09422 [cs.LG]

[2] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. 2021. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478* (2021).

[3] Tianle Cai, Shengjie Luo, Keyulu Xu, Di He, Tie-yan Liu, and Liwei Wang. 2021. Graphnorm: A principled approach to accelerating graph neural network training. In *International Conference on Machine Learning*. PMLR, 1204–1215.

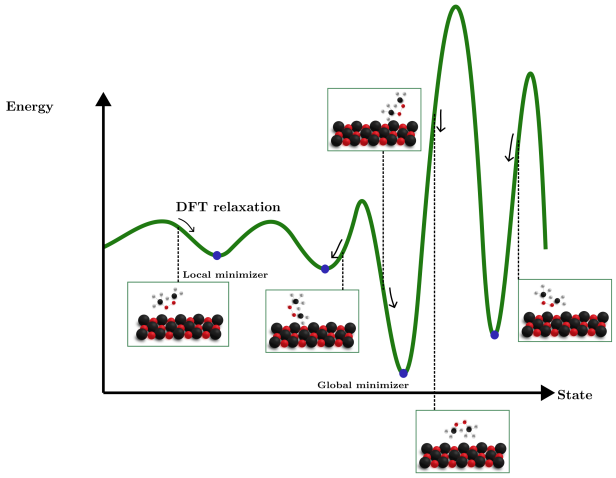

Figure 5: Relaxation energy for different configurations

[4] L. Chanussot, Abhishek Das, Siddharth Goyal, Thibaut Lavril, Muhammed Shuaibi, M. Rivière, Kevin Tran, Javier Heras-Domingo, Caleb Ho, Weihua Hu, Aini Palizhati, Anuroop Sriram, Brandon Wood, Junwoong Yoon, Devi Parikh, C. L. Zitnick, and Zachary W. Ulissi. 2020. The Open Catalyst 2020 (OC20) Dataset and Community Challenges. *ACS Catalysis* (2020). https://doi.org/10.1021/acscatal.0c04525

[5] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J Guibas. 2021. Vector neurons: A general framework for so (3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 12200–12209.

[6] Alexandre Duval, Simon Mathis, Chaitanya K. Joshi, and Victor Schmidt. 2023. A Hitchhiker's Guide to Geometric Graph Neural Networks for 3D Atomic Systems.

[7] Alexandre Duval, V. Schmidt, A. Garcia, Santiago Miret, Fragkiskos D. Malliaros, Y. Bengio, and D. Rolnick. 2023. FAENet: Frame Averaging Equivariant GNN for Materials Modeling. *International Conference on Machine Learning* (2023). https://doi.org/10.48550/arXiv.2305.05577

[8] Johannes Gasteiger, Shankari Giri, Johannes T. Margraf, and Stephan Günnemann. 2020. Fast and Uncertainty-Aware Directional Message Passing for Non-Equilibrium Molecules. *arXiv preprint arXiv: 2011.14115* (2020).

[9] Johannes Gasteiger, Janek Groß, and Stephan Günnemann. 2019. Directional Message Passing for Molecular Graphs. In *International Conference on Learning Representations*.

[10] Jonathan Godwin, Michael Schaarschmidt, Alexander Gaunt, Alvaro Sanchez-Gonzalez, Yulia Rubanova, Petar Veličković, James Kirkpatrick, and Peter Battaglia. 2022. Simple GNN Regularisation for 3D Molecular Property Prediction Beyond. arXiv:2106.07971 [cs.LG]

[11] Johannes Hoja, L. Medrano Sandonas, Brian G Ernst, Á. Vázquez-Mayagoitia, Robert A. DiStasio Jr., and A. Tkatchenko. 2021. QM7-X: A comprehensive dataset of quantum-mechanical properties spanning the chemical space of small organic molecules. *Scientific Data* (2021). https://doi.org/10.1038/s41597-021-00812-2

[12] Weihua Hu, Muhammed Shuaibi, Abhishek Das, Siddharth Goyal, Anuroop Sriram, Jure Leskovec, Devi Parikh, and C. Lawrence Zitnick. 2021. ForceNet: A Graph Neural Network for Large-Scale Quantum Calculations. *arXiv preprint arXiv: 2103.01436* (2021).

[13] Chaitanya K. Joshi and Simon V. Mathis. 2023. On the Expressive Power of Geometric Graph Neural Networks. *International Conference on Machine Learning* (2023). https://doi.org/10.48550/arXiv.2301.09308

[14] Sékou-Oumar Kaba, Arnab Kumar Mondal, Yan Zhang, Yoshua Bengio, and Siamak Ravanbakhsh. 2023. Equivariance with learned canonicalization functions. In *International Conference on Machine Learning*. PMLR, 15546–15566.

[15] Arthur Kosmala, Johannes Gasteiger, Nicholas Gao, and Stephan Günnemann. 2023. Ewald-based Long-Range Message Passing for Molecular Graphs. *arXiv preprint arXiv:2303.04791* (2023).

[16] Janice Lan*, Aini Palizhati*, Muhammed Shuaibi*, Brandon M Wood*, Brook Wander, Abhishek Das, Matt Uyttendaele, C Lawrence Zitnick, and Zachary W Ulissi. 2022. AdsorbML: Accelerating Adsorption Energy Calculations with Machine Learning. *arXiv preprint arXiv:2211.16486* (2022).

[17] Yi-Lun Liao and Tess Smidt. 2023. Equiformer: Equivariant Graph Attention Transformer for 3D Atomistic Graphs. arXiv:2206.11990 [cs.LG]

[18] Yi-Lun Liao, Brandon Wood, Abhishek Das, and Tess Smidt. 2023. EquiformerV2: Improved Equivariant Transformer for Scaling to Higher-Degree Representations. arXiv:2306.12059 [cs.LG]

[19] Omri Puny, Matan Atzmon, Edward J Smith, Ishan Misra, Aditya Grover, Heli Ben-Hamu, and Yaron Lipman. 2021. Frame Averaging for Invariant and Equivariant Network Design. In *International Conference on Learning Representations*.

[20] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. 2020. Learning to Simulate Complex Physics with Graph Networks. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 8459–8468. https://proceedings.mlr.press/v119/sanchez-gonzalez20a.html

[21] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. 2017. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems* 30 (2017).

[22] Kristof T. Schütt, Oliver T. Unke, and Michael Gastegger. 2021. Equivariant message passing for the prediction of tensorial properties and molecular spectra. arXiv:2102.03150 [cs.LG]

[23] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning proceedings*.

[24] Brad A Wells and Alan L Chaffee. 2015. Ewald summation for molecular simulations. *Journal of chemical theory and computation* 11, 8 (2015), 3684–3695.

[25] Oliver Wieder, Stefan Kohlbacher, Mélaine Kuenemann, Arthur Garon, Pierre Ducrot, Thomas Seidel, and Thierry Langer. 2020. A compact review of molecular property prediction with graph neural networks. *Drug Discovery Today: Technologies* 37 (2020), 1–12. https://doi.org/10.1016/j.ddtec.2020.11.009

[26] Sheheryar Zaidi, Michael Schaarschmidt, James Martens, Hyunjik Kim, Yee Whye Teh, Alvaro Sanchez-Gonzalez, Peter Battaglia, Razvan Pascanu, and Jonathan Godwin. 2022. Pre-training via Denoising for Molecular Property Prediction. arXiv:2206.00133 [cs.LG]

[27] C. Lawrence Zitnick, Lowik Chanussot, Abhishek Das, Siddharth Goyal, Javier Heras-Domingo, Caleb Ho, Weihua Hu, Thibaut Lavril, Aini Palizhati, Morgane Riviere, Muhammed Shuaibi, Anuroop Sriram, Kevin Tran, Brandon Wood, Junwoong Yoon, Devi Parikh, and Zachary Ulissi. 2020. An Introduction to Electrocatalyst Design using Machine Learning for Renewable Energy Storage. *arXiv preprint arXiv: 2010.09435* (2020).