Electrostatics from Laplacian Eigenbasis for Neural Network Interatomic Potentials

Maksim Zhdanov [1 2] **Vladislav Kurenkov** [1 3]

Abstract

In this work, we introduce Φ-Module, a universal plugin module that enforces Poisson's equation within the message-passing framework to learn electrostatic interactions in a self-supervised manner. Specifically, each atom-wise representation is encouraged to satisfy a discretized Poisson's equation, making it possible to acquire a potential ϕ and a corresponding charge density ρ linked to the learnable Laplacian eigenbasis coefficients of a given molecular graph. We then derive an electrostatic energy term, crucial for improved total energy predictions. This approach integrates seamlessly into any existing neural potential with insignificant computational overhead. Our results underscore how embedding a first-principles constraint in neural interatomic potentials can significantly improve performance while remaining hyperparameter-friendly, memory-efficient and lightweight in training.

1. Introduction

In quantum chemistry, the task of correct prediction of atomic energies is paramount, but stands a great challenge due to extensive computational requirements of *ab-initio* methods like Density Functional Theory (DFT) (Hohenberg & Kohn, 1964; Kohn & Sham, 1965). Modern deep learning presents a way to solve this problem with geometric graph neural networks (GNN). GNNs operate on molecular graphs by exchanging messages between nodes and edges, learning meaningful representations in the process. In recent years, a series of molecular modeling methods have been developed (Gasteiger et al., 2021; Wang et al., 2022; Passaro & Zitnick, 2023; Musaelian et al., 2023).

While those models and their alternatives demonstrate competitive performance, they rely on message passing which is local in nature (Dwivedi et al., 2022). The issue arises as molecular interactions are described using both local and non-local interatomic interactions. Local interactions include bond stretching, bending and torsional twists. They can be easily captured by message propagation in modern GNNs for molecular graphs (Zhang et al., 2023). At the same time, non-local interactions like electrostatics or van der Waals forces can span long distances and have cumulative effect (Stone, 2013). The main drawbacks of representations learned by GNNs are over-smoothing (Rusch et al., 2023) and over-squashing (Alon & Yahav, 2020) interfere the precise modeling of non-local interactions.

To tackle the problem of learning non-local interactions, a number of customizations have been proposed for molecular GNNs. Some of those require prior data in the form of partial charges or dipole moments, which is costly to retrieve using DFT (Unke & Meuwly, 2019; Ko et al., 2021), or carry inaccurate information derived from pre-defined empirical rules (Gasteiger & Marsili, 1978). Another distinct direction of research proposes merging of message passing and Ewald summation (Ewald, 1921) to approximate electrostatic interactions (Kosmala et al., 2023; Cheng, 2024).

In this paper, we explore a new viewpoint on the problem of learning non-local atomistic and molecular interactions. Our aim is to learn electrostatic energy in a completely self-supervised manner without any external labeled data. To fulfill this goal, we propose Φ-Module, a universal augmentation module, which can be embedded into any GNN. Φ-Module relies on the connection between the Laplacian of a molecular graph and partial charges to improve the quality of the neural network interatomic potentials.

Our contributions are highlighted as follows:

- We propose Φ-Module, a plugin module for GNNs on molecular graphs, which learns electrostatic information from atomic embeddings with estimation of Laplacian eigenbasis coefficients. See Section 3.
- We demonstrate how Φ-Module improves wide variety of baselines on challenging OE62 and MD22 benchmarks for energy prediction and molecular dynamics

Work done by dunnolab.ai. [1]AIRI [2]NUST MISIS [3]Innopolis University. Correspondence to: Maksim Zhdanov <zhdanov@airi.net>.

Proceedings of the Workshop on Generative AI for Biology at the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

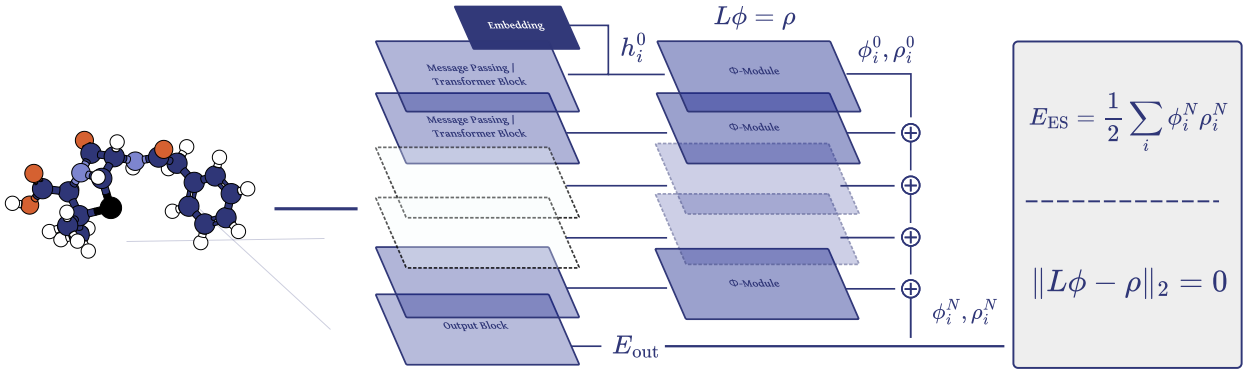


Figure 1: Overview of the proposed Φ-Module. Φ-Module encodes electrostatic constraints based on Poisson's equation into hidden representations of any neural network interatomic potential. Φ-Module is integrated at each step of message passing. It uses lightweight convolutional submodule which we refer to as $\boldsymbol{\alpha}$-Net to estimate coefficients of Laplacian eigenbasis directly from constantly updated atomic representations. Those eigenbasis coefficients are then used to optimize Poisson's equation residual $\|\mathbf{L}\boldsymbol{\phi} - \boldsymbol{\rho}\|_2 = 0$ and compute electrostatic energy term $\mathbf{E}^{\text{ES}}$ making an important contribution to predictions and leading to improved performance on computational chemistry problems. See Section 3.

respectively. On OE62 addition of Φ-Module results in error reductions from 4.5% to 17.8%. For MD22, the proposed solution improves baseline, which achieves best results among other models in 5 out of 14 cases and improves the baseline in 11 out of 14. See Section 4

- We provide valuable insights on the appealing properties of Φ-Module. Namely, its hyperparameter stability, properties of learned charges, physically informative formulation, stability under data scarsity and memory-efficiency crucial to molecular modeling. See Section 4

2. Background

Message Passing Neural Network Potentials. Geometric graph neural networks (GNN) function on molecular graphs with atoms as nodes $V \in \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and atomic bonds as edges $E \in \{(i,j) \mid i \neq j\}$. Each node and edge may include additional features $\mathbf{z}_i \in \mathbb{R}^{d_x}$ and $\mathbf{e}_{ij} \in \mathbb{R}^{d_e}$, which commonly are nuclear charge numbers and distances between nodes. The edges are constructed as a radius graph with a specific cutoff radius r_{c} as a hyperparameter, such that $i \in \mathcal{N}(j)$ if $\|\mathbf{x}_j - \mathbf{x}_i\|_2 \leq r_{\text{c}}$. GNN initially encodes atoms solely on the basis of local properties producing $\mathbf{h}^0 \in \mathbb{R}^F$ features.

In the following steps, GNN refines the initial node representations by applying several iterations of message $\mathbf{m}_i^{(l)}$ aggregations and updates:

$$\mathbf{m}_i^{(l)} = \bigoplus_{j \in \mathcal{N}(i)} \left(\mathcal{M}^l \left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}, \mathbf{e}_{ij} \right) \right)$$
$$\mathbf{h}_i^{(l+1)} = \mathcal{U}^l \left(\mathbf{h}_i^{(l)}, \mathbf{m}_i^{(l)} \right),$$

where $\mathcal{M}^l$ is a learnable function, which constructs the message, $\mathcal{U}^l$ is another learnable function to update the representations of nodes with aggregated messages and $\bigoplus$ is the message aggregation operator, which is typically *sum* or *mean*. Finally, the resulting representations are processed to output the energy $\mathbf{E}^{\text{model}}$. Neural network potentials are commonly optimized to approximate target energy $\mathbf{E}^{\text{target}}$ of a given structure using L1 loss:

$$\mathcal{L}_{\text{model}} = \frac{1}{N} \sum_{i=1}^{N} \left| \mathbf{E}_i^{\text{target}} - \mathbf{E}_i^{\text{model}} \right|.$$

Poisson's Equation for Electrostatics. Electrostatic energy is not directly included into $\mathbf{E}^{\text{model}}$, thus we have to derive it. In case of a discrete molecular graph, electrostatic energy can be acquired from the known potential and partial charges:

$$\mathbf{E}^{\text{ES}} = \frac{1}{2} \sum_i \boldsymbol{\rho}_i \boldsymbol{\phi}_i. \tag{1}$$

The electrostatic potential $\boldsymbol{\phi}$ explains how particles interact through *Coulomb forces*, which are centric to phenomena like bonding, molecular geometry, and protein-ligand interactions. At the same time, $\boldsymbol{\phi}$ and $\boldsymbol{\rho}$ are governed by a discrete Poisson's equation with the *graph Laplacian matrix*

$\mathbf{L}$, which is an approximation of the continuous Laplacian operator by the finite difference method (Smola & Kondor, 2003). This equation is expressed as

$$\mathbf{L}\boldsymbol{\phi} = \boldsymbol{\rho}. \tag{2}$$

The proposed Φ-Module operates on Equation (2) to learn $\boldsymbol{\phi}$ and $\boldsymbol{\rho}$ from atomic messages and obtain the electrostatic energy term $\mathbf{E}^{\text{ES}}$.

3. Φ-Module

In this section, we describe in detail how to encode electrostatic constraints coming from Poisson's equation into representations learned by neural network potentials. Firstly, we explain the importance of learning $\boldsymbol{\phi}$ and $\boldsymbol{\rho}$ in the eigenbasis of $\mathbf{L}$. Next, α-Net is introduced to learn the spectral coefficients essential to obtain the solution of the equation. Finally, we theoretically prove the appealing properties of the spectral decomposition approach in comparison with the direct learning of Poisson's equation components.

Spectral Decomposition of Laplacian. To infuse physical knowledge, resulting in improved learning dynamics, we propose to derive potential $\boldsymbol{\phi}$ and charges $\boldsymbol{\rho}$ in an eigenbasis of Laplacian $\mathbf{L}$. Note that $\mathbf{L}$ is identical for different 3D compositions of the same molecule, therefore we weigh Laplacian values by interatomic instances $\mathbf{d}_{ij} = \|\mathbf{x}_j - \mathbf{x}_i\|_2$ to be able to differentiate between molecular conformations.

Recall that $\mathbf{L}$ in Equation (2) is symmetric positive-semidefinite. Therefore, it can be decomposed as $\mathbf{L} = U\Lambda U^\top$, where $U = [\mathbf{u}_1, \ldots, \mathbf{u}_n]$ are orthonormal eigenvectors and $\Lambda = diag(\lambda_1, \ldots, \lambda_n)$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$ is the diagonal matrix of eigenvalues of $\mathbf{L}$.

Any vector $v \in \mathbb{R}^N$ can be expanded in the basis of U as $v = U\boldsymbol{\alpha}$, where $\boldsymbol{\alpha}$ is the eigenbasis coefficients of $\mathbf{L}$. We expand the Poisson's Equation using the spectral decomposition of $\mathbf{L}$ given potential and charge eigenbasis projections as $\phi = U\boldsymbol{\alpha}$ and $\rho = U\beta$:

$$\begin{aligned}
\mathbf{L}\boldsymbol{\phi} &= \boldsymbol{\rho} \\
\mathbf{L}(U\boldsymbol{\alpha}) &= U\beta \\
(U\Lambda U^\top)(U\boldsymbol{\alpha}) &= U\beta \\
U\Lambda\boldsymbol{\alpha} &= U\beta.
\end{aligned}$$

As U is invertible, we proceed as

$$\begin{aligned}
U\Lambda\boldsymbol{\alpha} &= U\beta \\
\Lambda\boldsymbol{\alpha} &= \beta \\
\lambda_i\boldsymbol{\alpha}_i &= \beta_i,
\end{aligned}$$

which leads to the simple linear relation:

$$\boldsymbol{\phi} = U\boldsymbol{\alpha} \tag{3}$$

$$\boldsymbol{\rho} = U\Lambda\boldsymbol{\alpha}. \tag{4}$$

Hence, both the potential $\boldsymbol{\phi}$ and the charge distribution $\boldsymbol{\rho}$ can be viewed as functions of the reduced parameter vector $\boldsymbol{\alpha} \in \mathbb{R}^k$, which can be retrieved in the data-driven manner from node representations. In principle, U can be truncated to the first k eigenvectors, effectively reducing the dimensionality of the problem while at the same time putting emphasis on the important low-frequency subspace of the Laplacian.

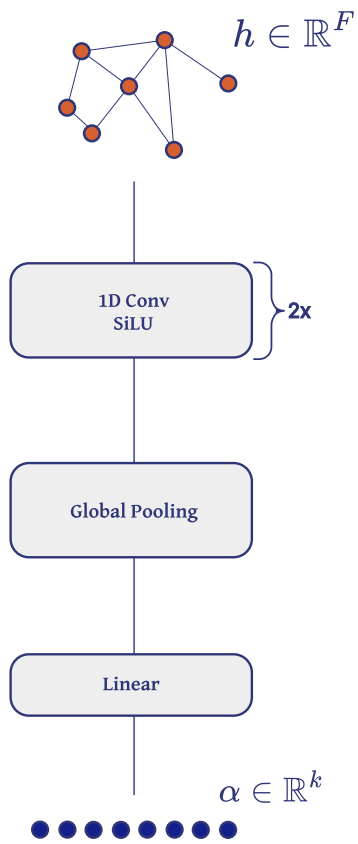


Figure 2: α-Net. It transforms dense atomic representations into sparser coefficients of Laplacian's eigenbasis to acquire potentials and charges. See Section 3.

Self-Supervised Learning of Potential and Charges. Equations 3 and 4 highlight that we need to estimate eigenbasis coefficients $\boldsymbol{\alpha}$ to calculate Poisson's equation residual. We propose learning $\boldsymbol{\alpha}$ from node representations $\mathbf{h}$ using convolutional subnetwork called $\boldsymbol{\alpha}$-Net denoted $\boldsymbol{\alpha}_\theta$. $\boldsymbol{\alpha}$-Net (Figure 2) consists of a pair of 1D convolutional layers combined with global pooling to map node embeddings to a distinct number of eigenvalues.

This lightweight architecture consistently computes the eigenbasis coefficients to update $\boldsymbol{\phi}$ and $\boldsymbol{\rho}$ at each iteration of the message passing. The specific design allows us to

operate on hidden dimensions of $\mathbf{h}$ to compress the most essential information into a low-dimensional representation. We calculate $\boldsymbol{\phi}$ and $\boldsymbol{\rho}$ using Equation (3) and Equation (4) and $\boldsymbol{\alpha}$ coefficients obtained from $\boldsymbol{\alpha}$-Net. In the initial step of message passing, $\boldsymbol{\alpha}$ is computed from the node features after first message passing step $\mathbf{h}^1$. In the subsequent steps potential and charges are aggregated via summation operation as $\boldsymbol{\phi}^N = \boldsymbol{\phi}^{N-1} + U\boldsymbol{\alpha}_\theta(\mathbf{h}^N)$ and $\boldsymbol{\rho}^N = \boldsymbol{\rho}^{N-1} + U\Lambda\boldsymbol{\alpha}_\theta(\mathbf{h}^N)$.

We compute residual $\mathcal{L}_{\text{PDE}} = \beta\|\mathbf{L}\boldsymbol{\phi} - \boldsymbol{\rho}\|_2$ in order for $\boldsymbol{\phi}$ and $\boldsymbol{\rho}$ to satisfy Equation (2), where β is a hyperparameter controlling the impact of the Φ-Module. If a training dataset consists of neutral molecules, we apply an additional constraint to enforce net zero charge $\mathcal{L}_{\text{net}} = \gamma \sum_i \boldsymbol{\rho}_i$, where γ is a hyperparameter. The final training objective for energy prediction is $\mathcal{L} = \mathcal{L}_{\text{model}} + \beta\mathcal{L}_{\text{PDE}} + \gamma\mathcal{L}_{\text{net}}$.

Electrostatic Energy. After final message passing step we calculate electrostatic term $\mathbf{E}^{\text{ES}}$ using Equation (1). As target energy may be rescaled against the baseline energy, we rescale the electrostatic term, resulting in $\kappa\mathbf{E}^{\text{ES}} + \omega$ with learnable offset κ and bias ω. The complete energy is obtained as a combination of energy $\mathbf{E}^{\text{model}}$ retrieved from the model and electrostatic term as $\mathbf{E} = \mathbf{E}^{\text{model}} + \kappa\mathbf{E}^{\text{ES}} + \omega$.

Theoretical Justification. A natural question arises: '*Can we predict $\boldsymbol{\phi}$ and $\boldsymbol{\rho}$ directly from node embeddings without spectral projection?*' We address this in Theorem 3.1 below.

Theorem 3.1 (Spectral Projection Consistency and Robustness)**.** *Let $L \in \mathbb{R}^{n\times n}$ be a symmetric positive semi-definite graph Laplacian with eigendecomposition $L = U\Lambda U^\top$, where $U = [\mathbf{u}_1, \dots, \mathbf{u}_n]$ are orthonormal eigenvectors and $\Lambda = diag(\lambda_1, \dots, \lambda_n)$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$. Consider the Poisson equation $L\phi = \rho$.*

Define two approximation methods:

1. ***Direct Learning (DL):*** *Independently parameterize $\phi_{DL} \in \mathbb{R}^n$ and $\rho_{DL} \in \mathbb{R}^n$*
2. ***Spectral Projection (SP):*** *Parameterize coefficients $\alpha \in \mathbb{R}^k$, compute $\phi_{SP} = U_k\alpha$ and $\rho_{SP} = U_k\Lambda_k\alpha$, where $U_k \in \mathbb{R}^{n\times k}$ contains the first k eigenvectors of L, and $\Lambda_k = diag(\lambda_1, \dots, \lambda_k)$*

Assume $\lambda_k > 0$. Then:

1. ***Exact Consistency:*** *The SP method achieves $\|L\phi_{SP} - \rho_{SP}\|_2 = 0$ with k parameters, while DL requires $2n$ parameters and cannot guarantee zero residual*
2. ***Noise Robustness:*** *For $\rho = U_k\Lambda_k\alpha + \epsilon$, SP satisfies $\|L\phi_{SP} - \rho\|_2 = \|\epsilon\|_2$, while DL amplifies noise such that $\|\phi_{DL} - \phi_{SP}\|_2 \leq \frac{1}{\lambda_n}\|\epsilon\|_2$*

Detailed proof can be found in Appendix G.

Implementation Details. We implement spectral decomposition of $\mathbf{L}$ using the "Locally Optimal Block Preconditioned Conjugate Gradient" method (LOBPCG) (Knyazev, 2001). LOBPCG enables us to compute only a selected amount of eigenvalues and gives the opportunity to process large macromolecules with the Φ-Module. This decision also keeps us away from the ambiguity of invariance and sorting of eigenvalues and eigenvectors during their computations - we strictly get k selected eigenvalues and their corresponding eigenvectors without the need to sort them anyhow. Block-diagonal nature of $\mathbf{L}$ and independence of its blocks allow us to compute eigendecomposition once for a single batch in an efficient vectorized manner without relying on any paddings.

The pseudocode for integration of the Φ-Module can be seen in Appendix B. The proposed augmentation fits into any neural network that iteratively operates on atomic representations.

4. Experiments

In this section, we conduct diverse experiments to establish the importance of Φ-Module. Firstly, performance of networks injected with the proposed module are tested against corresponding baselines on popular quantum chemical datasets. Next, we demonstrate that the Φ-Module exhibits robust hyperparameter stability, requiring minimal tuning to achieve improved performance. We further analyze how the learned charge distributions compare to the Hirshfeld charge system, which incorporates electronic-level information. Additionally, we show clear benefits from the memory-scaling dynamics of the Φ-Module and provide evidence that current architectural choices encode physically meaningful priors. Finally, we evaluate the model in data-scarce regimes and show that it outperforms baselines even with limited supervision.

In the following experiments, we modify various baseline models by integrating the Φ-Module and denote the resulting models with the prefix "**Φ**-". For example, "Network" becomes "**Φ**-Network". The Φ-Module introduces four tunable hyperparameters: k — the number of eigenvalues; β — the weight of the PDE loss $\mathcal{L}$PDE; γ — the weight of the network loss $\mathcal{L}$net and n_{warm} — the number of warm-up epochs before the Φ-Module is activated. For a detailed overview of the hyperparameter configurations, please refer to Appendix F.

OE62. We start our analysis with the challenging OE62 (Stuke et al., 2020) dataset to demonstrate how Φ-Module enhances neural network interatomic potentials. OE62 features about 62,000 large organic molecules with the energies calculated using Density Functional Theory (DFT). The molecules within OE62 have around 41 atoms on av-

Table 1: Energy MAEs and computation time of baselines and their alternatives with Φ-Module on OE62. The best option for each model is highlighted in bold. See Section 4.

Model	Version	OE62-val MAE, meV ↓	OE62-val Rel, % ↓	OE62-test MAE, meV ↓	OE62-test Rel, % ↓	Mean Epoch Time Runtime, s ↓	Mean Epoch Time Rel, % ↓
SchNet	Baseline	95.5	-	95.5	-	16.91	-
	Φ-SchNet	**85.2**	**10.7**	**84.0**	**12.0**	18.65	10.28
DimeNet++	Baseline	65.1	-	67.1	-	90.66	-
	Φ-DimeNet++	**57.1**	**12.2**	**58.5**	**12.8**	94.36	4.08
PaiNN	Baseline	75.6	-	77.3	-	56.70	-
	Φ-PaiNN	**62.1**	**17.8**	**66.0**	**14.6**	62.24	9.77
GemNet-T	Baseline	63.3	-	63.5	-	179.01	-
	Φ-GemNet-T	**55.1**	**12.9**	**56.1**	**11.6**	187.40	4.68
E_2GNN	Baseline	68.5	-	69.1	-	50.30	-
	Φ-E_2GNN	**65.4**	**4.5**	**65.2**	**5.6**	59.95	19.18

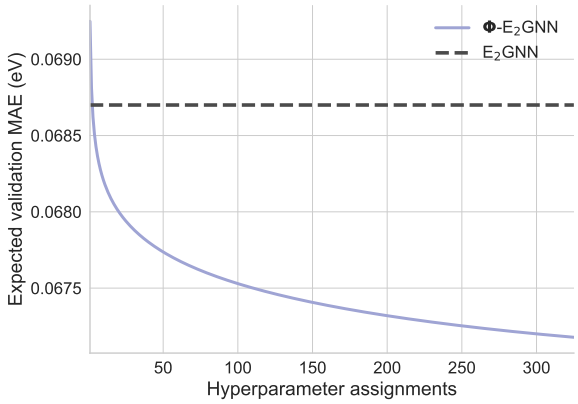


Figure 3: Test expected validation MAE for **Φ**-E_2GNN against the baseline model on OE62. Any choice of selected hyperparameters leads to improved performance, underlining tuning stability of the Φ-Module. See Section 4.

erage and may exceed the size of 20 Å. Dataset is divided into training, validation and testing parts and preprocessed according to the previous studies (Kosmala et al., 2023).

Common baselines namely SchNet (Schütt et al., 2017), DimeNet++ (Gasteiger et al., 2020), PaiNN (Schütt et al., 2021), GemNet-T (Gasteiger et al., 2021) and E_2GNN (Yang et al., 2025) are trained on OE62. Their counterparts with Φ-Module are named accordingly as **Φ**-SchNet, **Φ**-DimeNet++, **Φ**-PaiNN, **Φ**-GemNet-T and **Φ**-E_2GNN. The computational cost is computed as the average time for one epoch given selected hyperparameters in Appendix F.

The results in Table 1 demonstrate that Φ-Module improves performance for each model by a large margin ($\geq 10\%$) and for around 5% for E_2GNN while introducing a modest computational overhead.

MD22. The MD22 (Chmiela et al., 2019) dataset is a comprehensive collection of molecular dynamics (MD) trajectories of biomolecules and supramolecules. It covers a wide range of molecular sizes, with atom counts spanning from 42 to 370 atoms per system. Each dataset represents a single molecule's dynamic behavior, comprising between 5,032 and 85,109 structural snapshots captured over time. The MD22 is split into training, validation and testing sets according to sGDML (Chmiela et al., 2019).

We benchmark the ViSNet (Wang et al., 2022) model and its **Φ**-ViSNet modification on seven presented molecule in Table 2 and demonstrate that our method achieves consistent improvements over the baseline in both energy and force predictions for most of the cases.

Original ViSNet achieves the best results only in 2 out of 14 cases, while **Φ**-ViSNet sets the best results for the 5 metrics of the measured setups. Additionally, **Φ**-ViSNet outperforms basic ViSNet in 11 out of 14 cases. The average computational overhead for the insertion of the Φ-Module is only 9%. Note, that no hyperparameter search was performed for MD22 due to limited available resources, hence the results may be improved in practice.

Hyperparameter Stability. In this section, we study the hyperparameter stability of the Φ-Module. To conduct this experiment we employ *Expected Validation Performance* (EVP) (Dodge et al., 2019) which measures how performance of **Φ**-E_2GNN trained on OE62 changes with the increasing number of hyperparameter assignments.

The hyperparameter search includes k, β, γ and n_{warm}. In

Table 2: Mean absolute errors (MAE) of energy (kcal/mol) and forces (kcal/mol/Å) for seven large molecules on MD22. The best one in each category is highlighted in bold. The runs where **Φ**-ViSNet outperforms baseline are underlined. See Section 4.

Molecule	Type	Diameter	sGDML	SO3KRATES	Allegro	Equiformer	MACE	ViSNet	**Φ**-ViSNet
Ac-Ala3-NHMe	Energy	10.75	0.390	0.337	0.102	0.083	**0.062**	0.102	0.091
	Forces		0.797	0.244	0.107	**0.080**	0.088	0.086	0.082
DHA	Energy	14.58	1.312	0.379	0.115	0.179	0.132	0.072	**0.010**
	Forces		0.747	0.242	0.073	**0.051**	0.065	0.099	0.075
Stachyose	Energy	13.87	4.050	0.442	0.249	0.140	0.124	**0.017**	0.040
	Forces		0.674	0.435	0.097	0.064	0.088	0.107	**0.008**
AT-AT	Energy	17.63	0.724	0.178	0.143	0.131	0.109	0.008	**0.008**
	Forces		0.691	0.216	0.095	0.096	0.099	**0.086**	0.111
AT-AT-CG-CG	Energy	21.29	1.389	0.345	0.393	0.151	0.158	0.149	**0.074**
	Forces		0.703	0.332	0.128	0.125	**0.115**	0.199	0.182
Buckyball catcher	Energy	15.89	1.196	**0.381**	0.526	0.398	0.481	0.937	0.741
	Forces		0.682	0.237	0.089	0.111	**0.085**	0.690	0.731
Double-walled nanotube	Energy	32.39	4.012	0.993	2.210	1.195	1.655	1.023	**0.506**
	Forces		0.523	0.727	0.343	**0.275**	0.396	0.680	0.593

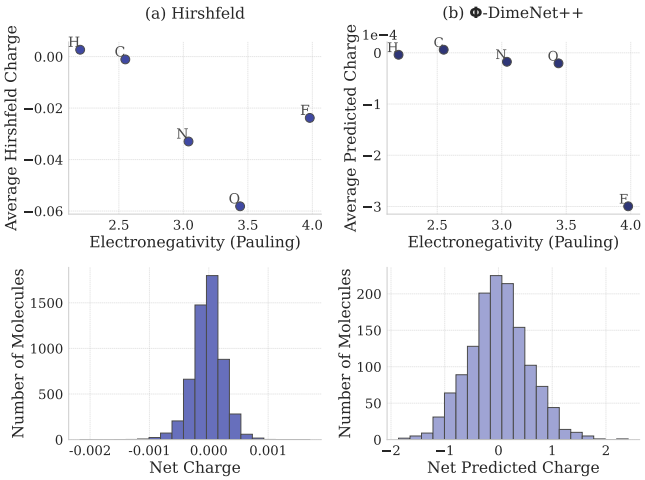


Figure 4: Charge distribution analysis for **Φ**-DimeNet++. The Φ-Module learns charges that correlate with physical principles and resemble the Hirshfeld system. See Section 4.

Figure 3 EVP curve for **Φ**-E$_2$GNN is below E$_2$GNN baseline performance line after hyperparameter search. The plot demonstrates that any configuration of hyperparameters results in improved performance against the baseline. This highlights the practical convenience of Φ-Module in terms of hyperparameter choice. Detailed information on EVP and plots for other models can be examined in Appendix E.

Charge Distribution Analysis. In this paragraph, we analyze the properties of learned potential and charges and compare them against the Hirshfeld charges (Hirshfeld, 1977) for **Φ**-DimeNet++ trained OE62 dataset. The Hirshfeld charges require electron density information, while Φ-Module learns in a self-supervised manner alongside a selected model. We study the relationship between the average predicted charge on H, C, N, O, F atoms and their electronegativity. We selected those atoms based on the number of occurrences in OE62 to acquire stable estimates. In addition, we compare average molecular charges. The hyperparameter settings coincide with those used for Table 1.

It can be seen in Figure 4 that the charges retrieved from **Φ**-DimeNet++ achieve a similar correlation coefficient in comparison with the Hirshfeld charges. The comparison of the distribution of the average total charges shows that both, Φ-Module and Hirshfeld charges are centered at zero.

This experiment demonstrates the physical relevance of charges learned in a self-supervised manner. Moreover, it opens up a new research venue for the future works to refine the charges learned in a self-supervised data-driven manner.

Φ-Module Memory Scaling in Comparison with Ewald Summation. To access one of the crucial benefits of the Φ-Module - *memory efficiency*, we set up experiment to run SchNet, **Φ**-SchNet, SchNet with Ewald message passing block (Kosmala et al., 2023), and SchNet with Neural P3M block (Cheng, 2024) on linear carbyne chains (Liu et al., 2013) of variable sizes from 10^3 to 10^5 atoms. We measure CUDA memory consumption on an NVIDIA 80GB H100 GPU in MBs.

The results can be seen in Figure 6. Ewald and Neural P3M quickly result in out-of-memory (OOM) error and do not scale favorably to large systems. This is an essential problem as electrostatic interactions die off much slower than other forms of long-range forces and are more evident in large systems. On the other hand, Φ-Module demon-

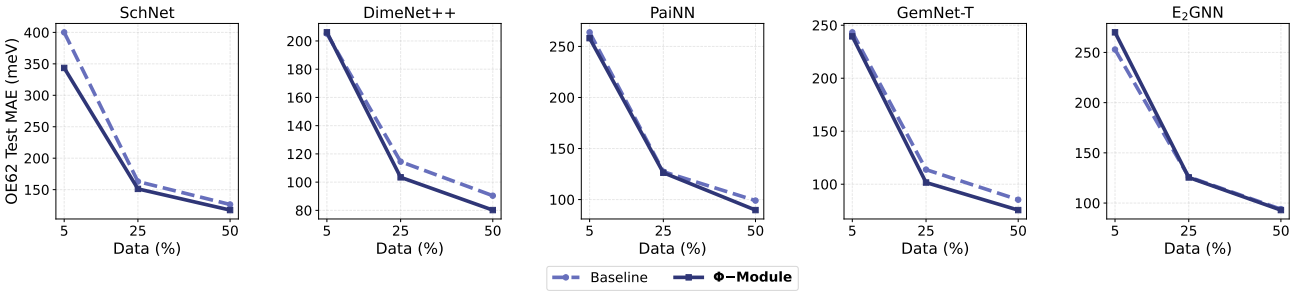


Figure 5: Performance of baseline models and models with Φ-Module in data-scarce setups. Φ-Module outperforms baseline in almost any case for all of the tested setups (5%, 25% and 50% of the initial training data). See Section 4.

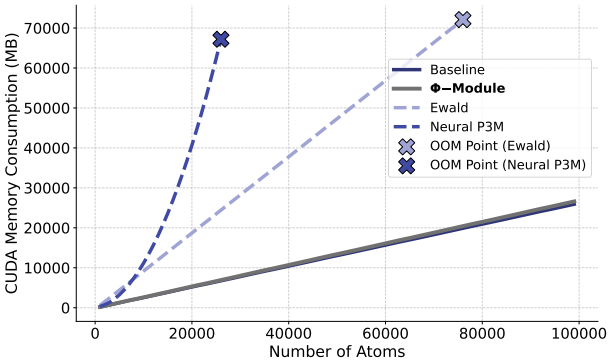


Figure 6: Memory consumption of baseline SchNet, **Φ**-SchNet, Ewald message passing, and Neural P3M on a carbyne chain with gradually increasing size. Out-of-memory (OOM) points are shown as crosses. Ewald and Neural P3M quickly result in OOM, while the Φ-Module demonstrates strong memory efficiency. See Section 4.

strates the same scaling as the baseline model showcasing its potential for extremely large molecules.

Design Choices. In this section, we elaborate on the main design choices made for Φ-Module. We take models used for OE62 and train them while gradually disabling main parts of the proposed method. Firstly, we replace **L** with a random matrix to eliminate any physical grounding. Secondly, we remove the optimization of the residual of Poisson's equation as in Section 3 to test if unconstrained addition of trainable parameters is helpful. In the Figure 7, a distinct trend can be seen of the complete solution for Φ-Module outperforming the version lacking physical grounding. This experiment supports the formulation proposed in this work.

Data Scarcity Configurations. In this experiment, we demonstrate that Φ-Module achieves performance gains over baselines even in data-scarce cases. We train OE62 baselines and their versions with Φ-Module on *5%, 25%, 50%* of initial training data. Results show that usage of Φ-Module leads to improved performance on nearly every setup and model highlighting stability under various data configurations. Refer to Figure 5 for more details.

5. Related Work

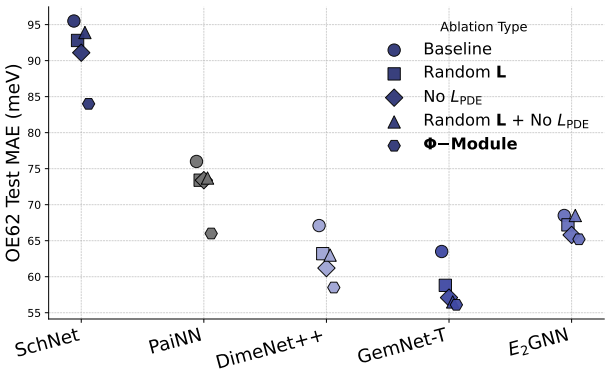


Figure 7: Ablation study on the main design choices for the Φ-Module. We remove structural information from **L** and the optimization of the PDE residual. These "non-physical" variants underperform compared to the baseline and full Φ-Module, highlighting the value of physical priors. See Section 4.

Electrostatic Constraints for Neural Network Potentials. There are a number of attempts to utilize electrostatic interactions with neural network potentials. Some of them rely on effective partial charges of atomic nuclei (Xie et al., 2020; Niblett et al., 2021) or incorporate precomputed electronegativities as a starting point (Ko et al., 2023). An alternative approach involves multipole expansion to express electrostatic potentials without reliance on fixed partial-charge approximation (Thürlemann et al., 2021). Although the approach brings performance benefits, it requires expensive training data with information at electronic level.

In contrast to mentioned methods, Φ-Module does not require any difficult-to-obtain prior information to deliver valuable improvements for neural networks potentials. Electrostatics are learned in the self-supervised manner using lightweight message passing submodule.

Ewald Summation. A separate track of research equips neural network potentials with the electrostatic knowledge via operations related to Ewald summation (Ewald, 1921). For instance, Kosmala et al. (2023) develops Ewald Message Passing, an augmentation to neural message passing with Fourier space interactions and the cutoff in frequency range. Later, Cheng (2024) extrapolated the idea of Particle-Particle-Particle-Mesh (P3M) (Hockney & Eastwood, 2021) to neural message passing setup, resulting in improved speed compared to regular Ewald Message Passing.

Those approaches focus on the incorporation of the Ewald summation into neural network interatomic potentials, which is an orthogonal line of work. Moreover, originally Ewald summation is constrained to periodic crystals and its usage on non-periodic systems relies on the definition of single large supercell, which serves only as an artificial workaround for such data types. In this paper, we discover a different route of efficient self-supervised learning of electrostatics from Poisson's equation formulation itself.

General Poisson Learning. The use of neural networks for solving the Poisson equation began in the mid-1990s, marked by early implementations of multilayer perceptrons to handle the two-dimensional case with Dirichlet boundary conditions (Dissanayake & Phan-Thien, 1994).

In subsequent years, physics-informed neural networks (PINNs) emerged as a powerful approach by embedding the governing differential equations directly into the loss function (Hafezianzade et al., 2023). This methodology has proven especially effective for challenging problems such as the nonlinear Poisson–Boltzmann equation, where traditional numerical methods often struggle with nonlinearity and complex geometries (Mills & Pozdnyakov, 2022).

Recent studies have investigated error correction strategies in neural network-based solvers for differential equations, often using Poisson's equation as a testbed due to its fundamental role as a second-order linear PDE and its broad relevance in theoretical physics (Wright, 2022).

Our work does not aim to solve Poisson's equation explicitly. Instead, we investigate how it can be used to enhance the learning dynamics and performance precision of neural network interatomic potentials.

6. Conclusion and Future Work

We introduced the Φ-Module, a universal and physically grounded framework for incorporating electrostatics into neural interatomic potentials. Our method integrates seamlessly with a wide range of deep learning architectures in computational chemistry, offering stable improvements in energy prediction and molecular dynamics tasks. It also demonstrates favorable memory and computational efficiency, with minimal need for hyperparameter tuning. Despite its strengths, the current implementation relies on partial charge approximations and does not yet account for more expressive electrostatic descriptors, such as multipole expansions or polarizability tensors. Extending the Φ-Module to capture these higher-order effects presents a promising direction for advancing self-supervised learning in quantum chemistry.

References

Alon, U. and Yahav, E. On the bottleneck of graph neural networks and its practical implications. *arXiv preprint arXiv:2006.05205*, 2020.

Cheng, B. Latent ewald summation for machine learning of long-range interactions. *ArXiv*, abs/2408.15165, 2024. URL https://api.semanticscholar.org/CorpusID:271963371.

Chmiela, S., Sauceda, H. E., Poltavsky, I., Müller, K.-R., and Tkatchenko, A. sgdml: Constructing accurate and data efficient molecular force fields using machine learning. *Computer Physics Communications*, 240:38–45, 2019.

Dissanayake, M. and Phan-Thien, N. Neural-network-based approximations for solving partial differential equations. *Communications in Numerical Methods in Engineering*, 10:195–201, 1994. URL https://api.semanticscholar.org/CorpusID:120328171.

Dodge, J., Gururangan, S., Card, D., Schwartz, R., and Smith, N. A. Show your work: Improved reporting of experimental results. *arXiv preprint arXiv:1909.03004*, 2019.

Dwivedi, V. P., Rampášek, L., Galkin, M., Parviz, A., Wolf, G., Luu, A. T., and Beaini, D. Long range graph benchmark. *Advances in Neural Information Processing Systems*, 35:22326–22340, 2022.

Ewald, P. P. Die berechnung optischer und elektrostatischer gitterpotentiale. *Annalen der physik*, 369(3):253–287, 1921.

Gasteiger, J. and Marsili, M. A new model for calculating atomic charges in molecules. *Tetrahedron letters*, 19(34): 3181–3184, 1978.

Gasteiger, J., Giri, S., Margraf, J. T., and Günnemann, S. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. *arXiv preprint arXiv:2011.14115*, 2020.

Gasteiger, J., Becker, F., and Günnemann, S. Gemnet: Universal directional graph neural networks for molecules. *Advances in Neural Information Processing Systems*, 34: 6790–6802, 2021.

Hafezianzade, F., Biagooi, M., and Oskoee, S. N. Physics informed neural network for charged particles surrounded by conductive boundaries. *Scientific Reports*, 13, 2023. URL https://api.semanticscholar.org/CorpusID:255440511.

Hirshfeld, F. L. Bonded-atom fragments for describing molecular charge densities. *Theoretica chimica acta*, 44: 129–138, 1977.

Hockney, R. W. and Eastwood, J. W. *Computer simulation using particles*. crc Press, 2021.

Hohenberg, P. and Kohn, W. Inhomogeneous electron gas. *Physical review*, 136(3B):B864, 1964.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Knyazev, A. V. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM journal on scientific computing*, 23(2):517–541, 2001.

Ko, T. W., Finkler, J. A., Goedecker, S., and Behler, J. General-purpose machine learning potentials capturing nonlocal charge transfer. *Accounts of Chemical Research*, 54(4):808–817, 2021.

Ko, T. W., Finkler, J. A., Goedecker, S., and Behler, J. Accurate fourth-generation machine learning potentials by electrostatic embedding. *Journal of chemical theory and computation*, 2023. URL https://api.semanticscholar.org/CorpusID:258762588.

Kohn, W. and Sham, L. J. Self-consistent equations including exchange and correlation effects. *Physical review*, 140(4A):A1133, 1965.

Kosmala, A., Gasteiger, J., Gao, N., and Günnemann, S. Ewald-based long-range message passing for molecular graphs. In *International Conference on Machine Learning*, pp. 17544–17563. PMLR, 2023.

Liu, M., Artyukhov, V. I., Lee, H., Xu, F., and Yakobson, B. I. Carbyne from first principles: chain of c atoms, a nanorod or a nanorope. *ACS nano*, 7(11):10075–10082, 2013.

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Mills, E. A. and Pozdnyakov, A. Stochastic scaling in loss functions for physics-informed neural networks. *ArXiv*, abs/2208.03776, 2022. URL https://api.semanticscholar.org/CorpusID:251402739.

Musaelian, A., Batzner, S., Johansson, A., Sun, L., Owen, C. J., Kornbluth, M., and Kozinsky, B. Learning local equivariant representations for large-scale atomistic dynamics. *Nature Communications*, 14(1):579, 2023.

Niblett, S. P., Galib, M., and Limmer, D. T. Learning intermolecular forces at liquid–vapor interfaces. *The Journal of chemical physics*, 155(16), 2021.

Passaro, S. and Zitnick, C. L. Reducing so (3) convolutions to so (2) for efficient equivariant gnns. In *International conference on machine learning*, pp. 27420–27438. PMLR, 2023.

Rusch, T. K., Bronstein, M. M., and Mishra, S. A survey on oversmoothing in graph neural networks. *arXiv preprint arXiv:2303.10993*, 2023.

Schütt, K., Kindermans, P.-J., Sauceda Felix, H. E., Chmiela, S., Tkatchenko, A., and Müller, K.-R. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017.

Schütt, K., Unke, O., and Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *International Conference on Machine Learning*, pp. 9377–9388. PMLR, 2021.

Smola, A. J. and Kondor, R. Kernels and regularization on graphs. In *Learning Theory and Kernel Machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003. Proceedings*, pp. 144–158. Springer, 2003.

Stone, A. *The theory of intermolecular forces*. oUP oxford, 2013.

Stuke, A., Kunkel, C., Golze, D., Todorović, M., Margraf, J. T., Reuter, K., Rinke, P., and Oberhofer, H. Atomic structures and orbital energies of 61,489 crystal-forming organic molecules. *Scientific data*, 7(1):58, 2020.

Thürlemann, M., Böselt, L., and Riniker, S. Learning atomic multipoles: Prediction of the electrostatic potential with equivariant graph neural networks. *Journal of chemical theory and computation*, 2021. URL https://api.semanticscholar.org/CorpusID:238582889.

Unke, O. T. and Meuwly, M. Physnet: A neural network for predicting energies, forces, dipole moments, and partial charges. *Journal of chemical theory and computation*, 15 (6):3678–3693, 2019.

Wang, Y., Li, S., He, X., Li, M., Wang, Z., Zheng, N., Shao, B., Liu, T.-Y., and Wang, T. Visnet: an equivariant geometry-enhanced graph neural network with vector-scalar interactive message passing for molecules. *arXiv preprint arXiv:2210.16518*, 2022.

Wright, M. J. H. Enhancing neural network differential equation solvers. *ArXiv*, abs/2301.13146, 2022. URL https://api.semanticscholar.org/CorpusID:256390579.

Xie, X., Persson, K. A., and Small, D. W. Incorporating electronic information into machine learning potential energy surfaces via approaching the ground-state electronic energy as a function of atom-based electronic populations. *Journal of chemical theory and computation*, 16 (7):4256–4270, 2020.

Yang, Z., Wang, X., Li, Y., Lv, Q., Chen, C. Y.-C., and Shen, L. Efficient equivariant model for machine learning interatomic potentials. *npj Computational Materials*, 11 (1):49, 2025.

Zhang, X., Wang, L., Helwig, J., Luo, Y., Fu, C., Xie, Y., Liu, M., Lin, Y., Xu, Z., Yan, K., et al. Artificial intelligence for science in quantum, atomistic, and continuum systems. *arXiv preprint arXiv:2307.08423*, 2023.

Algorithm 1 Message Passing with Φ-Module

Require: Mini-batch $\mathcal{B} = \{G_i = (V_i, E_i)\}_{i=1}^{B}$, # message-passing layers T, modes k
Ensure: Total loss $\mathcal{L}$ for back-propagation

$L \leftarrow \textsc{BlockDiag}\big(\{\textsc{GraphLaplacian}(G_i)\}_{i=1}^{B}\big)$
$(U, \Lambda) \leftarrow \text{LOBPCG}(L, k)$ {batched eigendecomposition}
$\{h_v^0\}_{v \in V} \leftarrow \textsc{Embedding}(\mathcal{B})$
$\phi^0 \leftarrow \mathbf{0}; \quad \rho^0 \leftarrow \mathbf{0}$
for $t = 0$ **to** $T - 1$ **do**
  $\{h_v^{t+1}\} \leftarrow \textsc{MessagePassing}(\{h_v^t\}, \mathcal{B})$
  if $t == 0$ **then**
    $\alpha^0 \leftarrow \textsc{AlphaNet}\big(\{h_v^{t+1}\}\big)$
    $\phi^1 \leftarrow U\,\alpha^0; \quad \rho^1 \leftarrow U\,\Lambda\,\alpha^0$
  else
    $\alpha^t \leftarrow \textsc{AlphaNet}\big(\{h_v^{t+1}\}\big)$
    $\phi^{t+1} \leftarrow \phi^t + U\,\alpha^t$
    $\rho^{t+1} \leftarrow \rho^t + U\,\Lambda\,\alpha^t$
  end if
end for
$\mathbf{E}^{\text{model}} \leftarrow \textsc{ReadoutEnergy}\big(\{h_v^T\}, \mathcal{B}\big)$
$\mathbf{E}^{\text{ES}} \leftarrow \frac{1}{2} \sum_{v \in V_i} (\phi_v^T \odot \rho_v^T)$ {electrostatic energy term}
$\mathbf{r} \leftarrow L\,\phi^T - \rho^T$ {PDE residual}
$\mathcal{L} \leftarrow \underbrace{\ell(\mathbf{E}^{\text{model}} + \mathbf{E}^{\text{ES}}, \mathbf{E}^{\text{target}})}_{\mathcal{L}_{\text{model}}} + \beta \underbrace{\|\mathbf{r}\|_2}_{\mathcal{L}_{\text{PDE}}} + \gamma \underbrace{\sum_{v \in V_i} (\rho_v^T)^2}_{\mathcal{L}_{\text{net}}}$
return $\mathcal{L}$ =0

A. Code Availability

We provide the source code to reproduce the experiments in the supplementary material to the submission as a file archive. The code will be released to public upon acceptance.

B. Pseudocode for Φ-Module

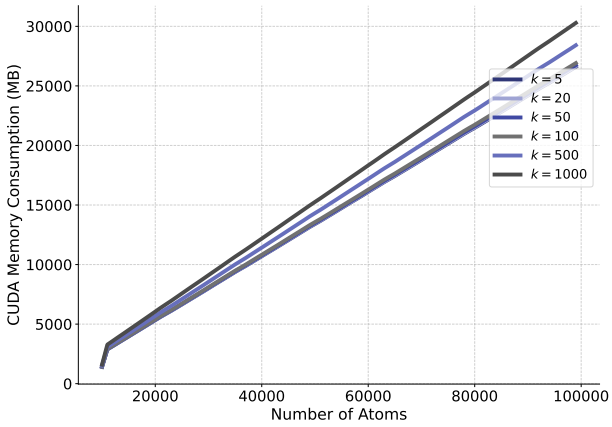
Complete and detailed pseudocode for Φ-Module can be examined in Algorithm 1.

C. Memory And Runtime Scaling with Increasing k.

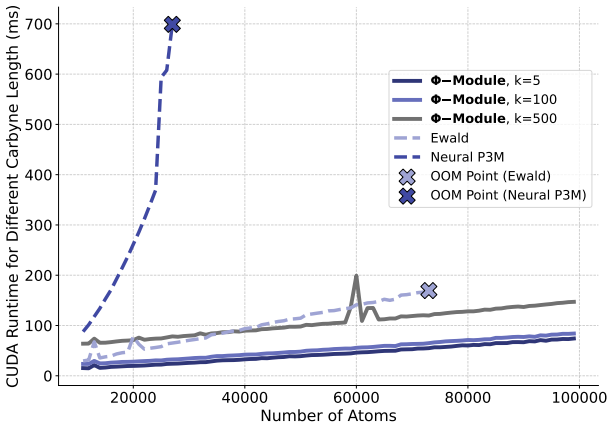
In this experiment, we follow the design of the task involving the linear carbyne chain in Section 4, but test the memory and runtime trends of Φ-Module with respect to the increasing number of estimated eigenvalues. In Figure 7(a), you can see that memory consumption increases at a very slow rate which allows efficient processing of large systems. Moreover, Φ-Module scales favorably in terms of GPU runtime for large systems (starting from 10^4 atoms) in comparison to Ewald and Neural P3M based on Figure 7(b).

D. Additional Analysis of Charge Distributions

In Figure 9, analysis of partial charges can be examined for **Φ**-SchNet, **Φ**-DimeNet++, **Φ**-PaiNN and **Φ**-GemNet-T. While, some of the atoms do not show alignment with widely used Hirshfeld partial charges, on average, the results are correlated and consistent given their self-supervised nature.

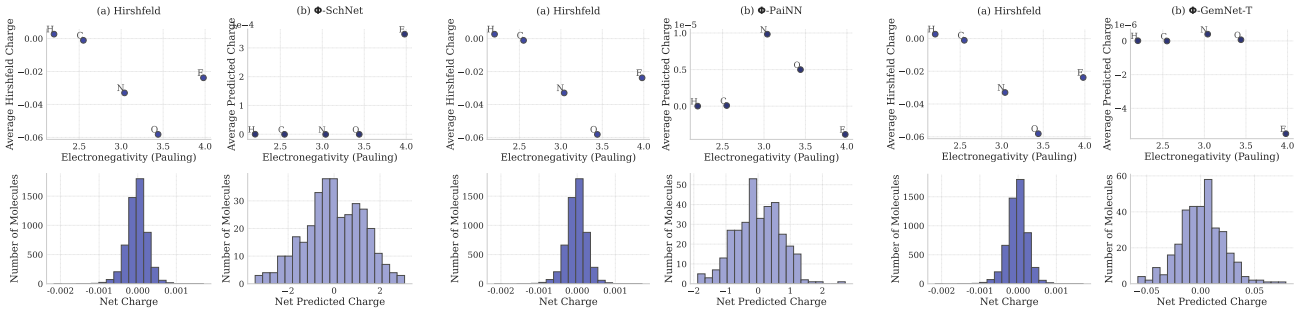


(a): Memory scaling of Φ-Module with varying k.



(b): GPU runtime comparison of Φ-Module with Ewald and Neural P3M.

Figure 8: Additional memory and runtime scaling experiments.



(a): **Φ**-SchNet. (b): **Φ**-PaiNN. (c): **Φ**-GemNet-T.

Figure 9: Charge distribution analysis for the main models on OE62. See Appendix D.

E. Prior on Expected Validation Performance and Additional Experiments

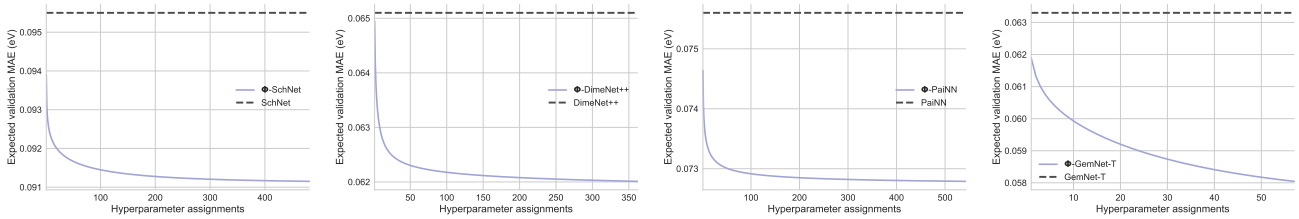
Expected Validation Performance (EVP) Dodge et al. (2019) curve represents how on average performance changes as the number of hyperparameter assignments increases during the random search. The X-axis represents the number of hyperparameter trials. The Y-axis represents the expected best performance for a given number of hyperparameter trials.

The expected best performance is computed as

$$\mathbb{E}[V_n^*|n] = \sum_v v \cdot (P(V_i \leq v)^n - P(V_i < v)^n),$$

where $V_n^* = \max_{i \in \{1,\ldots,n\}} V_i$ is the maximum for model performance evaluations V_i given a series of n i.i.d. hyperparameter configurations, which are acquired empirically from the random hyperparameter search process and $P(V_n^*|n)$ is the probability mass function for the max-random variable.

The EVP curves for **Φ**-SchNet, **Φ**-DimeNet++, **Φ**-PaiNN and **Φ**-GemNet-T can be seen in Figure 10. Φ-Module demonstrates hyperparameter stability for all of the baseline models.



(a): **Φ**-SchNet (b): **Φ**-DimeNet++ (c): **Φ**-PaiNN (d): **Φ**-GemNet-T

Figure 10: EVP curves for the main models on OE62. Each plot demonstrates that any selection of hyperparameters for Φ-Module achieves better performance than the baselines. See Section 4 and Appendix E.

Table 3: Φ-Module hyperparameters for the reported models on OE62.

Model	Hyperparameters			
	k	β	γ	n_{warm}
Φ-SchNet	9	10^{-4}	10^{-4}	10
Φ-DimeNet++	9	10^{-2}	10^{-4}	0
Φ-PaiNN	10	10^{-3}	10^{-1}	0
Φ-GemNet-T	3	$5 * 10^{-1}$	10^{-3}	3
Φ-E_2GNN	5	10^{-1}	0	0

F. Hyperparameters

Hyperparameter Search. We run a hyperparameter search with random uniform smapling for the Φ-Module with the following configuration for each model in this study: k: $\{3, 5, 7, 9, 10, 15\}$, β: $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 5 * 10^{-1}\}$, γ: $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 5 * 10^{-1}\}$, n_{warm}: $\{0, 3, 10, 25\}$. All experiments were conducted on one NVIDIA 80G H100 GPU.

OE62. We follow the same hyperparameters for the baselines as in Kosmala et al. (2023) in most cases. The main difference are the usage of Adam (Kingma & Ba, 2014) optimizer and cosine learning rate schedule (Loshchilov & Hutter, 2016) without warm restarts as well as gradient clipping of 10^3. For hyperparameters related to Φ-Module, refer to Table 3. For E_2GNN we use 256 hidden channels, 6 layers, 128 Gaussian RBFs, cutoff of 6.0 Å with a maximum 50 neighbors. Batch size is set to 64 and the training of E_2GNN was performed for 400 epochs with the same optimizer and scheduler as for other models without gradient clipping.

MD22. For the baseline ViSNet, we employ the same hyperparameters as in (Wang et al., 2022). Optimizer and schduler choice is the same as for OE62. The hyperparameters for **Φ**-ViSNet are the following: k - 9, β - 10^{-3}, γ - 10^{-4}, n_{warm} - 0.

G. Proof of Theorem 3.1

Proof. **1. Exact Consistency**

Spectral Projection (SP): By construction:

$$\phi_{\text{SP}} = U_k \alpha$$
$$\rho_{\text{SP}} = U_k \Lambda_k \alpha$$

Substituting into the Poisson equation:

$$\begin{aligned} L\phi_{\text{SP}} &= U\Lambda U^\top U_k \alpha \\ &= U \begin{bmatrix} \Lambda_k \\ 0 \end{bmatrix} \alpha \quad (\text{since } U^\top U_k = \begin{bmatrix} I_k \\ 0 \end{bmatrix}) \\ &= U_k \Lambda_k \alpha = \rho_{\text{SP}} \end{aligned}$$

Thus $\|L\phi_{\text{SP}} - \rho_{\text{SP}}\|_2 = 0$ by design, requiring only k parameters.

Direct Learning (DL): Independent parameterization of $\phi_{\text{DL}} \in \mathbb{R}^n$ and $\rho_{\text{DL}} \in \mathbb{R}^n$ requires $2n$ parameters. Even if $\rho_{\text{DL}} = L\phi_{\text{DL}}$, the solution space remains unconstrained, with no guarantee of minimal parameterization or residual minimization.

2. Noise Robustness

Let $\rho = U_k \Lambda_k \alpha + \epsilon$ with noise $\epsilon \in \mathbb{R}^n$.

Spectral Projection (SP):

$$\begin{aligned} \|L\phi_{\text{SP}} - \rho\|_2 &= \|U_k \Lambda_k \alpha - (U_k \Lambda_k \alpha + \epsilon)\|_2 \\ &= \|\epsilon\|_2 \end{aligned}$$

Direct Learning (DL): The DL solution $\phi_{\text{DL}} = L^\dagger \rho$ gives:

$$\begin{aligned} \phi_{\text{DL}} &= U\Lambda^{-1}U^\top (U_k \Lambda_k \alpha + \epsilon) \\ &= U_k \alpha + U\Lambda^{-1}U^\top \epsilon \end{aligned}$$

The approximation error satisfies:

$$\begin{aligned} \|\phi_{\text{DL}} - \phi_{\text{SP}}\|_2 &= \|U\Lambda^{-1}U^\top \epsilon\|_2 \\ &= \|\Lambda^{-1}U^\top \epsilon\|_2 \\ &\leq \frac{1}{\lambda_n} \|\epsilon\|_2 \end{aligned}$$

where the inequality follows from $\|\Lambda^{-1}\|_2 = 1/\lambda_n$.

□