Accelerated Deep Atomic Potential Transformer for long-range machine learning force fields without graphs

Evan Dramko,[1] Yizhi Zhu,[2,3,4] Yihuang Xiong,[4] Geoffroy Hautier,[2,3,4,*] Thomas Reps,[5] Christopher Jermaine,[1] and Anastasios Kyrillidis[1]

[1]*Department of Computer Science, Rice University, Houston, Texas, USA*
[2]*Department of Materials Science and Nanoengineering, Rice University, Houston, Texas, USA*
[3]*Rice Advanced Materials Institute, Rice University, Houston, Texas, USA*
[4]*Thayer School of Engineering, Dartmouth College, Hanover, New Hampshire, USA*
[5]*Department of Computer Sciences, University of Wisconsin-Madison, Madison, Wisconsin, USA*

(Received 2 March 2026; accepted 21 July 2026; published 3 September 2026)

Point defects play a central role in driving the properties of materials. First-principles methods are widely used to compute defect energetics and structures, including at scale for high-throughput defect databases. However, these methods are computationally expensive, making machine learning force fields (MLFFs) an attractive alternative for accelerating structural relaxations. Most existing MLFFs are based on graph neural networks (GNNs), which can suffer from oversmoothing, oversquashing, and poor representation of long-range interactions. Both of these issues are especially of concern when modeling point defects. To address these challenges, we introduce the Accelerated Deep Atomic Potential Transformer (ADAPT), an MLFF that replaces graph representations with a direct coordinates-in-space formulation and explicitly considers all pairwise atomic interactions. Atoms are treated as "tokens," with a transformer encoder modeling their interactions. Applied to a dataset of silicon point defects, ADAPT achieves an ∼22% reduction in force and an ∼40 reduction in energy prediction error relative to a state-of-the-art GNN-based model, while requiring only a fraction of the computational cost.

DOI: 10.1103/3xf3-ts5f

I. INTRODUCTION

First-principles electronic-structure methods provide accurate predictions of energies and forces, but their substantial computational cost limits the quantity and size of systems that can be practically computed. Machine learning force fields (MLFFs), also referred to as machine learning interatomic potentials (MLIPs), offer an attractive alternative by learning to approximate density functional theory (DFT) from collected datasets. MLFFs can provide predictions several orders of magnitude faster than DFT, enabling high-throughput screening and significantly reducing the cost of structural relaxations. Often, it is used a prerelaxation step, as seen in Fig. 1.

State-of-the-art architectures and continuing research in MLFFs have been dominated by graph-based and equivariant neural network architectures [1,2], which have demonstrated strong performance across bulk materials and molecular benchmarks [3–12]. Graph neural networks (GNNs), being a subclass of geometric machine learning, encode physically derived symmetries into the model directly. This not only ensures that model predictions will satisfy certain known properties, but increases model efficiency on limited datasets [11,13–15]. Recent approaches in GNNs have included the creation of specialized attention mechanisms [6,16] and higher-order message passing [3,8]. These models have also begun to be applied to defect prediction tasks, ranging from formation-energy estimation [17–19] to accelerating first-principles atomic relaxations [20].

Despite this progress, applying GNN-based MLFFs to point-defect systems remains challenging. Inherent to defect computations is a repetition of cells to form a supercell [21,22], in which only the central cell contains the defect. Large supercells are needed to avoid spurious interactions between defect images due to periodic boundary conditions. The resulting structures often contain hundreds to thousands of atoms, dramatically increasing the cost of message passing and exacerbating oversquashing and oversmoothing effects that degrade the propagation of long-range information [23]. An approach that modifies the GNN architecture to place extra emphasis on local neighborhoods [19] was shown to be effective for properties dominated by short-range interactions, such as defect-formation energies. One-hop relaxation strategies have shown promise in two-dimensional materials [24], yet such amortized approaches require extremely large and diverse datasets [25–27], which remains impractical for complex three-dimensional defect configurations.

These limitations motivate architectural alternatives for defect supercells that explicitly model nonlocal interactions.

*Contact author: geoffroy.hautier@rice.edu

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

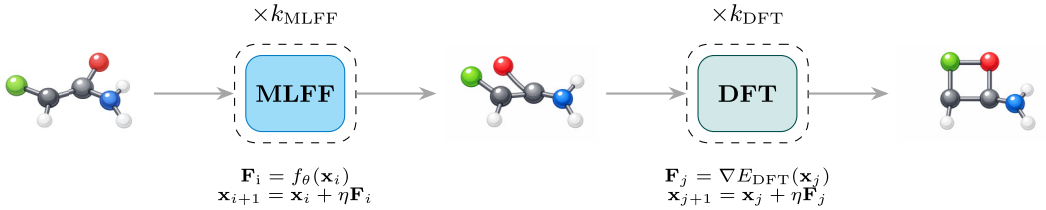


FIG. 1. Common two-stage relaxation workflow: An initial structure is first relaxed for k_{MLFF} steps using an MLFF-driven force update loop. After MLFF convergence, the resulting structure initializes a second relaxation loop of k_{DFT} steps using DFT-computed forces, producing the final relaxed configuration (a) ADAPT (force) Model, (b) Retrained MACE, (c) MACE OMAT-0 Medium, and (d) Pretrained MatterSim 5M.

Inspired by the global receptive fields of transformer models [28], which have demonstrated broad effectiveness in natural language processing [29], computer vision [30], and biomolecular modeling [31], we introduce the Accelerated Deep Atomic Potential Transformer (ADAPT). ADAPT is a coordinate-based transformer that computes full attention over all atomic pairs and predicts per-atom forces directly from Cartesian coordinates and elemental features, thereby circumventing the depth-dependent signal degradation inherent to message-passing architectures.

As access to high-performance computing increases, it is becoming increasingly viable to generate large datasets for supercell defect relaxations. We train ADAPT on a newly released large Perdew-Burke-Erzhenhof (PBE)-level DFT dataset of silicon point defects, including a broad range of complex defect configurations [32,33], as seen in Sec. II B and Appendix A. ADAPT achieves state-of-the-art accuracy in both energies and forces, outperforming universal MLFFs, such as pretrained MACE [3] and MatterSim [5], as well as a MACE model retrained on the same dataset. Notably, ADAPT trains at a cost approximately an order of magnitude lower than comparable graph-based architectures. These results indicate that globally receptive, coordinate-based transformer models provide a promising direction for large supercell structural relaxations of defects.

Since the initial release of this work, other studies have independently explored transformer-based MLFFs without explicit equivariance [34,35], as well as a related line of work developed in parallel and released in stages [36]. While the earliest version of this latter work predates ADAPT, its most recent results appeared after our initial release and overlap only partially with our approach. In particular, it relies on symmetry-aware data representations rather than architectural equivariance, whereas ADAPT enforces no symmetry beyond those implicit in the training data. Collectively, these results demonstrate that nonequivariant transformer-based MLFFs can achieve state-of-the-art accuracy on standard molecular benchmarks.

II. RESULTS

A. MLFFs without energy conservation

In contrast to MACE [3], MatterSim [5], and related architectures, ADAPT employs separate neural networks for force and energy prediction. While ADAPT's force and energy models each outperform state-of-the-art baselines on the target benchmarks, this design breaks strict energy conservation, since the predicted forces are not constrained to equal the gradient of a scalar potential energy surface. Conservative formulations have historically been viewed as essential for physically faithful atomistic simulation, particularly for long-timescale molecular dynamics (MD) and thermodynamic sampling [3,37]. However, several recent works have questioned whether strict conservation is always necessary for downstream tasks, such as structure relaxation, screening, or surrogate optimization [38–40]. In particular, direct-force or nonconservative models can substantially reduce computational overhead by avoiding backpropagation through atomic coordinates during inference, while also improving force-prediction accuracy in some regimes [36,39,41,42]. Recent analyses demonstrate that in some cases, such approaches may still remain usable in MD settings, where instability and energy drift are mitigated by the high accuracy of the models [36], and others advocate hybrid conservative/nonconservative approaches [39]. Despite these works, ADAPT is not intended as a general-purpose MD force field. Instead, it is explicitly created and verified for rapid structure relaxation and high-throughput defect screening, where the primary objective is accurate convergence to relaxed geometries rather than faithful reproduction of relaxation dynamics.

The energy-only predictor of ADAPT also serves several independent purposes. First, despite the overall ADAPT framework being nonconservative, the energy model can still be paired with the force predictor during downstream optimization procedures such as FIRE, where it demonstrates superior practical relaxation performance compared to a state-of-the-art conservative baseline, as seen in Sec. II E 2. Second, standalone energy prediction models are broadly useful for high-throughput materials discovery workflows, including materials stability estimation, candidate filtering, ranking, and rapid surrogate evaluation prior to expensive first-principles calculations [38,43,44].

B. Architecture preliminaries

Both the proposed force- and energy-predictor architectures eschew graphs and inductive biases entirely, instead focusing on precise representations of geometries. Our primary aim is to develop force and energy predictors tailored for defect computations, with the longer-term objective of reducing reliance on costly DFT steps in relaxations.

ADAPT force and energy architectures adopt the now standard tokenization paradigm [45] from transformer-based deep learning [28] of breaking inputs into sequences of tokens.

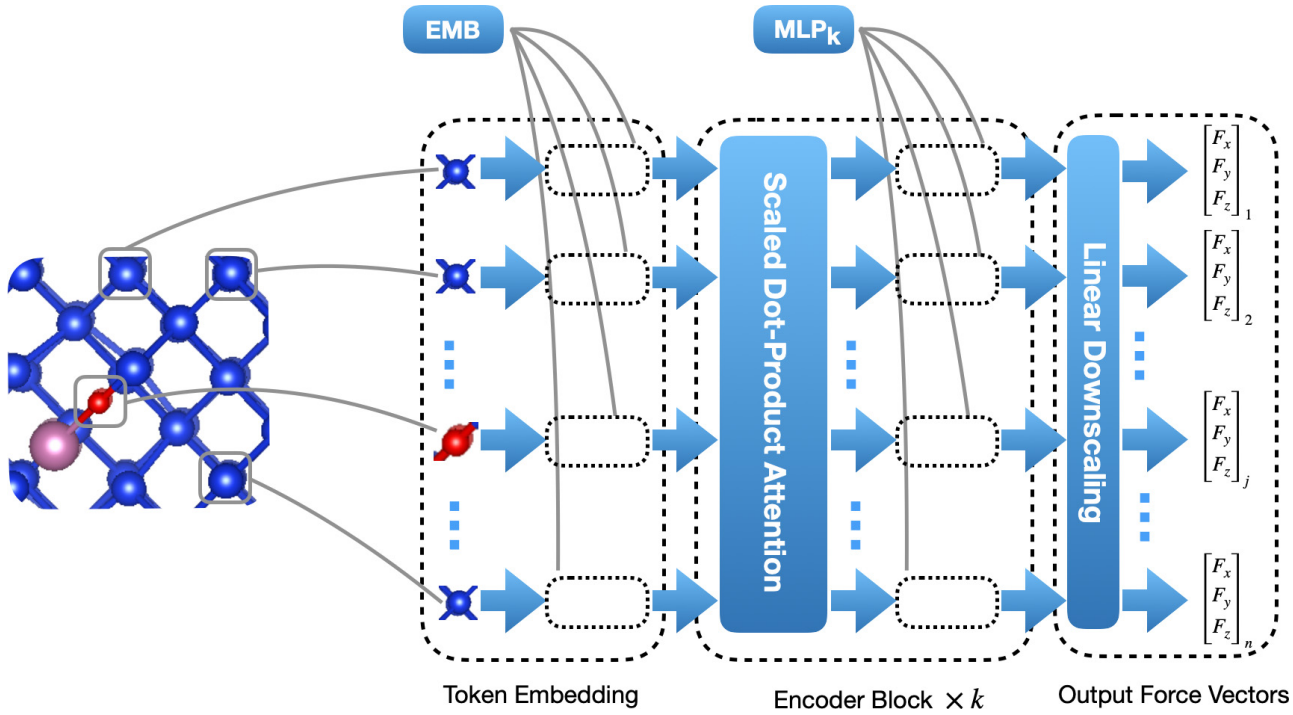


FIG. 2. ADAPT MLFF force-prediction architecture. Following the transformer encoder paradigm, each of n atoms is embedded via an identical multilayer perceptron denoted EMB, passed through a stack of attention-based encoder blocks, and linearly projected from $(n \times d_{\text{model}})$ to $(n \times 3)$ force vectors.

Here, each token corresponds to a single atom, so a structure with n atoms is represented by n tokens. Initially present in the dataset is a 12-dimensional vector representing each atom by

$$(x, y, z, \text{column}, \text{row}, \chi, r_{\text{cov}}, N_{\text{val}}, E_{\text{ion1}}, E_{\text{EA}}, r_{\text{atom}}, V_{\text{mol}}),$$

where we define x, y, z as the coordinates of the atom, column is the atom's group, row is the atom's period, χ is the electronegativity, r_{cov} is the covalent radius, N_{val} is the number of valence electrons, E_{ion_1} is the first ionization energy of the atom, E_{EA} is the electron affinity, r_{atom} is the atomic radius, and V_{mol} is the molar volume. Experiments in Appendix F show that there is a negligible difference between using the set of 12 features and the common approach of using just the coordinates and atomic number [3,4,8].

ADAPT has been designed to predict the forces and energy for structures that are simulated on computations in a supercell. We consider defect in silicon as our motivating example. DFT calculations were carried out using PBE functional [46] with a 216 atom-site supercell with projector augmented wave (PAW) pseudopotential [32,33]. The dataset is composed of roughly 6000 defect configurations in neutral charge state. Further details on the dataset creation are available in Appendix A.

C. Force-prediction methodology

Herein, we consider the model architecture used to predict per-atom force vectors, as shown in Fig. 2. It can be viewed as a function mapping each token to a corresponding force vector.

Embedding. Rather than working in the native 12-dimensional space, we embed each token into a higher-dimensional space of size d_{model} (a user-set hyperparameter). High-dimensional representations allow neural networks to transform complex nonlinear dynamics into latent spaces in which linear or low-complexity nonlinear transformations are sufficient to approximate the underlying oracle function, defined as the true generative process that produces the observed real-world data.

A multilayer perceptron (MLP) [47] is used to learn the embedding transformation, and can be represented as

$$\texttt{MLP}(\mathbf{x}) = \mathbf{W}_k\sigma(\mathbf{W}_{k-1}\sigma(\cdots\sigma(\mathbf{W}_0\mathbf{x}+\mathbf{b}_0)\cdots)+\mathbf{b}_{k-1})+\mathbf{b}_k, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^{12}$ is the input token, σ is the elementwise ReLU operation defined by $\text{ReLU}(x) = \max(0, x)$, $\mathbf{b}_j \in \mathbb{R}^{d_{\text{out},j}}$ are the trainable bias terms, and $\mathbf{W}_j \in \mathbb{R}^{d_{\text{out},j} \times d_{\text{in},j}}$ are learnable weight matrices. Here, $d_{\text{in},0} = 12$ and $\mathbf{W}_k \in \mathbb{R}^{d_{\text{model}} \times d_{\text{out},k-1}}$. The embedding MLP is applied independently to each token.

1. Transformer encoder

The embedded sequence is processed by k encoder blocks. Each block has the same structure but distinct parameters. A block is defined by

$$\mathbf{H}_1 = \texttt{LN}(\mathbf{X}_{\text{in}} + \texttt{Attn}(\mathbf{X}_{\text{in}})), \tag{2}$$

$$\mathbf{H}_2 = \texttt{FFN}(\mathbf{H}_1), \tag{3}$$

$$\mathbf{X}_{\text{out}} = \texttt{LN}(\mathbf{H}_2 + \mathbf{H}_1). \tag{4}$$

The main components are as follows:

(a) Layer normalization (LN). This is used to ensure numeric stability in training, and prevent the chaining together of multiplied terms from growing or shrinking rapidly. Given an input $\mathbf{x} \in \mathbb{R}^H$, layer norm normalizes across feature channels:

$$\mu = \frac{1}{H}\sum_{i=1}^{H} x_i, \quad \sigma^2 = \frac{1}{H}\sum_{i=1}^{H}(x_i - \mu)^2, \tag{5}$$

$$\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}}, \quad y_i = \gamma_i \hat{x}_i + \beta_i, \quad i = 1, \ldots, H, \tag{6}$$

where $\boldsymbol{\gamma}, \boldsymbol{\beta} \in \mathbb{R}^H$ are learnable parameters and ϵ is a small constant for stability.

(b) (Multiheaded) scaled dot-produce attention (Attn). In the model, this is the only place where the tokens interact and influence each other. In multiheaded attention, each "head" performs an attention operation over a subset of the data. Given $\mathbf{X} \in \mathbb{R}^{n \times d_{\text{model}}}$ (sequence length n), each head $i = 1, \ldots, h$ is defined by

$$\text{head}_i = \text{softmax}\left(\frac{\mathbf{Q}_i \mathbf{K}_i^\mathsf{T}}{\sqrt{d_k}}\right)\mathbf{V}_i, \tag{7}$$

where

$$\mathbf{Q}_i = \mathbf{X}\mathbf{W}_{\mathbf{Q}_i}, \quad \mathbf{K}_i = \mathbf{X}\mathbf{W}_{\mathbf{K}_i}, \quad \mathbf{V}_i = \mathbf{X}\mathbf{W}_{\mathbf{V}_i}, \tag{8}$$

with projection matrices $\mathbf{W}_{\mathbf{Q}_i}, \mathbf{W}_{\mathbf{K}_i}, \mathbf{W}_{\mathbf{V}_i} \in \mathbb{R}^{d_{\text{model}} \times d_k}$. The raw similarity matrix $\mathbf{Q}_i \mathbf{K}_i^\mathsf{T} \in \mathbb{R}^{n \times n}$ encodes pairwise token similarities. The row-wise softmax, where $\text{softmax}(\mathbf{z}_i) = \frac{e^{\mathbf{z}_i}}{\sum_{j=1}^{n} e^{\mathbf{z}_j}}$, maps each row into a probability distribution over tokens.

Outputs from all heads are concatenated and projected:

$$\texttt{Attn}(\mathbf{X}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\mathbf{W}_O, \tag{9}$$

with $\mathbf{W}_O \in \mathbb{R}^{hd_k \times d_{\text{model}}}$.

(c) Feed-forward network (FFN). FFNs are applied to individual tokens independently, and do not allow any interactions between tokens. They allow for expressive transformations of the token beyond what attention alone can capture. They are given by a positionwise MLP, applied identically to each token:

$$\texttt{FFN}(\mathbf{H}) = \mathbf{W}_2\,\text{ReLU}(\mathbf{W}_1\mathbf{H}^\mathsf{T} + \mathbf{b}_1) + \mathbf{b}_2, \tag{10}$$

where

$$\mathbf{H} \in \mathbb{R}^{n \times d}, \quad \mathbf{W}_1 \in \mathbb{R}^{d_{\text{ff}} \times d}, \quad \mathbf{W}_2 \in \mathbb{R}^{d \times d_{\text{ff}}},$$

$$\mathbf{b}_1 \in \mathbb{R}^{d_{\text{ff}}}, \quad \mathbf{b}_2 \in \mathbb{R}^{d}.$$

(d) Dropout. Dropout randomly masks neuron activations (set to 0), resampled at each pass during training. This has been shown to prevent models from overfitting to the data, and improve generalizability [48]. It is applied to the outputs of attention and feed-forward layers. Following convention, we exclude it from the equations for the model definition since it is only used during training and not inference.

Force projection. Finally, after the encoder blocks, forces are obtained by a linear projection:

$$\widehat{\mathbf{y}} = \mathbf{X}_{\text{enc}}\mathbf{W}_{\text{out}}, \quad \mathbf{W}_{\text{out}} \in \mathbb{R}^{d_{\text{model}} \times 3}, \tag{11}$$

producing per-token force vectors (f_x, f_y, f_z). The resulting tensor has shape $n \times 3$.

2. *Defect-aware loss function*

Training requires a differentiable objective that captures the mismatch between predicted and true atomic forces. A natural baseline is the mean-squared error (MSE). Plain MSE, however, does not bias toward any one atom implicitly, even though domain knowledge tells us that atoms nearest the defects dominate the crystal's mechanical response.

To emphasize these critical regions, we introduce a new loss function: importance-weighted MSE. In particular, we create an importance mask $\mathbf{m} \in \mathbb{R}_+^n$, where each of the n atoms, a_i, receives weight:

$$m_i = \prod_{j \in \mathcal{D}}\left(1 + \frac{\lambda_1}{\|\mathbf{r}_i - \mathbf{r}_j\|^2 + \lambda_2}\right), \quad \mathcal{D} = \{\text{defects}\}, \tag{12}$$

where $\mathbf{r}_i$ is the coordinate vector for atom i and $\mathcal{D}$ is the set of defect locations. The formulation used herein does not consider vacancies, but could easily be modified to do so if necessary. This is similar to laws observed in nature, where the effect of many interactions decay as a power law of the distance between them [49]. Hyperparameters λ_1, λ_2 are used to ensure numerical stability and to "temper" the scaling. The resulting loss becomes

$$\mathcal{L}(\widehat{\mathbf{y}}, \mathbf{y}) = \sum_i m_i \sum_j (\widehat{y}_{i,j} - y_{i,j})^2,$$

where $\mathbf{y}, \widehat{\mathbf{y}}$ are the actual and predicted forces for each of the atoms (indexed i) and across each of the three components of the force vectors (indexed j). Comparisons of the effect of the custom loss weighting component against three trials of standard MSE loss show the custom loss weighting version achieves roughly 12.5% less error on average than regular MSE. While this loss function is designed specifically for defects, other weighting rules can be designed to match other datasets and training scenarios.

D. Energy prediction methodology

We train a separate energy-predictor model to complement the force-prediction ADAPT MLFF. For this task, we consider three distinct architectures: (1) a specialized decoder (see Appendix D for architectural details), (2) a MLP (see Sec. II C), and (3) an MLP + residual network. In each case, the model receives only the atomic structure and returns an estimated crystal energy. Architectures (1) and (2) serve as natural baselines; the decoder as a single output is the natural extension of the encoder framework, and the MLP is a widely used approach [50–52]. Architecture (3), however, substantially outperforms both, and we adopt it as our primary design.

1. MLP + *residual architecture*

Residuals connections, where the input and output of a layer are added together, have become widespread in ML literature. It has been noted that the residual architecture bears striking resemblance to Euler integration [53,54], making it a common choice [55–57] when considering modeling physical

TABLE I. Selection performance.

Model type	Total $\mathcal{L}_2$ error
Decoder	23.5508
MLP only	50.3728
MLP + residual	11.1683

systems that are governed by differential equations. The architecture of a MLP with residual connections for raw input tokens $\mathbf{x}$ is

$$\begin{aligned}\mathbf{t}_0 &= \sigma(\mathbf{W}_0\mathbf{x} + \mathbf{b}_0),\\ \mathbf{h}_0 &= \mathtt{LN}(\mathbf{P}_0\mathbf{t}_0 + \mathbf{t}_0),\\ \mathbf{t}_1 &= \sigma(\mathbf{W}_1\mathbf{h}_0 + \mathbf{b}_1),\\ \mathbf{h}_1 &= \mathtt{LN}(\mathbf{P}_1\mathbf{t}_1 + \mathbf{t}_1),\\ &\vdots\\ \widehat{\mathbf{y}} &= \mathbf{W}_k\mathbf{h}_k + \mathbf{b}_k,\end{aligned}$$

where $\mathbf{W}_i$, $\mathbf{P}_i$, $\mathbf{b}_i$ are learnable weight matrices/vectors of any mathematically valid dimensions. However, if $\mathtt{Dim}(\mathbf{h}_{i-1}) = \mathtt{Dim}(\mathbf{h}_i)$, then $\mathbf{P} = \mathbf{I}$. Dropout (see Sec. II C 1) is applied after each ReLU activation function σ, and all other notation matches that used in Sec. II C 1. Unlike transformers, MLPs and MLP + residuals require fixed-length inputs. Based on the structures present in our data, we pad, i.e., add dummy atoms where all values are 0, to every structure to 220 atoms before feeding it to the network. The selection of 220 atoms stems from the regular Si lattice box in the dataset having $6^3 = 216$ atoms, and allowance for the inclusion of point defects. For larger systems, the energy-predictor model can be retrained or fine-tuned with a higher maximum length rather than truncating atoms.

2. Initial model selection

To quantify performance, we train each candidate for 200 epochs and evaluate the total loss on the test set. The results are shown in Table I. The MLP + residual achieves the lowest error, justifying its selection as the recommended architecture. After adopting it, we further refine the model with an additional 200 epochs of training until convergence was observed.

E. Numerical results

The primary criterion for comparing MLFFs is accuracy in force and energy prediction, typically measured by $\mathcal{L}_2$ or MAE. We benchmark ADAPT against two state-of-the-art models: MACE [3] and MatterSim [5]. To ensure comparability, we train both MACE and ADAPT from scratch on a dataset of 6082 charge neutral silicon defect DFT trajectories from our previous works, which contains both simple and complex defects with a total of 56 elements [32,33], and select a set of complex defects as the testing scenario. Details of DFT calculations are provided in Appendix A. We additionally report results from previously benchmarked MACE models [58]. For MatterSim, which is positioned as a large-scale foundation model, retraining is computationally prohibitive; we therefore evaluate using its publicly released checkpoints. All models are evaluated on 100 structures whose trajectories were excluded from training. Performance is assessed in terms of both force-prediction accuracy and the ability of ADAPT to recover correct final relaxed structures. For force evaluation, only the initial configurations from each trajectory are considered, as they contain a substantial fraction of non-negligible force components. As structures approach equilibrium, an increasing number of forces decay toward zero, causing $\mathcal{L}_2$-based error metrics to favor trivial or uninformative predictions. This bias arises because most atoms in the bulk lattice undergo negligible displacement, enabling a model to minimize error by globally suppressing atomic motion while failing to capture the small but physically critical displacements that govern structural evolution. Restricting evaluation to initial configurations mitigates this effect by emphasizing larger force magnitudes. The stability of ADAPT throughout the entire relaxation process is demonstrated by its ability to produce complete and accurate structural relaxations, underscoring its practical utility within atomistic relaxation pipelines.

TABLE II. Force-prediction comparison with SOTA on 100 test structures.

Architecture	Force MAE (eV/Å)
ADAPT small	0.0126
MACE retrained	0.0161
MACE MP0a large	0.0439
MACE MPA-0 medium	0.0349
MACE OMAT-0 medium	0.0283
MatterSim 1M	0.0323
MatterSim 5M	0.0335

1. Prediction accuracy results

Force predictions. Force-prediction results summarized in Table II show ADAPT achieves a 25% error reduction relative to retrained MACE and far outperforms the strongest pretrained model. Parity plots of force and energy errors across the selected test set are shown in Fig. 4, and examples visualizing the effect on selected structures are included in Fig. 3. The accuracy in forces obtained with ADAPT is around 0.01 eV/Å as MAE. This is in the order of magnitude of the stopping criteria for many atomic relaxation within DFT including in our dataset. This indicates that ADAPT could be a good surrogate to DFT relaxation and at least provide useful prerelaxation.

Energy predictions. We also show that the ADAPT defect-formation-energy-predictor model produces performance superior to both MACE and MatterSim. A table of results is given as Table III, and parity plots showing the results are given in Fig. 5. We achieve near identical error to MatterSim 5M, the best of the existing energy predictors, after 200 epochs, and reach our final result, a more than 30% reduction in MAE over MatterSim 5M, after 400 epochs.

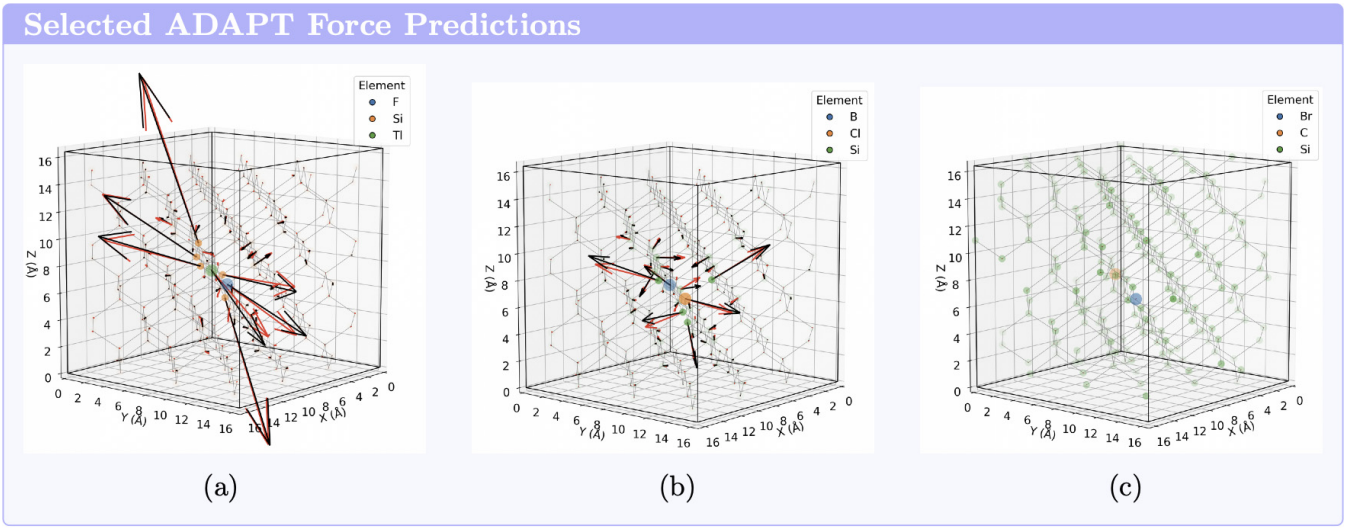


FIG. 3. Side-by-side comparison of instructive outputs. Top row: ADAPT. Bottom row: MACE retrained on the data used to train ADAPT. Predicted forces are shown in black, and actual forces are shown in red. (a) Test Sample 91, (b) Test Sample 80, (c) Test Sample 51, (d) Test Sample 91, (e) Test Sample 80, and (f) Test Sample 51.

2. Relaxation results

Recent studies have demonstrated that accuracy in force prediction alone is insufficient to assess the practical utility of a machine learning force field [40,59]. We show that, when evaluated on the downstream task of structural relaxation, ADAPT's improvements in force and energy predictions translate into more accurate relaxed geometries. Structural relaxations are performed by embedding each model within the iterative relaxation loop, defined in Algorithm II E, where predicted forces are used by an optimizer to update atomic positions until convergence. ADAPT and MACE are compared using the common FIRE optimizer [60], where despite having nonconservative force and energy predictions, ADAPT demonstrates empirical success.

Table IV compares ADAPT with one of the retrained versions MACE [61], yielding an approximately 50% reduction

TABLE III. Energy prediction comparison with SOTA on 100 test structures.

Architecture	Energy MAE (eV)
ADAPT	0.5782
MACE retrained	1.0031
MACE MP0a large	6.1012
MACE MPA-0 medium	2.0478
MACE OMAT-0 medium	3.2232
MatterSim 1M	1.7430
MatterSim 5M	0.8289

TABLE IV. Average ΔQ score after relaxation with FIRE optimizer.

Model	Avg. ΔQ
MACE	12.59
ADAPT	6.73

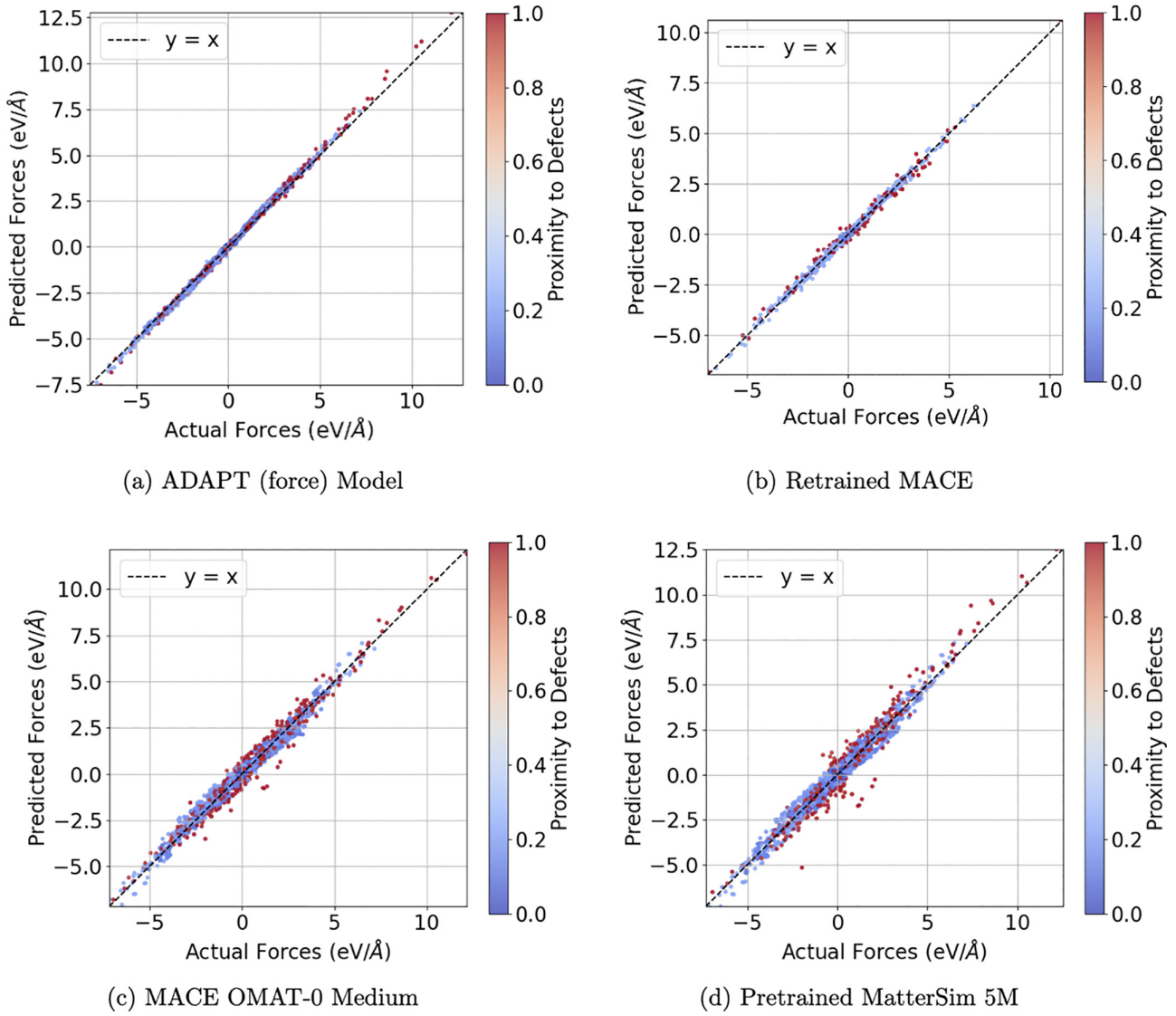


FIG. 4. Parity plots of predicted vs actual forces across test structures. (a) ADAPT (force) Model, (b) Retrained MACE, (c) MACE OMAT-0 Medium, and (d) Pretrained MatterSim 5M.

in the ΔQ score. The ΔQ metric is effectively a mean-squared error weighted by atomic mass [62]. Notably, a further study on relaxations conducted in Ref. [63] shows ADAPT consistently achieves $\Delta Q < 6$ using constant PES-level step sizes, surpassing the performance obtained with the FIRE optimizer. We hypothesize that this improvement arises from the non-conservative nature of employing separate models. Additional details regarding hyperparameter selection and reproducibility are provided in Appendix B.

The success of ADAPT over MACE in relaxations raises an interesting consideration regarding energy conservation; namely, that energy conservation alone does not guarantee correct relaxation dynamics. Rather, energy-conserving models guarantee only that the predicted trajectories remain internally consistent with the learned potential energy surface. In practice, inaccuracies in the learned surface can still produce trajectories that are physically consistent, yet ultimately incorrect. In this sense, all MLIPs introduce some degree of physical error, differing primarily in how those errors are distributed. ADAPT makes the explicit trade-off of reducing overall prediction error at the cost of relaxing the strict coupling between force and energy predictions. Although this sacrifices formal energy conservation, our empirical results suggest that the resulting improvements in predictive accuracy lead to substantially improved structural relaxation performance for the target application considered in this work.

F. Computational efficiency

Force predictions. An advantage of the ADAPT architecture is its computational efficiency. Table VI shows the per-epoch run-time for training ADAPT and MACE. Both are trained on the ∼216 atom dataset of Si defects. For timing comparison, ADAPT is also trained using simulated noise for 512, 1024, and 4096 "atom" dataset sizes. ADAPT computations only depend on the tensor size of the input, not the structure represented, meaning that the processing time for random noise is the same for physical structures. This testing procedure does not carry over to MACE; however, where

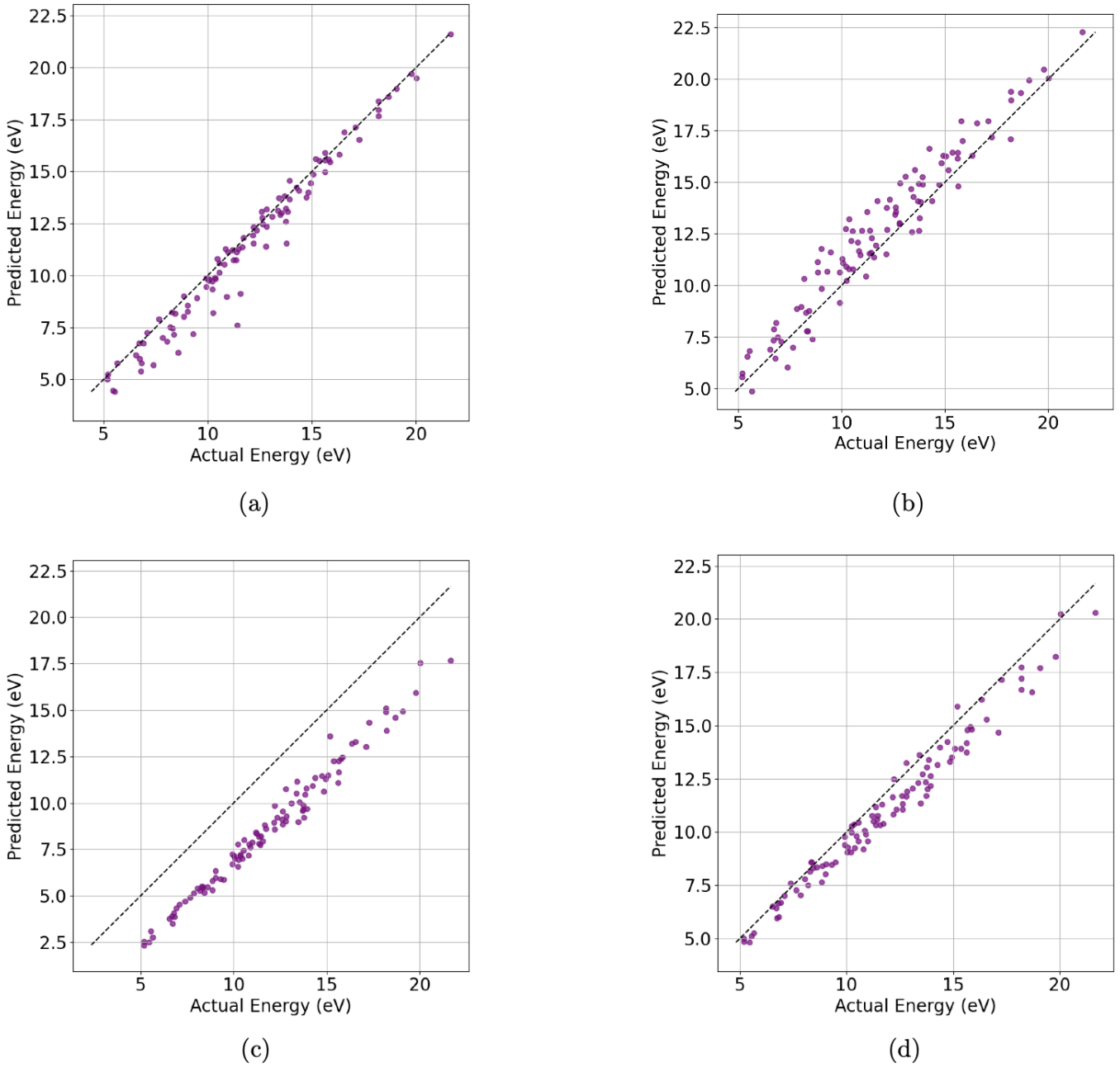


FIG. 5. Parity plots of predicted vs actual defect formation energies across test structures (a) ADAPT (Energy) Model, (b) Retrained MACE, (c) MACE OMAT-0 Medium, and (d) Pretrained MatterSim 5M.

TABLE V. Results of data-augmentation training. All rotations are performed about the centroid rather than the origin (or another fixed point). The reported error is the total $\mathcal{L}_2$ error over the test set. Five trials with random rotations are averaged to compute the augmented error. Training approximately invariant models with data augmentation increases computational cost, but can induce approximate equivariance. Appendix E contains detailed studies of performance under rotation and translation augmentation.

Augmented trainining	Epochs	Unaugmented error	Augmented error
No	80	4.32	121.67
Yes	80	8.92	9.00
Yes	160	6.90	6.74
Yes	320	4.17	3.49

TABLE VI. Per-epoch run-time. Models are trained on NVIDIA A100 GPUs; average minutes per epoch are reported. Note that while increasing the number of GPUs reduces wall-clock training time, parallel efficiency decreases due to communication, synchronization, and load-imbalance overheads inherent to distributed training.

Model	Atoms	A100 GPUs	Min./epoch	Batch size	Compute Min./epoch
MACE	216	16	8.50	8	136.00
MACE	216	1	89.40	8	89.40
ADAPT	216	1	2.24	64	2.24
ADAPT	512	1	7.99	64	7.99
ADAPT	1024	1	25.11	48	25.11
ADAPT	4096	1	341.23	2	341.23

random noise can result in unrealistic numbers of neighbors falling within a cutoff radius.

Training ADAPT converged after 80 epochs (totaling 3 compute hours). In comparison, training MACE took 60 epochs, amounting to 136 compute hours: More than 45× the amount of compute used to train ADAPT's force-prediction model on the same dataset. The compact design of ADAPT permits training on commodity hardware, including workstations and even consumer-grade laptops equipped with graphics processing units (GPUs) [64], thereby significantly reducing hardware requirements for adoption. This accessibility is consistent with the overarching objective of the MLFF literature: to accelerate structural determination by reducing dependence on large-scale computational resources.

These improvements are attributed to the departure from graph-based architectures. Graph neural networks inherently involve sparse operations, which are not easily expressed in the dense linear algebraic form favored by modern accelerators. Consequently, graph-based models typically exhibit lower hardware utilization due to sparse operations, which lack the extensive optimization and backend support available with dense-matrix operations [65]. By forgoing graph representations and adopting architectural paradigms widely developed in natural-language processing and computer vision—where such operations benefit from extensive backend and library support—ADAPT achieves markedly higher computational throughput.

Energy prediction. MACE generates energy predictions concurrently with force predictions within the same forward pass, yielding identical timing characteristics for both quantities. ADAPT trains an additional energy-predictor model, which required 1.93 compute hours on a single NVIDIA A100 GPU. Model training was conducted for 400 epochs, with the duration of a single epoch being 29 s on the same hardware. When including this cost, training both ADAPT models takes a total of 4.92 A100 hours, which is still more than 138× faster than MACE.

III. SYMMETRY HANDLING WITHOUT GRAPHS

A. Effect of rigid motion transformations

Data augmentation. Graph neural networks inherently encode three key symmetries of atomistic systems: (1) permutation invariance with respect to atom ordering, and (2) and (3) equivariance under global rigid translations and rotations [1,2,4–12]. The encoder in ADAPT has permutation invariance, but it does not impose rotational or translational equivariance. A common approach to ensuring that symmetries like rotational and translational equivariance are

ALGORITHM 1. Structural relaxation with an ML force field (MLFF).

Require: Initial structure $\mathbf{R}^{(0)} \in \mathbb{R}^{N\times 3}$ (atomic positions), MLFF model $\mathcal{M}$, optimizer $\mathcal{O}$, force tolerance $\varepsilon > 0$, maximum iterations $K_{\max}$
Ensure: Relaxed structure $\mathbf{R}^*$, final energy E^*
$k \leftarrow 0$
$\mathbf{R} \leftarrow \mathbf{R}^{(0)}$
Initialize optimizer state $\mathcal{S} \leftarrow \mathcal{O}$.init($\mathbf{R}$)
while $k < K_{\max}$ **do**
  $(E, \mathbf{F}) \leftarrow \mathcal{M}(\mathbf{R})$ ▷ E: total energy prediction; $\mathbf{F} \in \mathbb{R}^{N\times 3}$: forces on atoms
  $f_{\max} \leftarrow \max_{i\in\{1,\ldots,N\}} \|\mathbf{F}_i\|_2$
  $f_{\text{total}} \leftarrow \|\mathbf{F}\|_2$
  if $f_{\max} \leqslant \varepsilon$ **then** ▷ With constant PES-level step, $f_{\max}$ is replaced with f_{total}
    break
  end if
  $(\Delta\mathbf{R}, \mathcal{S}) \leftarrow \mathcal{O}$.step($\mathbf{R}, E, \mathbf{F}, \mathcal{S}$) ▷ e.g., FIRE; uses forces as the negative energy gradient
  $\mathbf{R} \leftarrow \mathbf{R} + \Delta\mathbf{R}$
  $k \leftarrow k + 1$
end while
return $\mathbf{R}^* \leftarrow \mathbf{R}$, $E^* \leftarrow E$

respected is to train with data augmentation: in which random rotations and translations of the data are used during training. As shown in Table V and Appendix E, ADAPT can be trained to exhibit a high degree of approximate equivariance to rigid-body transforms: Translating or rotating the input structure leads to little change in the accuracy of predicted energies and forces. This mirrors a very general result in machine learning literature referred to as "the bitter lesson:" With sufficiently large datasets, it is very often more beneficial to "get out of the way" of a machine learning architecture and let it learn itself than to encode restraints directly into the model [66].

B. Scaling laws and relaxation pipeline efficiency

ADAPT employs full self-attention, which incurs a computational complexity of $\mathcal{O}(n^2)$ in the number of atoms n. This scaling is inherent to the formulation: Explicitly modeling long-range, all-to-all interactions requires considering all atomic pairs and cannot be reduced without changing the underlying architecture. In this sense, ADAPT intentionally trades asymptotic complexity for expressive capacity.

In contrast, message-passing GNNs typically exhibit $\mathcal{O}(n)$ arithmetic complexity per layer by restricting interactions to a fixed local neighborhood. However, this theoretical advantage does not directly translate to practical performance on modern hardware. GNNs rely on sequential message-passing steps and sparse, irregular memory access patterns, which limit parallelism and result in poor GPU utilization [67,68]. Attention layers, by comparison, map naturally to dense linear algebra operations and benefit from highly optimized GPU kernels. As a result, despite its worse asymptotic scaling, ADAPT is often faster in practice. As shown in Table VI, ADAPT has a computational burden an order of magnitude less than MACE on ∼217 atom systems. Even when running systems of 512 or 1024 atoms (simulated with the process described in Appendix B), ADAPT uses less compute than MACE does for ∼217 atoms systems.

Benefit compared to DFT. Even in very large structure regimes where the quadratic scaling of attention leads to longer ADAPT versus GNN times, the trade-off may remain favorable. A core premise of MLFF-accelerated workflows is that force-field evaluations are negligible compared to *ab initio* calculations. Standard plane-wave-based Kohn-Sham DFT methods scale as $\mathcal{O}(n^3)$ with system size, and for systems with only $\mathcal{O}(10^2)$ atoms, MLFFs are already orders of magnitude faster than DFT. Consequently, in a situation in which first-principles refinement is utilized following the MLFF relaxation, any nontrivial improvement in MLFF accuracy that reduces the number of required DFT steps yields a net reduction in total computational cost. In this setting, a slightly more expensive MLFF that converges in fewer DFT iterations is strictly preferable to a faster but less accurate model.

IV. DISCUSSION

A. Effect of separate models on efficiency

ADAPT employs separate models for force and energy prediction, a design choice that offers several practical advantages. When only one quantity is required, the corresponding predictor can be deployed independently, reducing both runtime and memory overhead. This distinction is particularly relevant for defect-focused MLFF applications, where simulations are often performed on large supercells containing hundreds of atoms. Such efficiency gains may be important for practitioners operating on local workstations or resource-constrained computing clusters. The separation also improves modularity: Force and energy predictors can be retrained or updated independently, enabling the incorporation of datasets that contain only one target quantity and allowing incremental refinements without retraining the full system.

Modularity also introduces optimization-related limitations. Certain relaxation methods, such as the FIRE optimizer [60], require simultaneous access to both forces and energies. In these settings, we observe degraded relaxation performance relative to optimizers that depend only on force predictions, as discussed in Sec. II E 2. Additionally, joint force-energy models are often more parameter efficient [69], since they learn a shared representation across tasks and can exploit correlations between forces and energies to improve generalization when sufficient training data are available. We note, however, that conclusions regarding internal neural-network representations should be interpreted cautiously, as the black-box nature of these architectures makes their learned dynamics difficult to characterize directly.

Given the limited training data available for silicon defects, it is not surprising [70–72] that a simpler MLP with residual connections outperformed a transformer decoder in this setting (see Table I). Nonetheless, the authors expect that, with sufficient force and energy data, transformer architectures augmented with specialized heads may provide a more scalable and accurate solution. The design of such heads remains an active area of research, and identifying architectures that best balance modularity, efficiency, and accuracy is an open problem.

B. Coordinates versus graphs

GNNs are the default backbone for modern MLFFs [3–6,8], where atoms define nodes, and atomic bonds or proximity determine edge placement. By encoding geometric priors (permutation, rotation, and translation invariance), they incorporate strong inductive biases that improve data efficiency [1,73–75] and have been argued to stabilize relaxation trajectories [16].

Representing continuous atomic interactions using discrete graph topologies introduces mismatches that can limit accuracy, especially in defects where long-range effects and precise geometries are important. GNNs inherently restrict interactions to local regions, relying on network depth to propagate forward information that is outside the interaction radius. This approach often leads to oversmoothing and over-squashing [76,77], where long-range signals degrade rapidly as depth increases. Bulk crystal far from the defect core can substantially shape local defect structures. While long-range influences are less critical in many other chemical systems, neglecting them in crystalline materials can cause large errors. The poor performance of GNNs on large periodic systems—an issue especially relevant in modeling crystalline defects—has been noted [16,78]. Adding long-range interactions into graph architectures [6,16] often leads to significant cost in computation and model complexity. Thus, we arrive at the motivation for using an alternative MLFF strategy for

TABLE VII. Full vs local interaction. The allowed interaction percentage denotes the fraction of all possible atom-atom interactions retained during training and inference. Interactions are controlled in the attention mechanism via key-structural masks (Appendix C). Lower values indicate lower total $\mathcal{L}_2$ error. Data points marked with * converged after 80 epochs, while those marked with † were trained for 200 epochs until convergence.

Allowed interactions (%)	Total $\mathcal{L}_2$ loss
1.46	13.16*
18.7	13.61†
51.3	11.13†
100	8.11*

modeling crystal defects in ADAPT, and a need recognized in Refs. [16,78] as well.

With the advent of transformer architectures and growing datasets, it is now feasible to move away from hard-coded geometric priors and instead focus on explicit representations of global distances and angles. ADAPT employs a transformer encoder (Sec. II C and Appendix B) with full, unmasked self-attention, enabling all-to-all comparisons between atoms at each layer. This approach directly captures nonbonded and long-range interactions without depending on depth-based message passing. Although the model lacks explicit geometric equivariances, permutation invariance is inherent to unmasked attention, and experiments show that translational and rotational invariances can be learned sufficiently well from data. The importance of global attention is underscored in Table VII: Restricting attention to local neighborhoods—as in GNNs—drastically degrades performance.

Accurate representation of geometries. Graphs excel at representing connectivity, but they do not inherently encode precise distances or angular relationships. To address this limitation, many GNN variants augment node and edge features with geometric information [1,3,4,8,16]. However, for long-range interactions, this information is only available indirectly, propagated iteratively from neighbor to neighbor. As path lengths between atoms increase, this process can cause signal degradation due to oversmoothing and oversquashing, with errors compounding over distance.

In contrast, a coordinate-based approach gives direct access to precise pairwise distances and angles for all atoms in a single computation step. This approach not only avoids approximations from multihop propagation, but also preserves geometric detail across all interaction scales.

Limitations and future directions. The ADAPT architecture is not inherently limited to defect relaxation or force prediction. However, it remains an open problem to determine ADAPT's applicability to other problems including diverse bulk structures. Additionally, transformers typically require substantial quantities of data [70–72], making ADAPT unsuitable for tasks with limited training data. The key takeaway of the work, however, is that GNN-free MLFFs can reach high accuracy.

Future directions include (1) enforcing physical symmetries within both the architecture and the loss while remaining distinct from graph-based techniques; (2) extending training beyond silicon to encompass a wider class of defects and materials; and (3) extending the framework to simulate charged defects in semiconductors.

ACKNOWLEDGMENTS

This study was funded by NSF Grants No. CCF-2212558, No. CCF-2212557, and No. CCF 1918651. The first-principles work has been supported by the U.S. Department of Energy, Office of Science, Basic Energy Sciences in Quantum Information Science under Award No. DE-SC0022289. This research used resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231 using NERSC Award No. BES-ERCAP0020966. This research was funded in part by the Robert A. Welch Foundation (Grant No. C-2118, A.K.); Rice University (Faculty Initiative Aaward), NSF CAREER (Award No. 2145629), an Amazon Research Award, and a Microsoft Research Award.

E.D. created the software, designed methodology, curated data. Y.Z. trained the MACE models, curated/sourced data, and provided domain expertise. Y.X. curated/sourced data, and provided domain expertise. C.J., T.R., and T.K. designed methodology. G.H. provided domain expertise. All authors contributed to the writing and revisions of the paper.

The funder played no role in study design, data collection, analysis and interpretation of data, or the writing of this manuscript. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of

TABLE VIII. Training results after 160 epochs with restricted rotations and translations. Reported values are total errors for unaugmented and augmented models.

Degree of rotation	Range of translation	Unaugmented error	Augmented error
0–10	0	2.753	2.726
0–90	0	13.387	3.798
0–360	0	5.573	5.515
0	0–0.1	1.965	1.952
0	0–1	2.863	2.577
0	0–10	2.659	13.788
0–10	0–0.1	2.687	2.740
0–90	0–1	14.210	4.498
0–360	0–10	6.879	13.768

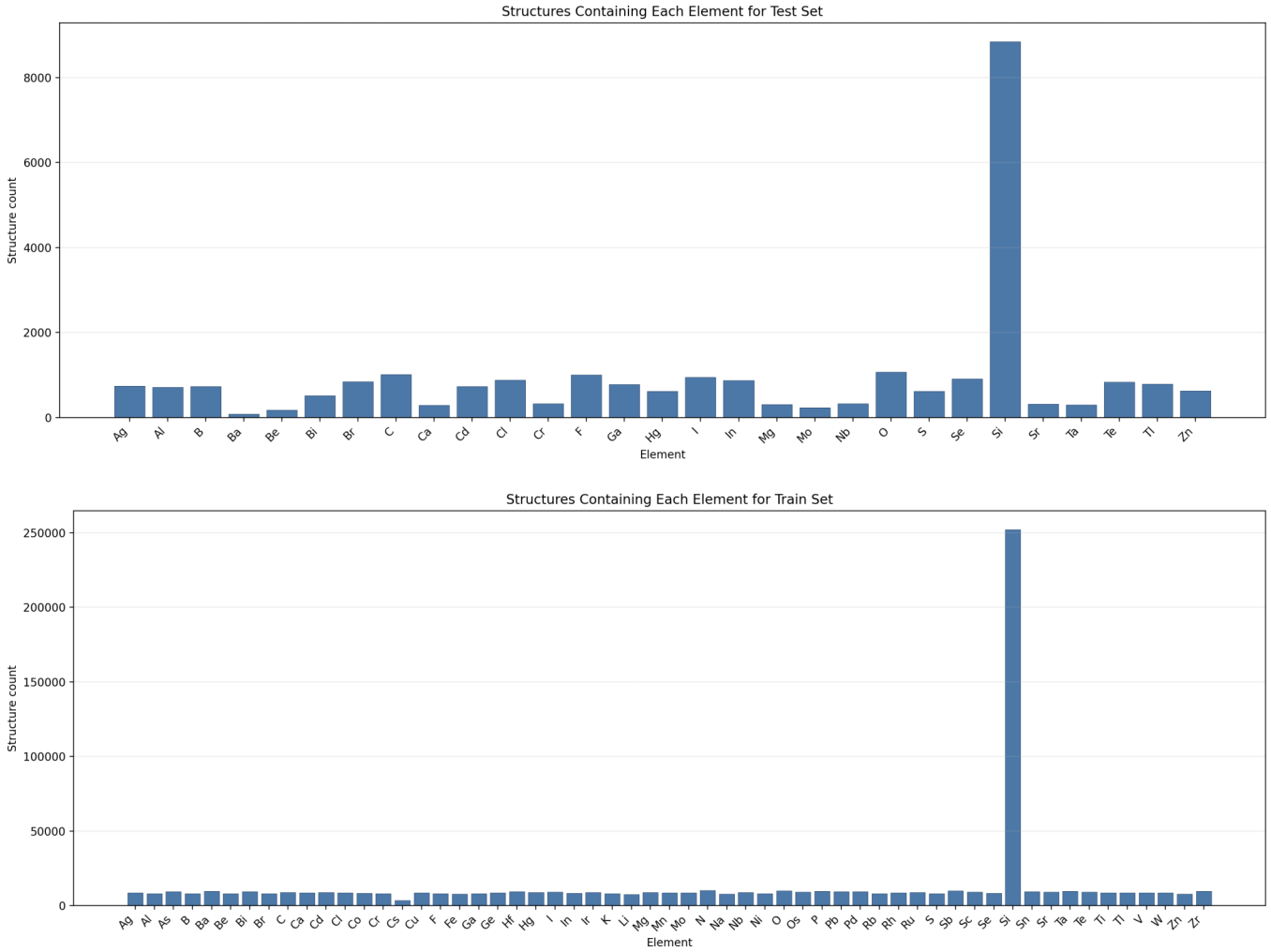


FIG. 6. Elemental frequency rate in training and testing datasets.

the authors, and do not necessarily reflect the views of the sponsoring entities.

DATA AVAILABILITY

The datasets by Xiong *et al.* [32,33] considered during the study are avaible in the "ADAPT_Released" Zenodo project [79]. For more information about the dataset creation and sourcing, see Appendix A. The datasets generated and/or analyzed during the current study are available in the "ADAPT_Released" Zenodo project [79]. The underlying code and training/validation datasets for this study are available in the GitHub repository "ADAPT_Released" and can be accessed at [80].

APPENDIX A: DATASET DETAILS

The DFT trajectories dataset contains both simple and complex defects in silicon, which correspond to previous works [32,33]. The complex defects are in substitutional-interstitial configuration. The defect elements in the dataset span most of the periodic table besides the noble gas, rare-earth elements, and the ones that are difficult to implant, giving in total 56 elements [32]. Charts showing the frequency of the elements in the dataset are included in Fig. 6. In this work, we extract 252 240 number of single-point calculations of neutral charge defects from the relaxation trajectories. The high-throughput defect computations were performed using the automatic workflows that are implemented in atomate software package [81–83]. The first-principles calculations were performed using Vienna *Ab initio* Simulation Package [84,85] with the PAW method [86]. All the calculations were spin polarized at the PBE level [46]. Defect atoms were embedded in a Si supercell with 216 atoms. 520 eV cutoff energies were used for the plane-wave basis, and the Brillouin zone was sampled with single Γ. All the defect structures were optimized at a fixed volume until the ionic forces were smaller than 0.01 eV/Å.

APPENDIX B: ARCHITECTURE DETAILS AND HYPERPARAMETERS

1. Hyperparameters

(1) ADAPT: We define the "small" model size by [$d_{\text{model}} = 256$, $d_{\text{ff}} = 512$, #-layers $= 8$, #-heads $= 8$, dropout

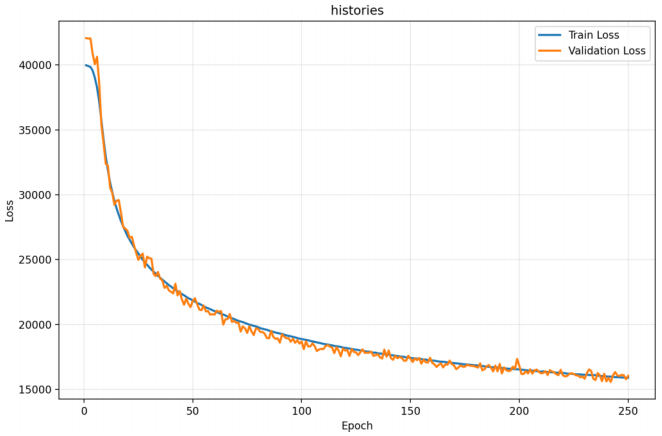


FIG. 7. Loss curves from a sample trial of training the small ADAPT size architecture.

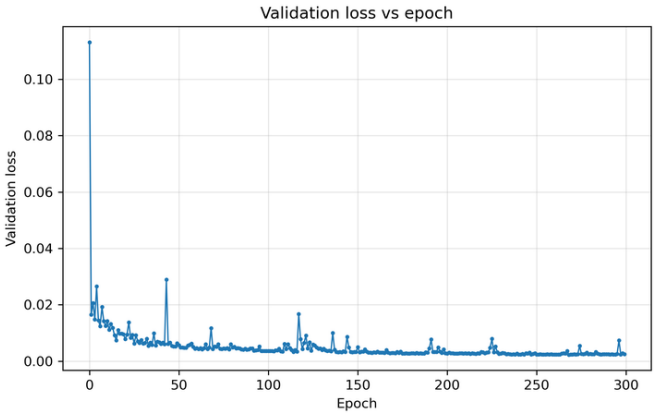


FIG. 8. Loss curve from a sample trial of training the MACE architecture (library only reports validation loss).

rate $= 0.05$] trained for 80 epochs. The "large" model size is [$d_{\text{model}} = 512$, $d_{\text{ff}} = 1024$, #-layers $= 8$, #-heads $= 8$, dropout rate $= 0.05$] trained for 750 epochs. All training was in single precision. The validation set was the first 500 frames, which helps prevent against data leakage through series interpolation [87]. The hyperparameters and training of ADAPT are not necessarily optimal, but are sufficient to demonstrate the key point of the paper that ADAPT-style models can perform competitively with GNNs.

(2) MACE: The retrained version of MACE (v0.3.14, PyTorch 2.6.0) uses num_interactions $= 2$, num_channels $=$ 256, max_L $= 2$, correlation $= 3$, r_max $= 5.0$, trained for 300 epochs on single precision (float32). Using early stopping as well, the weights used in this work result from 60 epochs of training, which is where the loss curve begins to exhibit long-tailed properties.

(3) FIRE optimizer: All parameters are set to the default parameters of FIRE, with $f_{\max} = 0.05$ and $K_{\max} =$ 100 000 000.

(4) Constant step size structural optimizer: Details on fitting procedures, hyperparameter selections, and practicality of results are available in Ref. [63]. Notably, a wide selection of constant step sizes outperform the FIRE based relaxation.

2. Loss curves and training

The small ADAPT configuration achieves an MAE of 0.0120, whereas the large version resulted in an error of 0.0136. Thus, we see that the large configuration exhibited overfitting, indicating that the smaller model already distilled nearly all available information from the data. Accordingly, under the assumption of no additional data or inductive biases, we do not expect training a larger model on the same inputs is likely to achieve a meaningful performance gain. Sample loss curves from ADAPT (small) and MACE training are included in Figs. 7 and 8, respectively.

During ADAPT force-prediction training, early stopping [88,89] was employed due to the high structural similarity among samples and the slow, long-tailed convergence behavior observed in the validation loss. The early-stopping criterion was determined solely from training and validation performance, without reference to the test set. The test set was used only for final evaluation after training and hyperparameter selection were completed. Post hoc evaluation on the held-out test set supports this choice: The full 250-epoch version of ADAPT-small achieves an MAE of 0.0131 eV/Å on the test set, which is 0.005 eV/Å worse than the early stopped model. Furthermore, the degraded performance of ADAPT-large relative to ADAPT-small suggests potential overfitting on the dataset.

3. Data simulation

To simulate large structures of 512, 1024, and 16 384 atoms, we create a dummy dataset of 250 000 structures of (512, 12) to represent the structural input, and (512, 3) to represent the force output. Because there is not cutoff radius in ADAPT, we are able to use random numbers to fill the tensors without introducing issues.

4. Evaluation at different levels

While $\mathcal{L}_2$ error is the conventional standard for comparing force predictions, we find that it is insufficient to fully capture the dynamics of point defects in crystals. To perform a more appropriate comparison, we use two complementary levels. (1) Model level (MLFF): accuracy of force and energy predictions. (2) Relaxation level: quality of the final relaxed structure obtained by running a geometry optimizer with the MLFF.

Model-level evaluation of forces: When comparing candidate models, in addition to the loss scores (see Sec. II C 2), we also consider the average angle and magnitude errors separately. We use the dot product to calculate the angular error in degrees via

$$\text{angle}(\mathbf{y}, \widehat{\mathbf{y}}) = \arccos\left(\frac{\mathbf{y}\cdot\widehat{\mathbf{y}}}{||\mathbf{y}||_2 \cdot ||\widehat{\mathbf{y}}||_2}\right)\cdot\frac{180}{\pi},$$

and we calculate the difference in magnitudes via

$$\text{mag}(\mathbf{y}, \widehat{\mathbf{y}}) = |\|\mathbf{y}\|_2 - \|\widehat{\mathbf{y}}\|_2|.$$

These results help to determine whether the model is genuinely learning the underlying dynamics or artificially minimizing error by predicting uniformly negligible forces—knowing that in reality, most of them will be close to

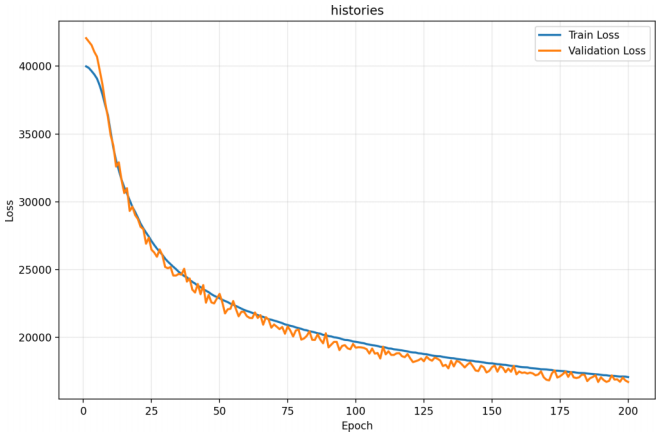


FIG. 9. Loss history from training with atomic number only.

zero. Although this angular-magnitude metric is differentiable and theoretically usable as a loss function for the MLFF, in practice it is difficult to balance the angular and magnitude components effectively. Empirical results show that angular-loss functions are often brittle and require significant engineering effort to implement reliably [90]—a result borne out in our own experiments. In contrast, using a weighted MSE loss is simpler, more robust, and yields strong performance at both the MLFF and structural-relaxation levels, making it the preferred choice.

Evaluation of energy: The total energy of the crystal is represented with a single number, making evaluation very easy. We use the common $\mathcal{L}_2$ distance metric.

Evaluation of tructural elaxation: In order to evaluate the final result of the full relaxation procedure, we use the well known SOAP and delta Q metrics. Other checkers (such as those which check bond lengths) are also viable, although we do not use them in this work.

APPENDIX C: MASKING IN ATTENTION

When restricting interactions in attention, we apply *masks* to the attention logit matrix

$$QK^{\mathsf{T}} \in \mathbb{R}^{B\times H\times T\times T},$$

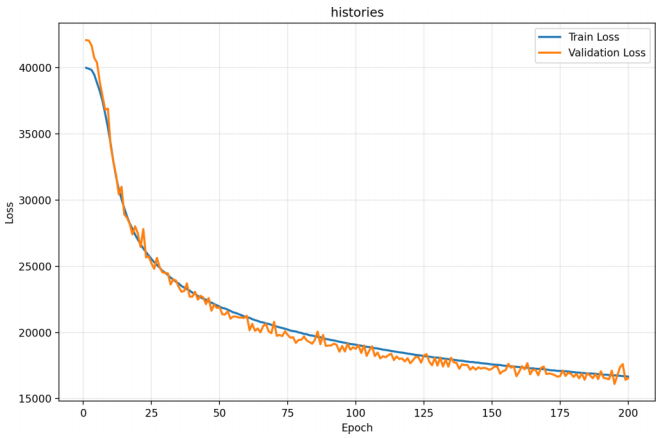


FIG. 10. Loss history from training with set of nine atomic descriptors.

where B is the batch size, H the number of heads, and T the sequence length (number of tokens). Masking is applied along the key dimension (the columns), so that certain tokens cannot be attended to. We use two types of masks:

(1) *Padding mask*. To enable batching, all sequences are padded, which means appending dummy tokens, typically all zeros, to make every sequence the same length. Padding tokens must not affect the model's output, so we mask them out of the attention computation.

(2) *Restricted visibility (local radius)*. To study the effect of limiting each token's visible neighborhood, we compute a restricted attention mask. Allowed interactions are precomputed from the $\mathcal{L}_2$ distances between raw coordinates, and then the same mask is applied to every attention step in the forward pass.

Key masking mechanism. After computing QK^{T}, all disallowed positions are replaced with `-inf`. During the row-wise softmax, these entries become zero, ensuring that they cannot contribute, regardless of the values in V. Consequently, masked tokens never influence the update of valid tokens. Query values at masked positions can be arbitrary "nonsense" numbers (some implementations explicitly zero them out after each attention layer for safety and clarity), but they cannot affect nonpadded tokens.

APPENDIX D: DECODER

The natural extension of using an encoder to predict forces is to use a decoder to predict energy. While the encoder architecture produces a per-token output II C 1, the decoder architecture produces individual outputs, like a scalar crystal energy, using a similar attention/transformer-based architecture. The decoder design we use starts with a stack of encoder layers like in the force-prediction model II C 1, but instead of the final linear down scaling, the stack is followed by a decoder head. This head defines a "dummy" token, $\mathbf{q}$, which is used to allow the calculations to shrink the output to a constant size. This modification requires us to use a slightly different notation; rather than having `Attn` as an function of a single variable, we denote it as a function of three variables. Each is used (in order) to provide the conditioning of one of $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$.

The decoder architecture is formulated as

$$\begin{aligned}\mathbf{M} &= \texttt{encoder}(\mathbf{X}),\\ \mathbf{h}_0 &= \texttt{LN}(\mathbf{q} + \texttt{Attn}(\mathbf{q}, \mathbf{M}, \mathbf{M})),\\ \mathbf{h}_1 &= \texttt{LN}(\mathbf{h}_0 + \texttt{MLP}(\mathbf{h}_0)),\\ \widehat{\mathbf{y}} &= \mathbf{W}\mathbf{h}_1 + \mathbf{b},\end{aligned}$$

where the notation follows that used in Sec. II C 1, and dropout is applied after `Attn` and `MLP`. Recall that $\mathbf{M} \in \mathbb{R}^{n\times d_{\text{model}}}$, and note that $\mathbf{W} \in \mathbb{R}^{\mathbb{1}\times\mathbb{m}}$. Although it is a matrix of shape $\mathbf{q} \in \mathbb{R}^{(1\times d_{\text{model}})}$, we denote it in lowercase vector form to make clear that it has only one nontrivial dimension. We train both the encoder and decoder layers jointly.

APPENDIX E: AUGMENTATION TRAINING EXPERIMENTAL RESULTS

For all experiments included in this section, rotations are performed around the centroid. Each axis has a random

translation applied to it independently. Error is calculated as total $\mathcal{L}_2$ error over the testing set. Five trials with random rotations are averaged to calculate the augmented error.

1. Preprocessing

In some relaxation problems, appropriate preprocessing of the data can eliminate the need for explicit consideration of some symmetries. Structures can be rigidly translated such that they are placed in a predetermined spot, or coordinates are determined relative to the center of mass. Although more dependent on the specifics of the dataset, it is in some cases even viable to define a canonical "correct" orientation and to prerotate all structures to match this configuration standard. In systems with the periodic boundary condition enforced, we can further shift the location of key features like defect centers to a prescribed location. In our target use case, the high-throughput screening of materials defects, candidate structures are all created through the same process by the user. This inherently creates a system in which all inputs share a common alignment by construction.

Throughout many ML fields, issues with invariances occur [91–96]. In order to enable more expressive architectures, robust preprocessing systems are often introduced. These methods seek to "register" inputs into a canonical pose or orientation. Many variations of these methods have been introduced, ranging from simple deterministic shifts and rotations [95,97] to advanced neural networks trained to reorient data [92,93]. This need has been recognized in atomistic calculations as well [96,98]. In fields conceptually related to MLFF literature, these approaches are often considered standard practice [92–94], recognizing the need to free architectures from architecturally costly symmetry constraints.

2. Trade-off reasoning

We emphasize that the lack of architectural enforcement of rotational and translational equivariance is a deliberate limitation of the present formulation and is not intended as a general recommendation for all MLFF architectures. In many settings, explicit equivariance provides a strong inductive bias and can substantially improve data efficiency. However, in the specific context of defect modeling, we find that the additional preprocessing required to approximately respect rigid-motion symmetries is a reasonable tradeoff in exchange for the ability to model long-range interactions without architectural constraints. This suggests a promising research direction in which symmetry handling is partially delegated to preprocessing, allowing architectural capacity to be focused on modeling physically relevant long-range interactions that are otherwise difficult to capture within strictly equivariant formulations.

Alternatively, one may replace absolute coordinates with relative geometric features, following many existing MLFF architectures that operate on pairwise distances and angular information [3–5,8,11]. In principle, this could be integrated into a transformer by constructing and slicing a $3 \times n \times n$ tensor of relative distances and angles for attention. However, such formulations incur substantially higher token representations, and we therefore do not pursue them in this work.

APPENDIX F: ATOMIC DESCRIPTORS

In Figs. 9 and 10, the effect of using the nine atomic descriptors from Sec. II is compared to using just the atomic number to describe atoms. The representations produce extremely similar errors. Finding optimal descriptor sets remains an open problem, and is orthogonal to this work's primary contribution.

[1] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, arXiv:2104.13478.

[2] P. Reiser, M. Neubert, A. Eberhard, L. Torresi, C. Zhou, C. Shao, H. Metni, C. van Hoesel, H. Schopmans, T. Sommer, *et al.*, Graph neural networks for materials science and chemistry, Commun. Mater. **3**, 93 (2022).

[3] I. Batatia, D. P. Kovacs, G. Simm, C. Ortner, and G. Csányi, MACE: Higher order equivariant message passing neural networks for fast and accurate force fields, Adv. Neural Inf. Process. Syst. **35**, 11423 (2022).

[4] B. Deng, P. Zhong, K. Jun, J. Riebesell, K. Han, C. J. Bartel, and G. Ceder, CHGNet as a pretrained universal neural network potential for charge-informed atomistic modelling, Nat. Mach. Intell. **5**, 1031 (2023).

[5] H. Yang, C. Hu, Y. Zhou, X. Liu, Y. Shi, J. Li, G. Li, Z. Chen, S. Chen, C. Zeni, *et al.*, MatterSim: A deep learning atomistic model across elements, temperatures and pressures, arXiv:2405.04967.

[6] J. T. Frank, O. T. Unke, K.-R. Müller, and S. Chmiela, A Euclidean transformer for fast and stable machine learned force fields, Nat. Commun. **15**, 6539 (2024).

[7] I. Poltavsky and A. Tkatchenko, Machine learning force fields: Recent advances and remaining challenges, J. Phys. Chem. Lett. **12**, 6551 (2021).

[8] C. Chen and S. P. Ong, A universal graph deep learning interatomic potential for the periodic table, Nat. Comput. Sci. **2**, 718 (2022).

[9] K. Choudhary and B. DeCost, Atomistic line graph neural network for improved materials property predictions, npj Comput. Mater. **7**, 185 (2021).

[10] K. Schütt, P.-J. Kindermans, H. E. Sauceda Felix, S. Chmiela, A. Tkatchenko, and K.-R. Müller, SchNet: A continuous-filter convolutional neural network for modeling quantum interactions, Adv. Neural Inf. Process. Syst. **30** (2017).

[11] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky, E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials, Nat. Commun. **13**, 2453 (2022).

[12] A. Musaelian, S. Batzner, A. Johansson, L. Sun, C. J. Owen, M. Kornbluth, and B. Kozinsky, Learning local equivariant representations for large-scale atomistic dynamics, Nat. Commun. **14**, 579 (2023).

[13] S. T. Miller, J. F. Lindner, A. Choudhary, S. Sinha, and W. L. Ditto, The scaling of physics-informed machine learning with data and dimensions, Chaos Solitons Fractals: X **5**, 100046 (2020).

[14] F. Hendriks, V. Menkovski, M. Doškář, M. G. Geers, and O. Rokoš, Similarity equivariant graph neural networks for homogenization of metamaterials, Comput. Methods Appl. Mech. Eng. **439**, 117867 (2025).

[15] C. Vignac, A. Loukas, and P. Frossard, Building powerful and equivariant graph neural networks with structural message-passing, Adv. Neural Inf. Process. Syst. **33**, 14143 (2020).

[16] J. T. Frank, O. T. Unke, and K.-R. Müller, So3krates: Equivariant attention for interactions on arbitrary length-scales in molecular systems, Adv. Neural Inf. Proc. Systems **35**, 29400 (2022).

[17] M. H. Rahman, P. Gollapalli, P. Manganaris, S. K. Yadav, G. Pilania, B. DeCost, K. Choudhary, and A. Mannodi-Kanakkithodi, Accelerating defect predictions in semiconductors using graph neural networks, APL Mach. Learn. **2**, 016122 (2024).

[18] X. Xiang, D. Soh, and S. Dunham, Exploration of deep learning models for accelerated defect property predictions and device design of cubic semiconductor crystals, J. Phys. Chem. C **128**, 8821 (2024).

[19] Z. Fang and Q. Yan, Leveraging persistent homology features for accurate defect formation energy predictions via graph neural networks, Chem. Mater. **37**, 1531 (2025).

[20] I. Mosquera-Lois, S. R. Kavanagh, A. M. Ganose, and A. Walsh, Machine-learning structural reconstructions for accelerated point defect calculations, npj Comput. Mater. **10**, 121 (2024).

[21] C. Freysoldt, B. Grabowski, T. Hickel, J. Neugebauer, G. Kresse, A. Janotti, and C. G. Van de Walle, First-principles calculations for point defects in solids, Rev. Mod. Phys. **86**, 253 (2014).

[22] V. Ivády, I. A. Abrikosov, and A. Gali, First principles calculation of spin-related quantities for point defect qubit research, npj Comput. Mater. **4**, 76 (2018).

[23] Q. Li, Z. Han, and X.-M. Wu, Deeper insights into graph convolutional networks for semi-supervised learning, in *Proceedings of the AAAI Conference on Artificial Intelligence*, (2018), Vol. 32.

[24] Z. Yang, X. Liu, X. Zhang, P. Huang, K. S. Novoselov, and L. Shen, Modeling crystal defects using defect informed neural networks, npj Comput. Mater. **11**, 229 (2025).

[25] A. D. Lopez-Rojas and C. A. Cruz-Villar, Neural networks as an approximator for a family of optimization algorithm solutions for online applications, Neural Comput. Appl. **36**, 3125 (2024).

[26] B. Amos, Tutorial on amortized optimization, Found. Trends Mach. Learn. **16**, 592 (2023).

[27] R. Qiu, Z. Sun, and Y. Yang, DIMES: A differentiable meta solver for combinatorial optimization problems, Adv. Neural Inf. Process. Syst. **35**, 25531 (2022).

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. **30** (2017).

[29] C. Zhou, Q. Li, C. Li, J. Yu, Y. Liu, G. Wang, K. Zhang, C. Ji, Q. Yan, L. He, *et al.*, A comprehensive survey on pretrained foundation models: A history from BERT to ChatGPT, Int. J. Mach. Learn. Cybern. 1 (2024).

[30] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, An image is worth 16 × 16 words: Transformers for image recognition at scale, arXiv:2010.11929.

[31] J. Abramson, J. Adler, J. Dunger, R. Evans, T. Green, A. Pritzel, O. Ronneberger, L. Willmore, A. J. Ballard, J. Bambrick, *et al.*, Accurate structure prediction of biomolecular interactions with AlphaFold 3, Nature (London) **630**, 493 (2024).

[32] Y. Xiong, C. Bourgois, N. Sheremetyeva, W. Chen, D. Dahliah, H. Song, J. Zheng, S. M. Griffin, A. Sipahigil, and G. Hautier, High-throughput identification of spin-photon interfaces in silicon, Sci. Adv. **9**, eadh8617 (2023).

[33] Y. Xiong, J. Zheng, S. McBride, X. Zhang, S. M. Griffin, and G. Hautier, Computationally driven discovery of T center-like quantum defects in silicon, J. Am. Chem. Soc. **146**, 30046 (2024).

[34] T. Kreiman, Y. Bai, F. Atieh, E. Weaver, E. Qu, and A. S. Krishnapriyan, Transformers discover molecular structure without graph priors, arXiv:2510.02259.

[35] A. A. Elhag, A. Raja, A. Morehead, S. M. Blau, G. M. Morris, and M. M. Bronstein, Learning inter-atomic potentials without explicit equivariance, arXiv:2510.00027.

[36] M. Eissler, T. Korjakow, S. Ganscha, O. T. Unke, K.-R. Müller, and S. Gugler, How simple can you go? An off-the-shelf transformer approach to molecular dynamics, J. Chem. Phys. **164**, 094308 (2026).

[37] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller, Machine learning of accurate energy-conserving molecular force fields, Sci. Adv. **3**, e1603015 (2017).

[38] D. Zhang, H. Bi, F.-Z. Dai, W. Jiang, X. Liu, L. Zhang, and H. Wang, Pretraining of attention-based deep learning potential model for molecular simulation, npj Comput. Mater. **10**, 94 (2024).

[39] F. Bigi, M. Langer, and M. Ceriotti, The dark side of the forces: Assessing non-conservative force models for atomistic machine learning, in *Proceedings of the 42nd International Conference on Machine Learning*, edited by A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, Proceedings of Machine Learning Research, Vol. 267 (PMLR, 2025), pp. 4384–4414.

[40] X. Fu, Z. Wu, W. Wang, T. Xie, S. Keten, R. Gomez-Bombarelli, and T. Jaakkola, Forces are not enough: Benchmark and critical evaluation for machine learning force fields with molecular simulations, arXiv:2210.07237.

[41] M. Neumann, J. Gin, B. Rhodes, S. Bennett, Z. Li, H. Choubisa, A. Hussey, and J. Godwin, Orb: A fast, scalable neural network potential, arXiv:2410.22570.

[42] B. Rhodes, S. Vandenhaute, V. Šimkus, J. Gin, J. Godwin, T. Duignan, and M. Neumann, Orb-v3: Atomistic simulation at scale, arXiv:2504.06231.

[43] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, SchNet—A deep learning architecture for molecules and materials, J. Chem. Phys. **148**, 241722 (2018).

[44] O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky, K. T. Schutt, A. Tkatchenko, and K.-R. Müller, Machine learning force fields, Chem. Rev. **121**, 10142 (2021).

[45] J. J. Webster and C. Kit, Tokenization as the initial phase in NLP, in *Proceedings of the 14th conference on Computational linguistics (COLING '92)* (Association for Computational Linguistics, Nantes, France, 1992), Vol. 4, pp. 1106–1110.

[46] J. P. Perdew, K. Burke, and M. Ernzerhof, Generalized gradient approximation made simple, Phys. Rev. Lett. **77**, 3865 (1996).

[47] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Control Signal Systems **2**, 303 (1989).

[48] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. **15**, 1929 (2014).

[49] An alternative weighting would be

$$\sum \ln \frac{1}{\|\mathbf{r}_i - \mathbf{r}_j\|^2 + \lambda_2}. \tag{F1}$$

We experimented with Eq. (F1), but found that for silicon defects Eq. (12) gave better results. It is possible that Eq. (F1) would perform better in some applications.

[50] D. Jha, L. Ward, A. Paul, W.-k. Liao, A. Choudhary, C. Wolverton, and A. Agrawal, ElemNet: Deep learning the chemistry of materials from only elemental composition, Sci. Rep. **8**, 17593 (2018).

[51] Y. Liang, M. Chen, Y. Wang, H. Jia, T. Lu, F. Xie, G. Cai, Z. Wang, S. Meng, and M. Liu, A universal model for accurately predicting the formation energy of inorganic compounds, Sci. China Mater. **66**, 343 (2023).

[52] L. Zhang, J. Han, H. Wang, R. Car, and W. E, Deep potential molecular dynamics: A scalable model with the accuracy of quantum mechanics, Phys. Rev. Lett. **120**, 143001 (2018).

[53] J. Müller, On the space-time expressivity of ResNets, arXiv:1910.09599.

[54] J. Baggenstos and D. Salimova, Approximation properties of residual neural networks for Kolmogorov PDEs, arXiv:2111.00215.

[55] M. M. Moghaddam, K. Parand, and S. R. Kheradpisheh, Advanced physics-informed neural network with residuals for solving complex integral equations, Anal. Num. Solutions Nonlinear Equat. **11**, 153 (2026).

[56] A. Noorizadegan, R. Cavoretto, D.-L. Young, and C.-S. Chen, Stable weight updating: A key to reliable PDE solutions using deep learning, Eng. Anal. Boundary Elem. **168**, 105933 (2024).

[57] K. Kashinath, M. Mustafa, A. Albert, J. Wu, C. Jiang, S. Esmaeilzadeh, K. Azizzadenesheli, R. Wang, A. Chattopadhyay, A. Singh *et al.*, Physics-informed machine learning: Case studies for weather and climate modelling, Philos. Trans. R. Soc. A **379**, 20200093 (2021).

[58] I. Batatia, P. Benner, Y. Chiang, A. M. Elena, D. P. Kovács, J. Riebesell, X. R. Advincula, M. Asta, M. Avaylon, W. J. Baldwin *et al.*, A foundation model for atomistic materials chemistry, J. Chem. Phys. **163**, 184110 (2025).

[59] S. Maheshwari, Z. Tang, J. Ock, A. Kolluru, A. B. Farimani, and J. R. Kitchin, Beyond force metrics: Pre-training MLFFs for stable MD simulations, arXiv:2506.14850.

[60] E. Bitzek, P. Koskinen, F. Gähler, M. Moseler, and P. Gumbsch, Structural relaxation made simple, Phys. Rev. Lett. **97**, 170201 (2006).

[61] The relaxation uses a version of MACE, which achieved a MAE of 0.0217, rather than the most recent retraining with error 0.0161. However, the difference in force accuracy is small compared to the difference in relaxation quality.

[62] A. Alkauskas, B. B. Buckley, D. D. Awschalom, and C. G. Van de Walle, First-principles theory of the luminescence lineshape for the triplet transition in diamond NV centres, New J. Phys. **16**, 073026 (2014).

[63] E. Dramko, Y. Zhu, A. Krivokapic, G. Hautier, T. Reps, C. Jermaine, and A. Kyrillidis, On the finetuning of MLIPs through the lens of iterated maps with BPTT, arXiv:2512.01067.

[64] The authors successfully trained ADAPT with 216 atom supercells on a 2023 M2 MacBook Air laptop.

[65] S. Liang, Y. Wang, C. Liu, L. He, H. Li, D. Xu, and X. Li, EnGN: A high-throughput and energy-efficient accelerator for large graph neural networks, IEEE Trans. Comput. **70**, 1511 (2020).

[66] R. Sutton, The bitter lesson, Incomplete Ideas (blog) **13**, 38 (2019).

[67] W. Deng and J. Rao, MEGA: More efficient graph attention for GNNs, in *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS), Jersey City, NJ, USA* (IEEE, 2024), pp. 71–81.

[68] A. Auten, M. Tomei, and R. Kumar, Hardware acceleration of graph neural networks, in *2020 57th ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA* (IEEE, 2020), pp. 1–6.

[69] Y. Zhang and Q. Yang, A survey on multi-task learning, IEEE Trans. Knowl. Data Eng. **34**, 5586 (2021).

[70] Y. Liu, E. Sangineto, W. Bi, N. Sebe, B. Lepri, and M. Nadai, Efficient training of visual transformers with small datasets, Adv. Neural Inf. Process. Syst. **34**, 23818 (2021).

[71] H. Zhu, B. Chen, and C. Yang, Understanding why ViT trains badly on small datasets: An intuitive perspective, arXiv:2302.03751.

[72] Y. Zhang, A. Warstadt, H.-S. Li, and S. R. Bowman, When do you need billions of words of pretraining data? in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, edited by C. Zong, F. Xia, W. Li, and R. Navigli (Association for Computational Linguistics, 2021), pp. 1112–1125.

[73] T. W. Ko and S. P. Ong, Data-efficient construction of high-fidelity graph deep learning interatomic potentials, npj Comput. Mater. **11**, 65 (2025).

[74] J. Kiechle, D. M. Lang, S. M. Fischer, L. Felsner, J. C. Peeken, and J. A. Schnabel, Graph neural networks: A suitable alternative to MLPs in latent 3D medical image classification? in *Graphs in Biomedical Image Analysis* (Springer, Cham, Switzerland, 2025), pp. 12–22.

[75] M. Oliva, S. Banik, J. Josifovski, and A. Knoll, Graph neural networks for relational inductive bias in vision-based deep reinforcement learning of robot control, in *2022 International Joint Conference on Neural Networks (IJCNN)* (IEEE, 2022), pp. 1–9.

[76] J. H. Giraldo, K. Skianis, T. Bouwmans, and F. D. Malliaros, On the trade-off between over-smoothing and over-squashing in deep graph neural networks, in *Proceedings of the 32nd ACM international Conference on Information and Knowledge Management (CIKM '23), Birmingham, United Kingdom* (Association for Computing Machinery, New York, NY, 2023), pp. 566–576.

[77] T. K. Rusch, M. M. Bronstein, and S. Mishra, A survey on oversmoothing in graph neural networks, arXiv:2303.10993.

[78] Q. Yan, S. Kar, S. Chowdhury, and A. Bansil, The case for a defect genome initiative, Adv. Mater. **36**, 2303098 (2024).

[79] E. Dramko, Y. Xiong, Y. Zhu, G. Hautier, T. Reps, C. Jermaine, and A. Kyrillidis, Dataset for ADAPT: Lightweight, long-range machine learning force fields without graphs [Dataset], Zenodo, 2025, https://doi.org/10.5281/zenodo.18962774.

[80] https://github.com/EvanDramko/ADAPT_Released.

[81] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, and G. Ceder, Commentary: The Materials Project: A materials genome approach to accelerating materials innovation, APL Mater. **1**, 011002 (2013).

[82] K. Mathew, J. H. Montoya, A. Faghaninia, S. Dwarakanath, M. Aykol, H. Tang, I. heng Chu, T. Smidt, B. Bocklund, M. Horton, J. Dagdelen, B. Wood, Z.-K. Liu, J. Neaton, S. P. Ong, K. Persson, and A. Jain, Atomate: A high-level interface to generate, execute, and analyze computational materials science workflows, Comput. Mater. Sci. **139**, 140 (2017).

[83] S. P. Ong, W. D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. L. Chevrier, K. A. Persson, and G. Ceder, Python materials genomics (pymatgen): A robust, open-source Python library for materials analysis, Comput. Mater. Sci. **68**, 314 (2013).

[84] G. Kresse and J. Furthmüller, Efficient iterative schemes for *ab initio* total-energy calculations using a plane-wave basis set, Phys. Rev. B **54**, 11169 (1996).

[85] G. Kresse and J. Furthmüller, Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set, Comput. Mater. Sci. **6**, 15 (1996).

[86] P. E. Blöchl, Projector augmented-wave method, Phys. Rev. B **50**, 17953 (1994).

[87] C. Bergmeir, R. J. Hyndman, and B. Koo, A note on the validity of cross-validation for evaluating autoregressive time series prediction, Comput. Stat. Data Anal. **120**, 70 (2018).

[88] L. Prechelt, Early stopping—But when? *Neural Networks: Tricks of the Trade* (Springer, 2002), pp. 55–69.

[89] M. Li, M. Soltanolkotabi, and S. Oymak, Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks, arXiv:1903.11680.

[90] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, ArcFace: Additive angular margin loss for deep face recognition, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 4690–4699.

[91] G. Ma, Y. Wang, D. Lim, S. Jegelka, and Y. Wang, A canonicalization perspective on invariant and equivariant learning, Adv. Neural Inf. Process. Syst. **37**, 60936 (2024).

[92] J. Burgess, J. J. Nirschl, M.-C. Zanellati, A. Lozano, S. Cohen, and S. Yeung-Levy, Orientation-invariant autoencoders learn robust representations for shape profiling of cells and organelles, Nat. Commun. **15**, 1022 (2024).

[93] C.-H. Chang, C.-N. Chou, and E. Y. Chang, Clkn: Cascaded Lucas-Kanade networks for image alignment, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 2213–2221.

[94] V. Boussot, C. Hémon, J.-C. Nunes, J. Dowling, S. Rouzé, C. Lafond, A. Barateau, and J.-L. Dillenseger, IMPACT: A generic semantic loss for multimodal medical image registration, arXiv:2503.24121.

[95] L. Pu, R. G. Govindaraj, J. M. Lemoine, H.-C. Wu, and M. Brylinski, DeepDrug3D: Classification of ligand-binding pockets in proteins with a convolutional neural network, PLoS Comput. Biol. **15**, e1006718 (2019).

[96] J. Martinka, M. Pederzoli, M. Barbatti, P. O. Dral, and J. Pittner, A simple approach to rotationally invariant machine learning of a vector quantity, J. Chem. Phys. **161**, 174104 (2024).

[97] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, PointNet: Deep learning on point sets for 3D classification and segmentation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 652–660.

[98] G. Chevrot, P. Calligari, K. Hinsen, and G. R. Kneller, Least constraint approach to the extraction of internal motions from molecular dynamics trajectories of flexible macromolecules, J. Chem. Phys. **135**, 084110 (2011).