Fine-Tuning MLIPs Through the Lens of Iterated Maps With BPTT

Evan Dramko [1] **Yizhi Zhu** [2 3] **Aleksander Krivokapic** [4] **Geoffroy Hautier** [2 3] **Thomas Reps** [5]
Christopher Jermaine [1] **Anastasios Kyrillidis** [1]

Abstract

Accurate structural relaxation is critical for advanced materials design. Traditional approaches built on physics-derived first-principles calculations are computationally expensive, motivating the creation of machine-learning interatomic potentials (MLIPs), which strive to faithfully reproduce first-principles computed forces. We propose a fine-tuning method to be used on a pretrained MLIP in which we create a fully-differentiable end-to-end simulation loop that optimizes the predicted final structures directly. Trajectories are unrolled and gradients are tracked through the entire relaxation. We show that this method consistently improves performance across all evaluated pretrained models; resulting in an average of roughly 32% reduction in prediction error. Interestingly, we show the process is robust to substantial variation in the relaxation setup, achieving negligibly different results across varied hyperparameter and procedural modifications.

1. Introduction

A central task in computational materials science is the identification of physically realizable atomic structures. In practice, this amounts to finding atomic configurations that correspond to local minima of the *potential energy surface* (PES), which maps atomic coordinates—given fixed species and electronic state—to potential energy. This work addresses how machine learning interatomic potentials (MLIPs) can be trained to more effectively predict relaxed states, without requiring additional expensive first-principles data.

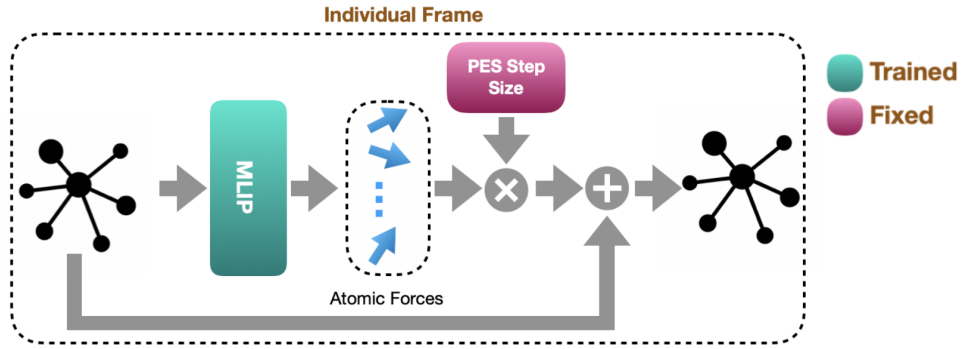
Structural Relaxation and Its Cost. Stable structures are typically obtained through *relaxation trajectories*: iterative, gradient-based procedures that update an initial configuration toward lower-energy states using PES gradients. These gradients correspond to interatomic forces and are conventionally computed using *density functional theory* (DFT). While DFT provides accurate forces at the quantum-mechanical level, each evaluation is computationally expensive, with cost scaling steeply with system size, basis choice, and functional. As a result, full structural relaxations can require hours to days on high-performance computing systems, making relaxation a major bottleneck in high-throughput atomistic workflows.

MLIPs for Efficient Relaxation. To reduce this cost, machine learning interatomic potentials (MLIPs), also referred to as machine learning force fields, are trained to approximate DFT forces and energies (Batatia et al., 2022; Chen & Ong, 2022; Choudhary & DeCost, 2021; Deng et al., 2023; Yang et al., 2024a; Cheon et al., 2020; Musaelian et al., 2023; Dramko et al., 2025). Rather than directly predicting relaxed structures, MLIPs are typically used within iterative optimization loops to emulate DFT-driven relaxation at a fraction of the computational cost. In practice, they often serve either as full replacements for DFT during relaxation or as pre-relaxation tools that move structures close to equilibrium before final DFT refinement (Rossignol et al., 2023; Dramko et al., 2025). Although direct structure prediction models have received growing attention, MLIPs remain the dominant and most reliable approach for nontrivial structural domains (see Section 2.2).

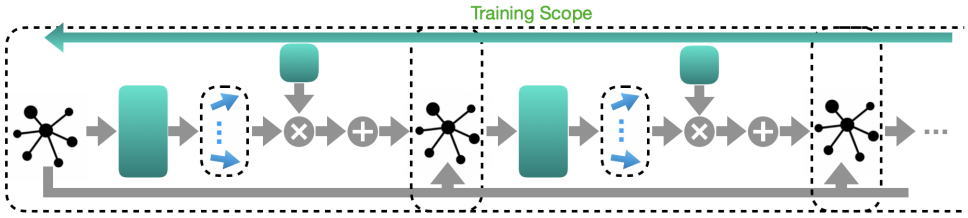
Data Limitations and Motivation. A fundamental challenge in MLIP development is data scarcity. Each new training example requires costly first-principles calculations, resulting in datasets that are orders of magnitude smaller and less diverse than those in many other machine learning domains (Huang et al., 2023; Hörmann et al., 2025). Consequently, improving performance by simply scaling datasets is often impractical. This work instead focuses on extracting more value from existing data by altering *how* MLIPs are trained, rather than *what* data they are trained on.

[1]Department of Computer Science, Rice University, Houston, USA [2]Department of Materials Science and Nanoengineering, Rice University, Houston, USA [3]Rice Advanced Materials Institute, Rice University, Houston, USA [4]Faculty of Technical Sciences, University of Novi Sad, Novi Sad, Serbia [5]Department of Computer Sciences, University of Wisconsin–Madison, Madison, USA. Correspondence to: Evan Dramko <evan.dramko@rice.edu>, Geoffroy Hautier <geoffroy.hautier@rice.edu>.

Preprint. March 24, 2026.



(a) Traditional Training Scope



(b) BPTT fine-tuning Scope

Figure 1. In previous work, only the MLFF itself is trained. In our work, we train through the whole relaxation, treating the entire unrolled trajectory as one large neural network.

1.1. Levels of Optimization

Two distinct optimization processes are central to this work: (1) optimization of the MLIP parameters during training, and (2) optimization of atomic coordinates along the PES during structural relaxation. To avoid ambiguity, we refer to parameter updates of the MLIP as *MLIP training*, and to coordinate updates during relaxation as *PES-level optimization*. The key idea of this paper is to explicitly couple these two levels during training.

1.2. Trajectory-Level Training via BPTT

Conventional MLIP training treats each structure–force pair independently, optimizing per-step force accuracy without regard to how errors accumulate during relaxation.We instead adopt a trajectory-level training strategy.

Specifically, we unroll full relaxation trajectories and apply *backpropagation through time* (BPTT) to update model parameters based on the quality of the *final relaxed structure*, rather than intermediate force errors. This reframes MLIP training as a problem of optimizing the outcome of the relaxation process itself. By supervising the model using trajectory-level signals, the MLIP learns to bias its force predictions using structural context, effectively allowing one data modality (final structures) to supervise another (forces). Crucially, this approach improves relaxation performance without requiring additional first-principles data. An overview of the training procedure is shown in Figure 1.

Contributions. Our primary contributions are as follows:

1. We introduce a full-trajectory, BPTT-based fine-tuning framework for MLIPs.
2. We provide ablation studies and analysis of the components involved in PES-level optimization.
3. We connect BPTT-based training to the theory of iterative maps and proxy functions.
4. We validate the approach across multiple structural domains and MLIP architectures.

2. Related Works

2.1. MLIPs

MLIPs are trained to take as inputs structure snapshots[1] and produce predictions for forces and formation energy. A frequent direction of research in MLIPs has been graph

[1]Structure snapshots: atomic locations, atomic descriptors, etc.

neural networks (GNNs), which have led to many of the leading architectures (Batatia et al., 2022; Chen & Ong, 2022; Choudhary & DeCost, 2021; Deng et al., 2023; Yang et al., 2024a; Cheon et al., 2020; Musaelian et al., 2023).

Another common approach is the use of multi-layer perceptrons (and related base architectures). Appendix A.2 covers many of the variations used in literature.

However, a recent push in literature has been to abandon graph representations in favor of Transformer encoders (Dramko et al., 2025; Kreiman et al., 2025; Elhag et al., 2025), or some variant such as Graphformer based models (Eissler et al., 2026). These techniques have shown state-of-the-art success across both molecular and crystal datasets.

2.2. Direct Structure Prediction

There has been growing interest in predicting relaxed atomic structures directly, without explicitly optimizing along the PES. However, because structural relaxation is a highly non-convex optimization problem, the data requirements for one-shot approaches are typically prohibitive (Sercu et al., 2021; Shu et al., 2018; Yang et al., 2024c). While limited success has been demonstrated in narrowly constrained settings, such as two-dimensional point defects (Yang et al., 2025), direct prediction remains challenging for nontrivial systems and consistently underperforms force-driven iterative ML relaxations on standard benchmarks (Kolluru et al., 2022; Yang et al., 2024c). As a result, iterative relaxation methods remain the clear gold standard and dominate practical deployment settings.

Another semi-direct approach, (Yang et al., 2024b), uses an internal iterative geometry solver to predict the final structure, but remains largely confined to the original study, with no demonstrated uptake or benchmarking in the broader literature or in industry.

2.3. Iterative Approaches

The literature has explored weight-tying and iterative maps extensively (Almeida, 1988; Bai et al., 2019), with *deep equilibrium networks (DEQs)* (Bai et al., 2019) matching our general problem formulation. The majority of the research (Bai et al., 2019; 2021; Daniele et al., 2025) has not been focused on the physical sciences. While such techniques have been shown to be successful under the right conditions (Bai et al., 2019; Agarwala & Schoenholz, 2022), they often exhibit poor training dynamics and generalizability (Sun & Shi, 2024; Agarwala & Schoenholz, 2022; Bai et al., 2021; Gabor et al., 2024). For this reason, they are often used in memory-constrained circumstances (Sun & Shi, 2024; Gabor et al., 2024), and for theoretical exploration (Sun & Shi, 2024; Gabor et al., 2024; Gao et al., 2023; Daniele et al., 2025). Recent work (Wang et al., 2024b) applies DEQs to predict self-consistent Hamiltonians directly, attempting to bypass the iterative SCF procedure that constitutes a major computational bottleneck in DFT.

One notable related technique from materials science literature is DOGSS (Yoon & Ulissi, 2020). This technique learns a network that creates parameters to condition a simple proxy function which matches desired physical properties at the minima. The network's conditioning of the proxy function is trained though the fully-differentiable gradient descent optimization of the proxy. DOGSS however, does not integrate with MLIP literature, and instead focuses on spring constants and equilibrium distances.

2.4. Sequence-Level Works

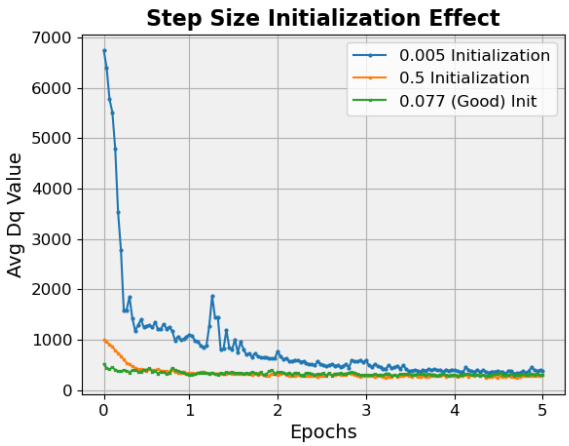
Backpropagation through time (Rumelhart et al., 1986; Werbos, 2002; Williams & Zipser, 1989) is the standard method for computing gradients in recurrent neural networks, and is widely used throughout sequence-learning literature. It remains the conventional baseline for training RNNs, and underpins most subsequent developments in recurrent and sequence-model learning.

There has been substantial work throughout machine learning on full-sequence training, often in the form of reinforcement learning from human feedback (RLHF) (Christiano et al., 2017; Ouyang et al., 2022). While some aspects such as the use of proximal term weight controls (Cohere et al., 2025) or gradient clipping (Schulman et al., 2017; Raffel et al., 2020) overlap with our work, we differ strongly in that we have a fully-differentiable setup.

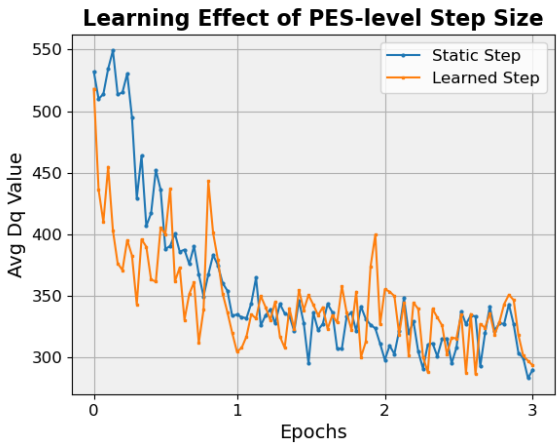
Some work, (Greener, 2024; 2025), has used fully-differentiable training to match molecular dynamics (MD) trajectory coefficients to those observed in practice. These values are useful in many aspects of materials science, but are orthogonal to our goal of structural relaxation. Another work (Krueger et al., 2024) finds BPTT unsuitable to train a specific coarse-grained molecular dynamics simulation engine.[2]

Some papers have looked at trajectory-level supervision through differential equations (Chen et al., 2018; Cranmer et al., 2020), and in the training of Hamiltonian approximators (Greydanus et al., 2019; Chen et al., 2019; Zhong et al., 2019; David & Méhats, 2023). They learn a single function which is then integrated over many steps using a symplectic integrator. Backpropagation is computed through the steps of the integration.

[2]This engine is not a neural network, nor should it be considered neural-network-like.



(a) Effect of varying learned PES-step parameter initializations



(b) Sample run of fixed vs learned step

Figure 2. Experimental results comparing learned and fixed scalar step sizes as applied to Si defects. Error reported as Total Dq across test set.

3. Method

3.1. Preliminaries

Notation. In a slight abuse of notation, we denote $\mathbf{X}$ as the atomic structure under consideration, and also as an $\mathbb{R}^{n\times 3}$ matrix of their coordinates. The ground truth relaxed structure is given by $\mathbf{X}^*_F$, and the predicted structure is $\widehat{\mathbf{X}}_\mathbf{F}$. We denote actual forces with $\mathbf{y}$ and predicted forces as $\widehat{\mathbf{y}}$.

Evaluation: Loss and Delta Q. Many works use MSE as a measure for structural error. However, in crystal defect cases, MSE can overemphasize the effect of the bulk lattice, and underemphasize the effect of the defect center (Dramko et al., 2025). To handle this issue, we use Delta Q (`Dq`) as a loss function, which is MSE where the error from each atom is weighted by its atomic mass. Given a set $\mathbf{X}^*$ of n atoms with masses m_i, and a predicted set of atom locations $\widehat{\mathbf{X}}$, `Dq` is defined by:

$$\texttt{Dq}(\mathbf{X}^*_\mathbf{F}, \widehat{\mathbf{X}}_\mathbf{F}) = \sqrt{\sum_{j=1}^{n} m_j \|\mathbf{X}^*{}_{j,:} - \widehat{\mathbf{X}}_{j,:}\|_2^2}, \qquad (1)$$

This metric is referred as the mass-weighted displacement between different states and is physically meaningful, often used to quantify structural shift in first-principles calculations (Alkauskas et al., 2014).

Guiding Example. Throughout Sections 3 and 4 we use the ADAPT architecture (Dramko et al., 2025) as described in Appendix A and the crystal defects dataset (Appendix E.1) as a motivating example. ADAPT is a graph-free, coordinate-based Transformer encoder architecture used for atomic force prediction and long-range interaction modeling. It is introduced with the crystal defects dataset and contains studies showing near-optimal performance with regard to network size and training length, giving a strong point of comparison for BPTT fine-tuning. Section 5, however, shows that the performance gains in relaxed structure prediction are not limited to this architecture and dataset, but generalize across other architecture and model combinations. Experimental results indicate that, consistent with observations from RLHF-style fine-tuning, only a small number of epochs are required to achieve meaningful improvements (Ziegler et al., 2019; Ouyang et al., 2022). Figure 2 presents outcomes obtained after five epochs of fine-tuning showing that training has stabilized. We adopt five epochs as the standard training length across experiments.

3.2. Defining `frame`

We denote by `frame`, a single DFT/MLIP force-prediction step followed by a corresponding structural-update step. Without loss of generality, `frame` is expressed in Eq. 2, as follows.

$$\texttt{frame}(\mathbf{X}_t) \to \mathbf{X}_{t+1} \overset{\text{def}}{=} \begin{cases} \widehat{\mathbf{y}} & = \text{MLIP}(\mathbf{X}_t) \\ \mathbf{X}_{t+1} & = \mathbf{X}_t + \eta\widehat{\mathbf{y}} \end{cases} \qquad (2)$$

where we denote the matrix of coordinates for the structure as $\mathbf{X}_t$ and predicted forces as $\widehat{\mathbf{y}}$, and η gives the chosen PES-level step size. While our modeling employs discrete gradient descent along the PES, the physical realization of the system follows gradient-flow dynamics, guaranteeing convergence to a local minimum.

3.3. Trajectory Unrolling

In this BPTT-driven fine-tuning scheme, we unroll and train end-to-end through entire relaxation trajectories. Recalling

frame from Eq. 2, we define a rollout as:

$$\widehat{\mathbf{X}}_{\mathbf{F}} = \underbrace{\texttt{frame} \circ \cdots \circ \texttt{frame}}_{\times k}(\mathbf{X}_0) \tag{3}$$

The stopping condition is non-differentiable; it is defined as either a small $\mathcal{L}_2$ distance between successive steps, or reaching a maximum threshold of steps, whichever comes first. We do not train with knowledge of or through this stopping condition explicitly; instead, we use it as a computational stopping point and separately take the Dq between the predicted and final structure as our loss metric:

$$\mathcal{L} = \texttt{Dq}(\mathbf{X}_F^*, \widehat{\mathbf{X}}_{\mathbf{F}}) \tag{4}$$

Fundamentally, we are training an AI model such that: $\texttt{frame}^k(\mathbf{X_0}) = \widehat{\mathbf{X}}_{\mathbf{F}} \rightarrow \mathbf{X}_F^*$. Note that k may not be the same for each rollout, meaning that we are inherently training a different function for different samples from the dataset.

3.4. PES-Level Step-Size Controls

Central to any gradient-descent procedure is the choice of step size. In existing atomistic relaxation pipelines, the Fast Inertial Relaxation Engine (FIRE) optimizer (Bitzek et al., 2006) is a common choice. FIRE monitors the alignment between forces and velocities and adaptively adjusts the dynamics by updating the time step, preserving momentum during consistent downhill motion and damping it when the motion becomes unstable, thereby enabling efficient and robust relaxation toward nearby energy minima.

Because FIRE is non-differentiable, it cannot be used within a BPTT fine-tuning loop. We therefore use FIRE together with pretrained ADAPT force and energy models (Dramko et al., 2025) solely as a non-trainable baseline. As a second baseline, we also evaluate a constant step size using the same pretrained models.

Beyond these baselines, we investigate trainable step-size parameterizations under BPTT fine-tuning. Specifically, we consider (1) a single learned scalar step size shared across all relaxation steps, and (2) a small neural network that predicts the step size dynamically during the relaxation trajectory.

Table 1. Ablation studies with ADAPT on the Si defect dataset on determining PES-level step sizes.

Type	Avg. Dq
No BPTT	5.32
Scalar Step Size	2.73
Decoder (Single Value)	3.37

Scalar Step. As another comparison baseline, we perform a grid search to assess the near-optimal performance of pretrained ADAPT under perfect information. We observe that performance is robust across a broad range of step sizes, with values between 0.6 and 0.8 producing the best results on the test set. We select 0.077 as a representative setting for comparison, because it yielded the lowest error. Results for all evaluated step sizes are provided in Appendix F.

We also test using a fixed step size as well as making the step size a learnable parameter. Figure 2b shows there is a negligible difference in performance between the two configurations. Experimental results show that BPTT prefers to update the MLIP weights: in the example shown in Figure 2b, the step size was initialized to 0.50, but ended at ~ 0.49 when exposed to gradient updates. Interestingly, Figure 2a shows that even when we initialize the procedure with a value that deviates far from the pre-computed "ideal" step size, BPTT is not significantly hindered and reaches near identical performance. This reinforces the idea that BPTT fine-tuning is learning about the interplay of the force predictions and the larger descent procedure. The MLIP modifications are however tied to the PES-level step size; when using a model trained on a learned step size initialized to 0.5 in a inference time loop with a step of 0.005, the average Dq score worsens to over 13.77. Furthermore, we find that standard optimization controls such as momentum, noise injection, and annealing do not improve performance and can be detrimental; details are provided in Appendix C.3.

Neural Network Step. For comparison against a non-scalar, but still BPTT-tuned PES-step size, we implement a single-value decoder as seen in Appendix B. This decoder uses the structure and the predicted forces as input, and predicts a PES-level step-size scalar. We evaluate two network initializations: (i) standard random initialization, and (ii) a constant-output initialization obtained by pretraining the model on random noise to reproduce the well-performing value 0.077 identified previously. In practice, both initializations yield indistinguishable performance. Table 1 shows that using a NN-learned step size does not improve model performance as compared to the scalar, further reinforcing the idea that the MLIP is modified to match the PES-step size.

Table 1 shows that BPTT-tuned generated trajectories outperform even the trajectories made with optimal, perfect-information step size with the untuned ADAPT. We find that using BPTT and a fixed step size yields the best overall performance, creating a $\sim 50\%$ reduction in error over even the perfect-information untuned case.

3.5. Connection to Iterative Maps.

Interpreting frame 2 as a gradient-descent update on the potential energy surface, for sufficiently small η, $\mathbf{X}^*$ is a

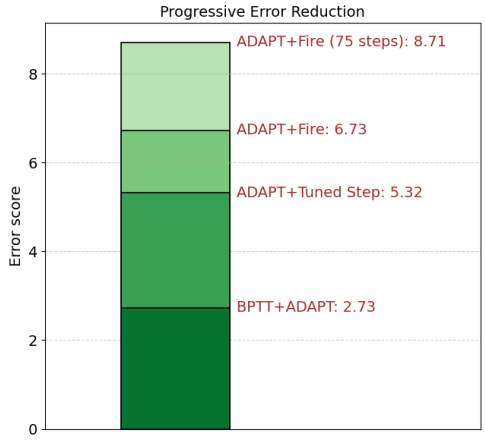


Figure 3. Effect of training schemes on Si defect relaxations

stable fixed point of the iteration:

$$\lim_{n\to\infty} \texttt{frame}^n(\mathbf{X}_n) = \mathbf{X}^* \tag{5}$$

In this formulation, meta-stable structures are naturally identified with stable fixed points of the map `frame` governing the update between consecutive configurations. Under this view, we see that when using our BPTT procedure to fine-tune the model, we are actually learning to approximate the contraction dynamics of the PES through the discretized function `frame`, which serves as a proxy that matches fixed points and relevant basins-of-attraction to the PES.

Key takeaways from these studies are:

1. Figure 3 shows that BPTT is capable of learning a step size that outperforms domain-engineered optimizers,
2. Figure 2a shows that BPTT is capable of learning around a highly non-optimal step size such that the relaxations perform on par with grid-searched, perfect-information-selected initial hyperparameters,
3. Figure 2 shows that BPTT modifies the force predictions such that they produce better relaxed structures, even at the cost of deviating from accuracy in DFT force reproduction as shown in Table 2,
4. Table 1 shows using a neural network defined by $S : \mathbf{X}|_{\text{cat}}\hat{\mathbf{y}} \to \texttt{step}$ does not outperform using a scalar or fixed step size.

4. Analysis and Interpretation

BPTT produces an apparently paradoxical outcome. As shown in Table 2, BPTT degrades force-prediction accuracy, yet Table 1 demonstrates that it substantially improves the final structures produced by the relaxation loop. Figure 5a further shows that allowing BPTT to update only the PES step size—while keeping the MLIP fixed—does not yield performance gains over untrained model. This indicates that the observed improvements are not explained by learning a better set of descent hyperparameters at the PES level.

Taken together, these results suggest that BPTT fine-tuning is neither improving the physical fidelity of force predictions nor optimizing the underlying relaxation dynamics in the conventional sense. Instead, it learns a higher-level structural bias of the dataset. A standard MLIP is not trained to solve structure relaxation directly; rather, it is optimized to reproduce molecular dynamics (MD) trajectories, which are the path atoms travel during relaxation. In principle, perfectly reproducing this path would recover the relaxed structure. In practice, however, limited data coverage and model approximation error lead MLIP-driven MD simulations to accumulate deviations, often resulting in significant error in the predicted relaxed states (Liu et al., 2023; Li et al., 2025).

BPTT reframes this problem. By fine-tuning through the relaxation trajectory, we construct a proxy iterative map, `frame` (Eq. 2), in which the MLIP serves only as an initialization. Recall from Section 3.5 that iterative maps are highly sensitive to initializations. The objective is not to improve adherence to PES-level descent dynamics, but to instead learn a map that preserves key properties of the true PES: most importantly, the locations of fixed points and their associated basins of attraction. As shown in Section 3.5, such fixed points must exist, because DFT relaxation itself is a fixed-point computation. Learning proxy functions that preserve essential properties of first-principles calculations has been shown to be effective for other contexts and objectives in materials science (Yoon & Ulissi, 2020).

Table 2. BPTT finetuning reduces total force accuracy on ADAPT force predictions for Si defects.

Type	$\mathcal{L}_2$ Force Errors
Pretrained MLIP	4.32
BPTT-tuned MLIP	13.36

Finding minima of the potential energy surface (PES) is an inherently non-convex optimization problem. More fundamentally, structural relaxation is an n-body problem; there is no general closed-form analytic expression that exactly captures the mapping from initial to final atomic configurations. The only "perfect" contraction that reliably maps a general structure toward the correct fixed point is the first-principles relaxation itself.

This distinction matters for how to interpret our BPTT training. It does not directly recover the underlying physical dynamics; instead, it trains an iterative map that moves states toward stable fixed points under a chosen update rule. Because the map is parameterized by a neural network and

trained on finite data, it effectively learns a simplified contraction of the true DFT-driven dynamics. In practice, it also implicitly incorporates "direct prediction" information: rather than matching local force accuracy at every point in configuration space, the optimization pressure is dominated by whether repeated application of the learned update steers structures toward the correct endpoints. As a consequence, BPTT-tuned models should not be viewed as universally valid, foundational MLIPs. For general-purpose use, approaches that directly learn forces/energies and then apply physically motivated relaxation (e.g., classical structure-to-force MLIPs and PES-level optimizers) are the more natural target.

However, many of the use cases for MLIPs are not "universal"; they are frequently task-driven. The literature contains many MLIPs designed for specific materials classes or narrowly defined regimes (Takahashi et al., 2017; Saleem & Mallikarjun Sharada, 2025; Stippell et al., 2024; Chen et al., 2025; Dramko et al., 2025). In these settings, BPTT can be highly effective because the model is asked to learn a contraction only over a restricted manifold of structures. The silicon defect relaxation used throughout this work is a representative example: the training distribution is constrained, and success is defined by the model's ability to identify the correct metastable states rather than to reproduce the entire PES. Importantly, our results indicate that the benefit is not confined to extremely narrow tasks. The pure crystal dataset (Section E.2) exhibits substantial diversity—spanning orders of magnitude in system size[3] and covering a broad range of chemistries. However, BPTT fine-tuning still introduces substantial performance gains. This result supports a pragmatic interpretation: while BPTT does not "solve the physics," it can reliably improve the trajectory-level behavior of MLIPs across many applied regimes.

Figure 4. ADAPT Si defect training loss sees negligible improvement, but testing loss is substantially improved

[3]The dataset contains examples ranging from single-digit atom counts to nearly 400 atoms.

Table 3. Full effect of BPTT tuning

Type	Avg Dq	% reduction
ADAPT + FIRE	6.73	48.00
Constant Step ADAPT	5.25	
BPTT Tuned ADAPT	2.73	
ResMLP + FIRE	21.17[†]	45.09
Constant Step ResMLP	10.40	
BPTT Tuned ResMLP	5.71	

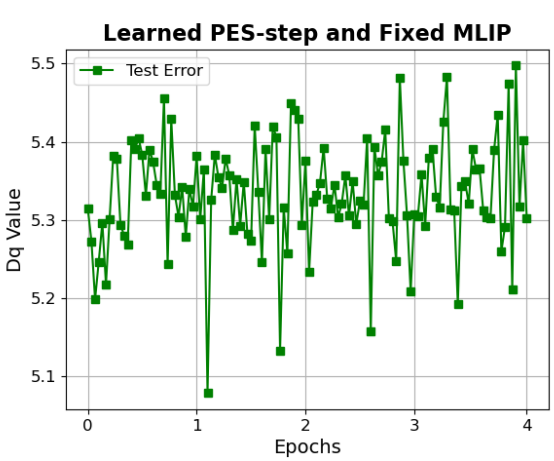
Viewed through this lens, BPTT primarily encourages learned update rules that are stable and contractive in the regions that matter for the dataset, even if local dynamics are distorted. In an idealized scenario of unlimited data and compute, one might hope to learn local force fields so accurately that the learned dynamics match DFT nearly everywhere. In today's MLIP setting, that expectation is unrealistic. Accordingly, we suggest treating BPTT as the final stage of a staged training pipeline: start from a pretrained, broadly applicable MLIP; refine it with supervised structure-to-force training on the target domain; then apply trajectory-level BPTT to shape the long-horizon relaxation behavior. While this approach increases training overhead, the trade-off in high-throughput or computationally heavy studies of specific structure-classes is favorable because improved relaxation fidelity reduces the need for expensive DFT evaluations.

Our experiments reinforce this framing. Figure 5b shows that ADAPT trained with BPTT from random initialization on the silicon defect task fails to make an improvement in performance, indicating that BPTT is not an effective standalone learning paradigm. Its strength is fine-tuning. Consistent with this framing, Figure 4 shows that test accuracy improves substantially even when the training loss remains relatively stable, suggesting gains in trajectory-level generalization rather than simple in-distribution loss-minimization. This pattern mirrors observations in reinforcement learning from human feedback (RLHF), where sequence-level objectives can improve downstream behavior without significant reductions in token-level training loss (Wang et al., 2024a; Hou et al., 2024).

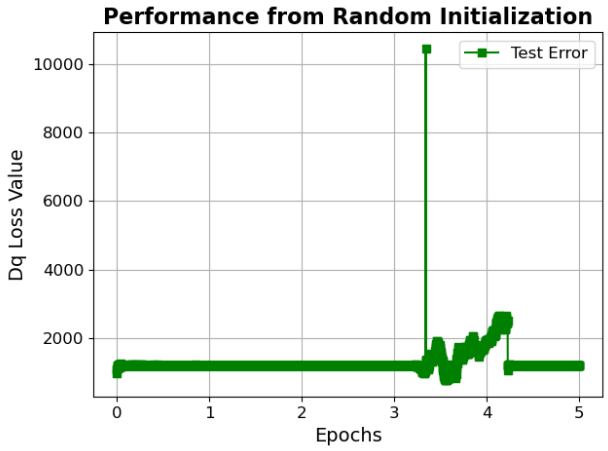
5. Results

We consider three datasets: (1) crystal defects, (2) pure crystals and (3) catalysts. Collectively, these datasets encompass a wide range of applications and modeling scenarios, including both periodic and non-periodic systems, and structures spanning from fewer than 10 atoms to more than 200 atoms. Details on the content and sourcing of the datasets are avail-

[†]12 structures lead to degenerative trajectories and were removed from the Dq calculation.



(a) BPTT cannot adjust the PES-level descent parameters alone, it must affect the MLIP to achieve performance gains.



(b) BPTT performance when training from MLIP random initialization

Figure 5. Effect of adjusting learning setup on Si defect samples

Table 4. Characterizing the effect of BPTT tuning across datasets and models.

Pure Crystals			Catalysts		
Type	**Avg Dq**	**% reduction**	**Type**	**Avg Dq**	**% reduction**
Constant Step ADAPT	8.24	58.00	Constant Step ADAPT	81.82	13.21
BPTT Tuned ADAPT	3.46		BPTT Tuned ADAPT	71.01	
Constant Step ResMLP	129.87	26.13	Constant Step ResMLP	76.45	4.18
BPTT Tuned ResMLP	95.93		BPTT Tuned ResMLP	73.25	

able in Appendix E. We also consider two different models: (1) ADAPT (Dramko et al., 2025), and (2) ResMLP A. Summaries of model architectures and justification of model selections are available in Appendix A.

5.1. Crystal Defects

As summarized in Table 3, fine-tuning with BPTT produces notably improved final configurations. Qualitative analysis of the resulting defect structures from BPTT-tuned ADAPT shows that the defect centers, the regions of greatest practical relevance, exhibit the most pronounced improvements. In particular, BPTT has a significant influence on the localization of interstitial defect atoms in the test cases. These observations confirm that the improvements are not attributable to trivial stabilization of the bulk lattice, which would be unlikely to produce a substantial reduction in the number of DFT steps required for full relaxation. It is shown in (Dramko et al., 2025) that increasing the size and training time of the pretrained ADAPT model on this dataset does not improve its performance; this result demonstrates that BPTT fine-tuning is achieving accuracy in relaxations that would have been impossible with regular MLIP training.

5.2. Pure Crystals and Catalysts

We further validate our approach by fine-tuning ADAPT and ResMLP on datasets of pure crystals and catalysts (see Appendix E). Table 4 shows that, in all cases, BPTT fine-tuning improves performance. For three of the four extra testing cases, and five of six total cases, the performance gain is substantial. Only one testing scenario resulted in a relatively small gain. Figure 6 demonstrates that the BPTT fine-tuning test-loss curves for pure crystals closely resemble that of crystal defects. These results support the conclusion that BPTT fine-tuning is a generalizable strategy that can yield benefits in a variety of scenarios.

6. Discussion

We present a method for fine-tuning MLIPs that uses structure information to modify force predictions. We calculate fully-differentiable relaxation trajectories using MLIPs, and then perform a BPTT update based on a structural accuracy metric. This work shows dramatic increases in predicted structure accuracy, even while decreasing the accuracy of force predictions. We provide ablation studies demonstrating the impact different common gradient-descent schemes have when integrated into the BPTT pipeline, and find the algorithm is capable of adapting around a fixed scalar PES-

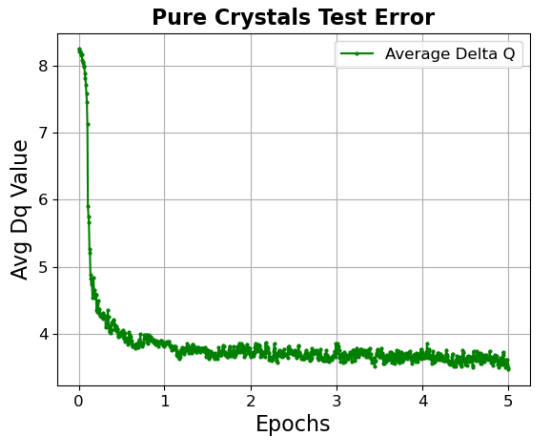


Figure 6. BPTT tuning of ADAPT improves pure crystal results (reduces average Dq) by roughly 50% on the test set.

level step size to match or outperform any other tested procedure. Experimental results indicate this is due to a "preference" of BPTT to modify the MLIP rather than the other trainable parameters. Of particular interest to practitioners is that this approach lowers the data requirements for producing an effective domain-specific MLIP, addressing a common bottleneck in practical deployment.

References

Agarwala, A. and Schoenholz, S. S. Deep equilibrium networks are sensitive to initialization statistics. In *International Conference on Machine Learning*, pp. 136–160. PMLR, 2022.

Alkauskas, A., Buckley, B. B., Awschalom, D. D., and Van de Walle, C. G. First-principles theory of the luminescence lineshape for the triplet transition in diamond nv centres. *New Journal of Physics*, 16(7):073026, 2014.

Almeida, L. B. Backpropagation in perceptrons with feedback. In *Neural computers*, pp. 199–208. Springer, 1988.

Artrith, N. and Urban, A. An implementation of artificial neural-network potentials for atomistic materials simulations: Performance for tio2. *Computational Materials Science*, 114:135–150, 2016.

Bai, S., Kolter, J. Z., and Koltun, V. Deep equilibrium models. *Advances in neural information processing systems*, 32, 2019.

Bai, S., Koltun, V., and Kolter, J. Z. Stabilizing equilibrium models by jacobian regularization. *arXiv preprint arXiv:2106.14342*, 2021.

Batatia, I., Kovacs, D. P., Simm, G., Ortner, C., and Csányi, G. Mace: Higher order equivariant message passing neural networks for fast and accurate force fields. *Advances in neural information processing systems*, 35: 11423–11436, 2022.

Behler, J. Perspective: Machine learning potentials for atomistic simulations. *The Journal of chemical physics*, 145(17), 2016.

Behler, J. and Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical review letters*, 98(14):146401, 2007.

Bitzek, E., Koskinen, P., Gähler, F., Moseler, M., and Gumbsch, P. Structural relaxation made simple. *Physical review letters*, 97(17):170201, 2006.

Chanussot, L., Das, A., Goyal, S., Lavril, T., Shuaibi, M., Riviere, M., Tran, K., Heras-Domingo, J., Ho, C., Hu, W., Palizhati, A., Sriram, A., Wood, B., Yoon, J., Parikh, D., Zitnick, C. L., and Ulissi, Z. W. The open catalyst 2020 (oc20) dataset and community challenges. *ACS Catalysis*, 11(10):6059–6072, 2021. doi: 10.1021/acscatal.0c04525.

Chen, B., Hua, Z., Watkins, J. K., Malakkal, L., Khafizov, M., Hurley, D. H., and Jin, M. Machine learning interatomic potential for predicting the thermal properties of uranium nitride. *Journal of Applied Physics*, 138(20), 2025.

Chen, C. and Ong, S. P. A universal graph deep learning interatomic potential for the periodic table. *Nature Computational Science*, 2(11):718–728, 2022.

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Chen, Z., Zhang, J., Arjovsky, M., and Bottou, L. Symplectic recurrent neural networks. *arXiv preprint arXiv:1909.13334*, 2019.

Cheon, G., Yang, L., McCloskey, K., Reed, E. J., and Cubuk, E. D. Crystal structure search with random relaxations using graph networks. *Preprint at https://arxiv.org/abs/2012.02920*, 2020.

Choudhary, K. and DeCost, B. Atomistic line graph neural network for improved materials property predictions. *npj Computational Materials*, 7(1):185, 2021.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Cohere, T., Ahmadian, A., Ahmed, M., Alammar, J., Alizadeh, M., Alnumay, Y., Althammer, S., Arkhangorodsky, A., Aryabumi, V., Aumiller, D., et al. Command a:

An enterprise-ready large language model. *arXiv preprint arXiv:2504.00698*, 2025.

Cranmer, M., Greydanus, S., Hoyer, S., Battaglia, P., Spergel, D., and Ho, S. Lagrangian neural networks. *arXiv preprint arXiv:2003.04630*, 2020.

Daniele, C., Villa, S., Vaiter, S., and Calatroni, L. Deep equilibrium models for poisson imaging inverse problems via mirror descent, 2025. URL https://arxiv.org/abs/2507.11461.

David, M. and Méhats, F. Symplectic learning for hamiltonian neural networks. *Journal of Computational Physics*, 494:112495, 2023.

Deng, B., Zhong, P., Jun, K., Riebesell, J., Han, K., Bartel, C. J., and Ceder, G. Chgnet as a pretrained universal neural network potential for charge-informed atomistic modelling. *Nature Machine Intelligence*, 5(9):1031–1041, 2023.

Dickel, D., Nitol, M., and Barrett, C. Lammps implementation of rapid artificial neural network derived interatomic potentials. *Computational Materials Science*, 196: 110481, 2021.

Dramko, E., Xiong, Y., Zhu, Y., Hautier, G., Reps, T., Jermaine, C., and Kyrillidis, A. Adapt: Lightweight, long-range machine learning force fields without graphs. *arXiv preprint arXiv:2509.24115*, 2025.

Eissler, M., Korjakow, T., Ganscha, S., Unke, O. T., Müller, K.-R., and Gugler, S. How simple can you go? an off-the-shelf transformer approach to molecular dynamics. *The Journal of Chemical Physics*, 164(9), 2026.

Elhag, A. A., Raja, A., Morehead, A., Blau, S. M., Morris, G. M., and Bronstein, M. M. Learning inter-atomic potentials without explicit equivariance. *arXiv preprint arXiv:2510.00027*, 2025.

Gabor, M., Piotrowski, T., and Cavalcante, R. L. Positive concave deep equilibrium models. *arXiv preprint arXiv:2402.04029*, 2024.

Gao, T., Huo, X., Liu, H., and Gao, H. Wide neural networks as gaussian processes: Lessons from deep equilibrium models, 2023. URL https://arxiv.org/abs/2310.10767.

Greener, J. G. Differentiable simulation to develop molecular dynamics force fields for disordered proteins. *Chemical Science*, 15(13):4897–4909, 2024.

Greener, J. G. Reversible molecular simulation for training classical and machine-learning force fields. *Proceedings of the National Academy of Sciences*, 122(22): e2426058122, 2025.

Greydanus, S., Dzamba, M., and Yosinski, J. Hamiltonian neural networks. *Advances in neural information processing systems*, 32, 2019.

Hörmann, L., Stark, W. G., and Maurer, R. J. Machine learning and data-driven methods in computational surface and interface science. *npj Computational Materials*, 11(1):196, 2025.

Hou, Z., Du, P., Niu, Y., Du, Z., Zeng, A., Liu, X., Huang, M., Wang, H., Tang, J., and Dong, Y. Does rlhf scale? exploring the impacts from data, model, and method. *arXiv preprint arXiv:2412.06000*, 2024.

Huang, B., von Rudorff, G. F., and von Lilienfeld, O. A. The central role of density functional theory in the ai age. *Science*, 381(6654):170–175, 2023.

Jain, A., Ong, S. P., Hautier, G., Chen, W., Richards, W. D., Dacek, S., Cholia, S., Gunter, D., Skinner, D., Ceder, G., et al. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL materials*, 1(1), 2013.

Kalayan, J., Ramzan, I., Williams, C. D., Bryce, R. A., and Burton, N. A. A neural network potential based on pairwise resolved atomic forces and energies. *Journal of Computational Chemistry*, 45(14):1143–1151, 2024.

Kolluru, A., Shuaibi, M., Palizhati, A., Shoghi, N., Das, A., Wood, B., Zitnick, C. L., Kitchin, J. R., and Ulissi, Z. W. Open challenges in developing generalizable large-scale machine-learning models for catalyst discovery. *ACS Catalysis*, 12(14):8572–8581, 2022.

Kreiman, T., Bai, Y., Atieh, F., Weaver, E., Qu, E., and Krishnapriyan, A. S. Transformers discover molecular structure without graph priors. *arXiv preprint arXiv:2510.02259*, 2025.

Krueger, R. K., Engel, M. C., Hausen, R., and Brenner, M. P. A differentiable model of nucleic acid dynamics. *arXiv e-prints*, pp. arXiv–2411, 2024.

Li, X., Grandvalet, Y., and Davoine, F. Explicit inductive bias for transfer learning with convolutional networks, 2018. URL https://arxiv.org/abs/1802.01483.

Li, Y., Zhang, X., and Shen, L. A critical review of machine learning interatomic potentials and hamiltonian. *Journal of Materials Informatics*, 5(4):N–A, 2025.

Liu, Y., He, X., and Mo, Y. Discrepancies and error evaluation metrics for machine learning interatomic potentials. *npj Computational Materials*, 9(1):174, 2023.

López-Zorrilla, J., Aretxabaleta, X. M., Yeu, I. W., Etxebarria, I., Manzano, H., and Artrith, N. ænet-pytorch: A gpu-supported implementation for machine learning atomic potentials training. *The Journal of Chemical Physics*, 158 (16), 2023.

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.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21 (140):1–67, 2020.

Rossignol, H., Minotakis, M., Cobelli, M., and Sanvito, S. Machine-learning-assisted construction of ternary convex hull diagrams, 2023. URL https://arxiv.org/abs/2308.15907.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

Saleem, U. and Mallikarjun Sharada, S. Anomalous diffusion of metal atoms on oxide surfaces: A machine learning molecular dynamics study of pt1/tio2. *The Journal of Physical Chemistry C*, 129(18):8663–8676, 2025.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Sercu, T., Verkuil, R., Meier, J., Amos, B., Lin, Z., Chen, C., Liu, J., LeCun, Y., and Rives, A. Neural potts model. *bioRxiv*, pp. 2021–04, 2021.

Shu, R., Bui, H. H., Zhao, S., Kochenderfer, M. J., and Ermon, S. Amortized inference regularization. *Advances in Neural Information Processing Systems*, 31, 2018.

Stippell, E., Alzate-Vargas, L., Subedi, K. N., Tutchton, R. M., Cooper, M. W., Tretiak, S., Gibson, T., and Messerly, R. A. Building a dft+ u machine learning interatomic potential for uranium dioxide. *Artificial Intelligence Chemistry*, 2(1):100042, 2024.

Sun, H. and Shi, Y. Understanding representation of deep equilibrium models from neural collapse perspective. *Advances in Neural Information Processing Systems*, 37: 9634–9667, 2024.

Takahashi, A., Seko, A., and Tanaka, I. Conceptual and practical bases for the high accuracy of machine learning interatomic potentials: application to elemental titanium. *Physical Review Materials*, 1(6):063801, 2017.

Tam, K.-M., Walker, N., Kellar, S., and Jarrell, M. Interatomic potential in a simple dense neural network representation. *arXiv preprint arXiv:1911.01365*, 2019.

Wang, B., Zheng, R., Chen, L., Liu, Y., Dou, S., Huang, C., Shen, W., Jin, S., Zhou, E., Shi, C., et al. Secrets of rlhf in large language models part ii: Reward modeling. *arXiv preprint arXiv:2401.06080*, 2024a.

Wang, Z., Liu, C., Zou, N., Zhang, H., Wei, X., Huang, L., Wu, L., and Shao, B. Infusing self-consistency into density functional theory hamiltonian prediction via deep equilibrium models, 2024b. URL https://arxiv.org/abs/2406.03794.

Werbos, P. J. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 2002.

Williams, R. J. and Zipser, D. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

Xiong, Y., Bourgois, C., Sheremetyeva, N., Chen, W., Dahliah, D., Song, H., Zheng, J., Griffin, S. M., Sipahigil, A., and Hautier, G. High-throughput identification of spin-photon interfaces in silicon. *Science Advances*, 9 (40):eadh8617, 2023.

Xiong, Y., Zheng, J., McBride, S., Zhang, X., Griffin, S. M., and Hautier, G. Computationally driven discovery of t center-like quantum defects in silicon. *Journal of the American Chemical Society*, 146(44):30046–30056, 2024.

Yang, C. and Ma, X. Improving stability of fine-tuning pretrained language models via component-wise gradient norm clipping, 2022. URL https://arxiv.org/abs/2210.10325.

Yang, H., Hu, C., Zhou, Y., Liu, X., Shi, Y., Li, J., Li, G., Chen, Z., Chen, S., Zeni, C., et al. Mattersim: A deep learning atomistic model across elements, temperatures and pressures. *arXiv preprint arXiv:2405.04967*, 2024a.

Yang, Z., Zhao, Y., Liu, X., Zhang, X., Li, Y., Lyu, Q., Yu-Chian Chen, C., and Shen, L. Scaling crystal structure relaxation with a universal trustworthy deep generative model. *arXiv e-prints*, pp. arXiv–2404, 2024b.

Yang, Z., Zhao, Y.-M., Wang, X., Liu, X., Zhang, X., Li, Y., Lv, Q., Chen, C. Y.-C., and Shen, L. Scalable crystal structure relaxation using an iteration-free deep generative model with uncertainty quantification. *Nature Communications*, 15(1):8148, 2024c.

Yang, Z., Liu, X., Zhang, X., Huang, P., Novoselov, K. S., and Shen, L. Modeling crystal defects using defect informed neural networks. *npj Computational Materials*, 11(1):229, 2025.

Yoon, J. and Ulissi, Z. W. Differentiable optimization for the prediction of ground state structures (dogss). *Physical Review Letters*, 125(17):173001, 2020.

Zhong, Y. D., Dey, B., and Chakraborty, A. Symplectic ode-net: Learning hamiltonian dynamics with control. *arXiv preprint arXiv:1909.12077*, 2019.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A. ADAPT and ResMLP Architectures

A.1. ADAPT

We adopt the ADAPT architecture (Dramko et al., 2025) as a pretrained MLIP test case due to both its effectiveness and its algorithmic simplicity. The model is defined as three-layer embedding MLP, a series of n Transformer encoder blocks and a final layer linear projection. Each encoder block (denoted $\texttt{enc}$) is given by:

$$\texttt{enc}(\mathbf{X}) \to \mathbf{X}_{out} = \begin{cases} \mathbf{H}_1 & = \texttt{LN}\big(\mathbf{X}_{\text{in}} + \texttt{Attn}(\mathbf{X}_{\text{in}})\big), \\ \mathbf{H}_2 & = \texttt{FFN}\big(\texttt{LN}(\mathbf{H}_1)\big), \\ \mathbf{X}_{\text{out}} & = \texttt{LN}(\mathbf{H}_2 + \mathbf{H}_1) \end{cases} \tag{6}$$

ADAPT also uses projection and embedding operations defined by:

$$\texttt{proj} = \mathbf{W}_p\mathbf{X} \tag{7}$$

$$\texttt{emb} = \mathbf{W}_2 \circ \sigma \circ \mathbf{W}_1 \circ \sigma \circ \mathbf{W}_0(\mathbf{X}) \tag{8}$$

where $\mathbf{W}_p \in \mathbb{R}^{3\times d_{model}}$, $\texttt{emb}$ is a multi-layer perceptron (MLP), and $\circ$ is the function composition operator. Note that the embedding does not use a positional encoding because the coordinates of the each atom is included in the atom features given to the model. The relative ordering of atoms does not matter.

The overall ADAPT architecture is given by:

$$\texttt{MLIP} = \texttt{proj} \circ \underbrace{\texttt{enc} \circ \ldots \circ \texttt{enc}}_{\times n} \circ \texttt{emb}(\mathbf{X}) \tag{9}$$

A.2. ResMLP

Variations on the multilayer-perceptron architecture are a mainstay of MLIP architectures; having been used historically (Behler & Parrinello, 2007; Behler, 2016; Artrith & Urban, 2016) and currently (Tam et al., 2019; Dickel et al., 2021; López-Zorrilla et al., 2023; Kalayan et al., 2024; Dramko et al., 2025) to model the dynamics of atomistic calculations. We adopt a residual+MLP architecture as a testing scenario model, and denote it as ResMLP.

$$\begin{aligned} \mathbf{t}_0 &= \sigma(\mathbf{W}_0\mathbf{x} + \mathbf{b}_0) \\ \mathbf{h}_0 &= \texttt{LN}(\mathbf{P}_0\mathbf{t}_0 + \mathbf{t}_0) \\ \mathbf{t}_1 &= \sigma(\mathbf{W}_1\mathbf{h}_0 + \mathbf{b}_1) \\ \mathbf{h}_1 &= \texttt{LN}(\mathbf{P}_1\mathbf{t}_1 + \mathbf{t}_1) \\ &\vdots \\ \mathbf{t}_5 &= \sigma(\mathbf{W}_5\mathbf{h}_4 + \mathbf{b}_5) \\ \mathbf{h}_5 &= \texttt{LN}(\mathbf{P}_5\mathbf{t}_5 + \mathbf{t}_5) \\ \mathbf{y} &= \mathbf{W}_6\mathbf{h}_5 \end{aligned} \tag{10}$$

where each layer has 4096 nodes, and the output project down to $220 \times 3 = 660$ values. While this may not be the most effective possible architecture in literature, variations of it are commonly used and it provides a strong argument for the generalizability of the BPTT fine-tuning results.

B. Scalar Decoder Head For Step Size

ADAPT provides the forces (gradient) of the PES (objective function), but does not tell us how large of a step we should take. We present a method that takes both the structure and forces as input, and outputs a step size for this iteration of the descent. This step-scaling and update procedure, $\texttt{step}$ is defined in two parts: an upscaling projection to map from the native to embedding dimensions, and a scaler decoder head to produce a single step size for the whole structure. The scaler

decoder head, `dec` is defined by:

$$\texttt{dec}(\mathbf{X}, \mathbf{q}) \to \widehat{\mathbf{y}} = \begin{cases} \mathbf{M} = \texttt{MLIP}(\mathbf{X}), \\ \mathbf{h}_0 = \texttt{LN}(\mathbf{q} + \texttt{Attn}(\mathbf{q}, \mathbf{M}, \mathbf{M})), \\ \mathbf{h}_1 = \texttt{LN}(\mathbf{h}_0 + \texttt{MLP}(\mathbf{h}_0)), \\ \widehat{\mathbf{y}} = \mathbf{W}\mathbf{h}_1 + \mathbf{b}. \end{cases} \tag{11}$$

where the notation follows that of Section 3.1, and dropout is applied after `Attn` and `MLP`. Recall that $\mathbf{M} \in \mathbb{R}^{n \times d_{model}}$, and note that $\mathbf{W} \in \mathbb{R}^{\mathbb{1} \times \mathrm{n}}$. We use a dummy tensor $\mathbf{q}$ of all $1s$ to control the dimension of the output and force it to be scalar. Although it is a matrix, we denote $\mathbf{q} \in \mathbb{R}^{(1 \times d_{model})}$ in lower-case vector form to make clear that it has only one non-trivial dimension.

The full `step` method is given by:

$$\texttt{step}(\mathbf{X}) \to \mathbf{X}' = \begin{cases} \mathbf{Z} = \mathbf{X}\big|_{cat}\texttt{MLIP}(\mathbf{X}), \\ \mathbf{H} = \mathbf{W}_2\mathbf{Z} + \mathbf{W}_1\sigma(\mathbf{W}_0\mathbf{Z}), \\ s = \texttt{dec}(\mathbf{H}), \\ \mathbf{X}' = s \cdot \texttt{MLIP}(\mathbf{X}) + \mathbf{X} \end{cases} \tag{12}$$

Where $\mathbf{W}_2 \in \mathbb{R}^{d_{model} \times 15}$ and if $d_{model} = 15$ then $W_2 = I$.

C. Ablation Studies

C.1. Exploration Noise

A common technique used in sequence fine-tuning tasks is the addition of noise into model weights or update steps with the intention to force the model to explore weights. Such noise is usually annealed over epochs, as the goal of the model shifts from exploration to exploitation. We conduct studies on the effect of noise in our BPTT approach, and find that it leads to vastly diminished results, even performing worse than the original model before the fine-tuning. This is a surprising result given the findings of (Cheon et al., 2020) which claims that noise injection improves overall performance.

C.2. Proximal Term vs Gradient Clipping

Often when performing fine-tuning tasks, a proximal $\mathcal{L}_2$-loss term anchoring the model weights to the pretrained version is added to ensure that the model does not stray too far and overfit (Li et al., 2018). Other have proposed using aggressive gradient clipping as an alternative (Yang & Ma, 2022). Our experiments find there is a negligible performance difference between the two approaches. We adopt the gradient clipping becuase it is more computational lighter compared to the proximal-term approach.

C.3. Use of Momentum and Step Annealing

In optimization literature, pure stochastic gradient descent (SGD) is seldom employed in practice. Instead, machine learning typically utilizes step-size schedulers, momentum variants, or second-order approximations. Domain-specific optimizers have also emerged, such as the FIRE algorithm in atomistic modeling. Noise injection—in various forms[4]—during training can enhance generalization. Moreover, the use of proximal weight penalties and gradient clipping helps constrain fine-tuning updates, ensuring that new policies remain close to their predecessors and mitigating catastrophic forgetting or overfitting.

Since the relaxation trajectory is inherently an optimization procedure, it is reasonable to expect that such modifications to the PES-level descent may have an impact on the BPTT-derived weight updates. Specifically, we perform ablation studies to test the effect of the PES-level application of: (1) noise injection[5], (2) momentum and annealing in steps, and (3) intentional varying of trajectory lengths though modifications of stopping conditions. We also test the effect of the MLIP-level usage of proximal term vs gradient clipping update controls.

[4]Stochasticity can be seen as a form of "noise injection" as well, although we do not test a version of the model without due to compute limitations.

[5]Noise injection has been utilized in literature by (Cheon et al., 2020)

Table 5. Characterizing the effect of momentum weight of 0.05 and 0.99^k annealing-step factor on step k.

Setup	Avg Dq
ADAPT	8.47
ADAPT+Momentum	7.67
ADAPT+Anneal	7.28
ADAPT+Momentum+Anneal	7.49
BPTT	2.73
BPTT+Momentum	3.47
BPTT+Anneal	1928.30
BPTT+Momentum+Anneal	1969.71

We find that none of these modifications substantively improve results, and some cause degenerative trajectories. This indicates that: (1) the model is capable of learning a strong set of optimization controls, and (2) these controls are substantively different than those we typically use.

Applying a momentum term to the descent term of ADAPT without BPTT yields improved results. Similarly, using a multiplicative annealing—such as 0.99^k on step k—yields beneficial results. In Table 5, we report experimental results investigating the effect that these two changes produce when added into the BPTT descent trajectories. We note that the initialization value for the scalar step size hyperparameter is 0.5.

C.4. Intentional Modification of Trajectory Length

During the training process, different examples may run for different numbers of steps. Recalling the view of this problem setup as a deep neural network made of a repeated block, this differing numbers of block instances means that we are optimizing for an inherently different situation in each example. To establish the impact of such potential mismatches, we present experiments in which we artificially vary the stopping threshold after each batch of training. In previous experiments, a threshold of 0.001 in total root mean squared error (RMSE) atomic movement across the structure is used, with a maximum of 75 steps being allowed. Under this new paradigm, we randomly varied the threshold between 0.01 and 0.0001, and randomly varied the maximum number of frames between 50 and 75. Results from this are presented in Table 6, and show that such a training setup leads to slightly worse results. The hope of this training scheme— now disproved—is that it would worsen results on the training set, but could force better generalization and improve test set performance.

Table 6. Characterizing the effect of varying stopping conditions

Stopping Condition	Avg Dq
Fixed	2.92
Variable	3.19

D. Hyperparameters and Reproducibility

Unless otherwise noted in Section C.4 or Section C.3, results are from trials using an RMSE threshold of 0.001. We impose a maximum of 75 frames in a trajectory for Silicon defects and pure crystals, and 200 frames for catalysts.

E. Datasets

Among modern research directions in advanced materials, *crystals*—periodic, infinitely repeating arrangements of atoms known as lattices—are of particular relevance. We validate BPTT fine-tuning by testing on two different datasets which represent fundamentally different types of crystal domains. The success of BPTT fine-tuning under such different domains shows that its benefit is not limited to a specific problem scope.

E.1. Silicon Crystal Defects

A central theme in materials design is the deliberate introduction of *defects*: controlled disruptions of an otherwise perfect lattice to engineer specific electrical, magnetic, or quantum properties. This dataset (Xiong et al., 2023; 2024) focuses on

point defects in silicon. Point defects occur wherever the crystal's regular structure is broken in a constrained local region. They are typically classified into three categories: i) **substitution:** a lattice atom is replaced with a different element; ii) **interstitial:** an additional atom is inserted between lattice sites; and, iii) **vacancy:** a lattice site is missing an atom.

These seemingly simple atomic irregularities—vacancies or irregular atoms within a crystal lattice—are fundamental to modern electronic and quantum technologies. They can also combine to form defect complexes, where multiple point defects interact within the same region of the crystal. Such engineered defects are the basis of many modern technologies such as semiconductor doping for high-performance computing, and they are central to emerging efforts to create stable qubits for quantum communication and sensing applications.

E.2. CHGNet Pure Crystals Data Subset

We use the dataset curation from CHGNet (Deng et al., 2023), which is itself a curation of data from the Materials Project (Jain et al., 2013). To avoid prohibitively large runtimes, we select the first $20,101$ examples as the dataset. This dataset contains pure crystal cells. Each cell is assumed to be subject to the periodic boundary condition, and is part of an infinitely repeating regular lattice. Due to the smaller number of atoms and lack of specific interest areas, we use standard MSE loss as our metric for this dataset. Since no pretrained version of ADAPT, we train ADAPT (Dramko et al., 2025) first on first predictions, then use BPTT fine-tuning. We show that BPTT leads to a substantial increase in final-structure accuracy, as shown in Figure 6.

Matching to the quantities set in (Dramko et al., 2025), we use the first 100 trajectories as testing data and the rest as training data. Pure crystal trajectories are often much shorter than those from defects, so we use a substantially larger quantity to create a dataset of similar number of frames to that of the silicon defects from (Dramko et al., 2025). The MLIP is first trained on the structure-force pairs from the flattened $20,001$ training trajectories. After, the model is fine-tuned using BPTT on the $16,984$ trajectories of length greater than two with constant atom counts throughout the trajectory. Finally, testing results on the 96 of the 100 test trajectories are reported, four trajectories (sequentially trajectories 15, 17, 19, 63) has mismatched atom counts throughout the trajectory in the dataset, and were thus discarded.

E.3. Catalyst Dataset

We use datasets from the open catalyst project (Chanussot et al., 2021). We use the "s2ef_train_200K" dataset to train the force-to-structure underlying model, and we use the "is2res_train_val_test_lmdbs" to train the sequence-level BPTT updates. Matching with the other evaluation setups, the first 100 trajectories from the sequence-level dataset are used as the testing case, with the other trajectories providing the training data.

F. ADAPT Optimal PES Step

In Table 7 we show the average testing `Dq` for selected values of the PES-level step size. Notably, comparing BPTT against one of these values unfairly disadvantages BPTT because BPTT has no knowledge of the the test set, whereas these hyperparameters are fit directly on the test set.

G. The Effect of Graph Cutoff Building

Both ADAPT and ResMLP utilize coordinate-in-space based architectures. This means that the data representation does not change based on its structural evolution. With graphs however, as the structure evolves, atoms may move in or out of the range of other atoms' cutoff radii. This can result in different representations of the graph over the course of the trajectory. Since this is a non-differentiable decision boundary, we cannot train BPTT directly through graph construction. In many structures, this creates a relatively small problem, as structural evolution often induces in a very limited movement among any given atom in the structure. While BPTT can be implemented to ignore these non-differentiable decisions, it is a potential area of concern for structures with widespread and substantial structural evolution.

Additionally, many graph architectures have adopted the implementation choice of using backpropagation to calculate forces from energy predictions. This can result in a very convoluted gradient chain during BPTT, and users must be careful not to consume the computation graph during the force prediction stage.

Table 7. Hyperparameter Values for PES-Level Descent of ADAPT

Step Size	**Avg `Dq`**
0.001	12.38
0.005	9.88
0.007	9.15
0.009	8.61
0.011	8.23
0.013	7.90
0.015	7.63
0.017	7.46
0.019	7.29
0.021	7.16
0.025	6.90
0.029	6.71
0.033	6.49
0.037	6.35
0.039	6.27
0.045	6.05
0.049	5.91
0.063	5.42
0.067	5.38
0.071	5.33
0.073	5.34
0.075	5.33
0.077	5.32
0.079	5.37
0.081	5.41
0.083	5.40
0.085	5.40
0.087	5.45
0.091	5.35
0.095	5.37
0.15	9.18
0.25	20.46