

Equivariant Atomic and Lattice Modeling Using Geometric Deep Learning for Crystal Structure Optimization

Ziduo Yang¹, Yi-Ming Zhao², Xian Wang², Wei Zhuo³, Xiaoqing Liu², Lei Shen^{2,4*}

¹College of Information Science and Technology, Jinan University, Guangzhou, 510632, China

²Department of Mechanical Engineering, National University of Singapore, 9 Engineering Drive 1, 117575, Singapore

³College of Computing and Data Science, Nanyang Technological University, 50 Nanyang Avenue, 639798, Singapore

⁴National University of Singapore (Chongqing) Research Institute, Chongqing, 401123, China

Abstract

Structure optimization, which yields the relaxed structure (minimum-energy state), is essential for reliable materials property calculations, yet traditional *ab initio* approaches such as density-functional theory (DFT) are computationally intensive. Machine learning (ML) has emerged to alleviate this bottleneck but suffers from two major limitations: (i) existing models operate mainly on atoms, leaving lattice vectors implicit despite their critical role in structural optimization; and (ii) they often rely on multi-stage, non-end-to-end workflows that are prone to error accumulation. Here, we present E³Relax, an end-to-end equivariant graph neural network that maps an unrelaxed crystal directly to its relaxed structure. E³Relax promotes both atoms and lattice vectors to graph nodes endowed with dual scalar-vector features, enabling unified and symmetry-preserving modeling of atomic displacements and lattice deformations. A layer-wise supervision strategy forces every network depth to make a physically meaningful refinement, mimicking the incremental convergence of DFT while preserving a fully end-to-end pipeline. We evaluate E³Relax on four benchmark datasets and demonstrate that it achieves remarkable accuracy and efficiency. Through DFT validations, we show that the structures predicted by E³Relax are energetically favorable, making them suitable as high-quality initial configurations to accelerate DFT calculations.

Code — <https://github.com/Shen-Group/E3Relax>

Datasets — <https://zenodo.org/records/14607178>

Extended version — <https://arxiv.org/abs/2511.12243>

Introduction

Structural optimization (or relaxation), which yields the lowest energy state or the most stable atomic configuration of a material (commonly called the relaxed structure), is a fundamental step in computational materials science because material properties are typically calculated based on relaxed structures. Traditionally, this process is achieved using *ab initio* methods, particularly density functional theory (DFT) (Gibson, Hire, and Hennig 2022; Lyngby and Thygesen 2022). However, DFT calculations are computationally intensive due to the iterative procedures involved in

*Corresponding author.

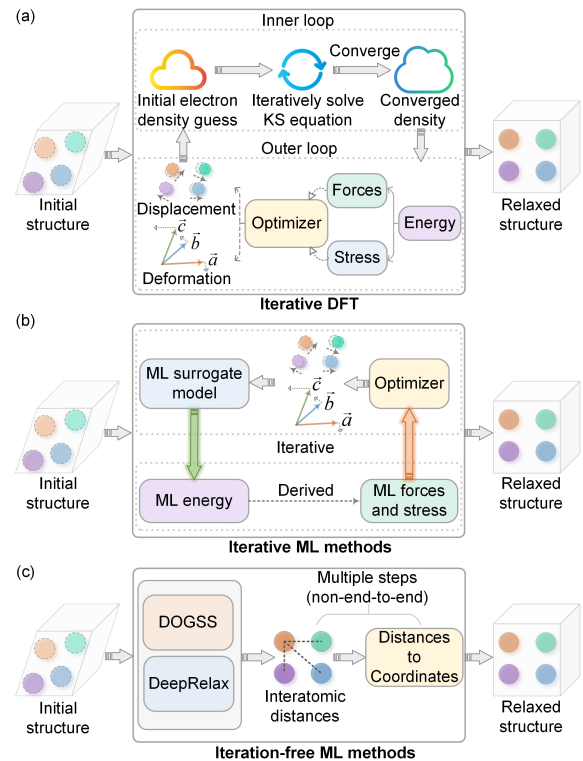


Figure 1: Structural optimization using DFT and ML methods. (a) DFT-based structural optimization consists of two nested loops. (b) Iterative ML models replace the DFT inner loop but still run a geometry-update outer loop. (c) Existing iteration-free ML methods bypass iterative updates by first predicting the relaxed-structure interatomic distances and then reconstructing atomic coordinates.

two nested loops, as depicted in Figure 1(a). The inner loop solves the Kohn–Sham equations to self-consistency, yielding the total energy and atomic forces for a fixed geometry. The outer loop then moves each atom according to those forces and repeats the inner loop until the forces fall below a convergence threshold. Together, these loops drive the structure to its minimum-energy configuration.

To overcome these limitations, there has been growing in-

terest in alternative methods, particularly machine learning (ML), for predicting relaxed structures more efficiently. ML methods for structural optimization can be broadly categorized into two types: iterative approaches (Chen and Ong 2022; Deng et al. 2023; Batatia et al. 2022; Gasteiger, Groß, and Günnemann 2020; Mosquera-Lois et al. 2024; Jiang et al. 2024; Yang et al. 2025b) and iteration-free methods (Kim et al. 2023; Yoon and Ulissi 2020; Yang et al. 2024). Iterative ML methods simplify the DFT calculation by eliminating the inner loop while retaining the outer loop calculations. At each step, a GNN-based interatomic potential predicts the energy, forces, and stresses for the current atomic configuration; these quantities then drive a geometry update, after which the updated structure is fed back into the model for a new prediction, as illustrated in Figure 1(b). Despite their efficiency gains, iterative ML approaches have two key limitations. First, the development of iterative approaches heavily relies on the availability of comprehensive databases with detailed labels for energy, forces, and stresses during structural optimizations. Second, although these methods eliminate the inner loop, the outer loop remains, meaning the optimization process is still iterative and thus limits parallel scalability.

To alleviate these issues, iteration-free methods have been developed to predict relaxed structures directly from the input unrelaxed ones using GNNs, entirely bypassing iterative loops. For example, DeepRelax (Yang et al. 2024) and DOGSS (Yoon and Ulissi 2020) employ a two-stage scheme: they first predict the set of interatomic distances for the relaxed structure, then reconstruct the atomic coordinates by solving a Euclidean distance geometry problem (Figure 1(c)). These iteration-free approaches nonetheless exhibit two key limitations. First, they model atomic interactions via message passing but do not explicitly account for lattice deformations. However, structure optimization typically involves both atomic displacements and lattice deformations, and variations in lattice parameters can substantially influence material properties. Consequently, the absence of explicit lattice modeling limits the accuracy of current GNNs in predicting relaxed structures. Second, although they avoid both inner and outer loop iterations, these methods often use a multi-step workflow instead of a fully end-to-end mapping, which can accumulate errors and reduce overall efficiency.

To address these challenges, we introduce E³Relax, an equivariant graph neural network designed for the end-to-end prediction of relaxed structures in a single step, as shown in Figure 2(a). E³Relax offers three advantages:

- **Unified atomic and lattice modeling.** E³Relax promotes both atoms and lattice vectors to graph nodes, each equipped with an invariant scalar and an equivariant 3D vector feature. This dual-node design enables the network to capture atomic displacements and lattice deformations simultaneously, overcoming the limitation of earlier models that represent only atoms.
- **True end-to-end optimization.** E³Relax achieves end-to-end trainability by directly learning the mapping from unrelaxed to relaxed structures. This single-step frame-

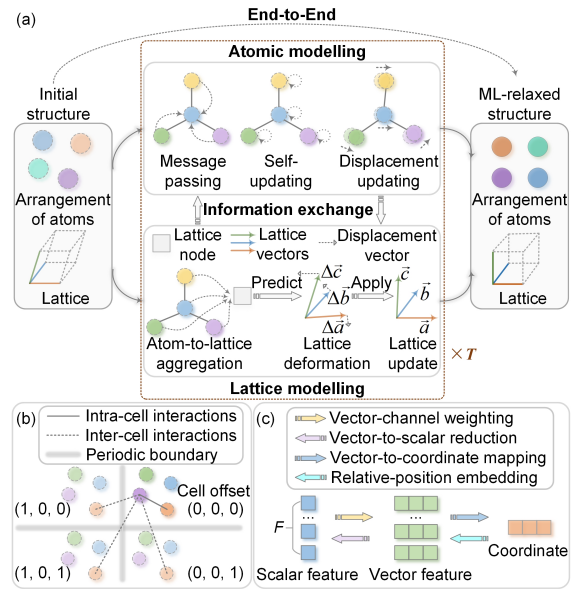


Figure 2: An overview of E³Relax. (a) The model jointly captures atomic displacements and lattice deformations during structural optimization by modeling both atoms and the lattice vectors. (b) Illustration of a multi-edge graph showing an atom connected to the same neighboring atoms in different translated unit cells. (c) Depiction of the three varieties of node features along with their interrelations.

work avoids the potential cumulative error of multi-step approaches, thereby improving performance and efficiency.

- **Progressive structure refinement.** E³Relax utilizes layer-wise supervision to control the refinement process of atomic coordinates and lattice vectors throughout its network layers. This layer-by-layer guidance allows the model to progressively refine its predictions by mirroring the incremental convergence of DFT optimizations within a single, iteration-free pass. We empirically demonstrate that this strategy yields substantial performance gains.

Related Work

Iterative ML Models for Structural Optimization

Iterative ML methods streamline DFT optimization by replacing the self-consistency inner loop with a GNN that iteratively predicts energies, forces, and stresses, while still retaining the outer geometry-update loop. The central challenge in these methods is designing GNN architectures that accurately capture atomic interactions. To ensure consistent energy predictions under translations, rotations, and reflections, many previously state-of-the-art (SOTA) models enforce invariance to the Euclidean group E(3) by relying on features such as bond lengths and angles (Xie and Grossman 2018; Schütt et al. 2017; Deng et al. 2023; Gasteiger, Groß, and Günnemann 2020; Choudhary and DeCost 2021; Chen and Ong 2022; Gasteiger, Becker, and Günnemann 2021;

Omee et al. 2024; Li et al. 2025). More recently, equivariant networks, which explicitly incorporate crystal symmetries and provide a richer geometric description, have demonstrated superior accuracy (Batatia et al. 2022; Batzner et al. 2022; Xu et al. 2024; Wang et al. 2025; Park et al. 2024; Schütt, Unke, and Gastegger 2021; Deng et al. 2023). However, iterative ML potentials still rely on extensive energy, force, and stress labels and retain an outer optimization loop, making the process inherently sequential and limiting parallel scalability.

Iteration-Free Models for Structural Optimization

Iteration-free methods bypass both the inner and outer optimization loops entirely (Yang et al. 2024; Yoon and Ulissi 2020; Kim et al. 2023; Yang et al. 2025a), aiming to directly map an unrelaxed structure to its relaxed counterpart without iterative computation. However, current SOTA models, such as DeepRelax (Yang et al. 2024), DOGSS (Yoon and Ulissi 2020), and Cryslator (Kim et al. 2023), typically adopt multi-step pipelines, making them non-end-to-end. Crucially, all existing iteration-free models focus exclusively on atoms, without providing an explicit representation for the lattice vectors. Since the cell’s geometry (its lattice parameters) directly affects properties like electronic band structure, density, and mechanical behavior, leaving out an explicit lattice representation may lead to less accurate or physically inconsistent predictions.

Methodology

Figure 2(a) summarizes the E³Relax workflow. Starting from an unrelaxed crystal defined by its atomic coordinates $\vec{r}^{(0)}$ and lattice vectors $\vec{l}^{(0)}$, the network applies a sequence of equivariant graph-convolution layers. After the t -th layer the structure is updated to $\vec{r}^{(t)}$ and $\vec{l}^{(t)}$; successive layers therefore refine both atomic positions and lattice parameters until the final layer outputs $\vec{r}^{(T)}$ and $\vec{l}^{(T)}$.

The key challenge is to maintain strict SE(3)-equivariance at every layer so that rotations or translations of the input produce identically transformed outputs for both atoms and lattice vectors. Preserving this symmetry is essential for physical correctness, robust generalization, and orientation-independent predictions (Batatia et al. 2022; Batzner et al. 2022; Satorras, Hoogeboom, and Welling 2021; Schütt, Unke, and Gastegger 2021). We achieve it by representing each atom and lattice vector with a paired scalar-vector feature set and by constraining all network operations to be SE(3)-equivariant.

Each equivariant graph-convolution layer performs four steps: (1) message passing, during which a node gathers information from its neighbors to mimic atomic interactions; (2) self-updating, which refines the node’s internal features; (3) atom position updating; and (4) lattice updating, which jointly adjusts atomic coordinates and lattice vectors toward lower energy. An additional lattice-atom interaction module enables the model to capture the coupling between atomic displacements and lattice deformations. All operations are SE(3)-equivariant, ensuring that atomic coordinates and lat-

tice vectors transform consistently under rotation and translation.

Materials Graph Representation

A crystal is encoded as a multi-edge graph $\mathcal{G} = (\mathcal{V}^a, \mathcal{V}^l, \mathcal{E})$, where $v_i \in \mathcal{V}^a$ ($i = 1 \dots M$) are atom nodes and $v_c \in \mathcal{V}^l$ ($c = 1, 2, 3$) are lattice nodes corresponding to the three lattice vectors. Unlike traditional materials graphs, which include only atom nodes, this dual-node formulation makes the lattice an explicit part of the graph structure.

Edges $e_{ij} \in \mathcal{E}$ connect each atom to its K nearest neighbours within a cut-off D . Under periodic boundary conditions, the same atom pair may be connected by multiple edges, each annotated with a unit-cell offset (k_1, k_2, k_3) corresponding to a translation $k_1\vec{l}_1 + k_2\vec{l}_2 + k_3\vec{l}_3$ (Figure 2(b)), where $\vec{l}_1, \vec{l}_2, \vec{l}_3$ are the three primitive lattice vectors. Every lattice node is linked to all atom nodes in the unit cell. Inspired by Matformer (Yan et al. 2022), we further connect each atom node to its images in the six neighbouring cells—offsets $(0, 0, 1)$, $(0, 1, 0)$, $(1, 0, 0)$, $(0, 1, 1)$, $(1, 0, 1)$, and $(1, 1, 0)$ —to expose long-range periodicity.

Each atom node carries an invariant scalar $x_i \in \mathbb{R}^F$, an equivariant vector $\vec{x}_i \in \mathbb{R}^{F \times 3}$, and its position $\vec{r}_i \in \mathbb{R}^3$. Each lattice node is also assigned three distinct features, denoted as $y_c \in \mathbb{R}^F$, $\vec{y}_c \in \mathbb{R}^{F \times 3}$, and $\vec{l}_c \in \mathbb{R}^3$. Initialisation details of these features are provided in Appendix A. The interactions among these feature types are illustrated in Figure 2(c). For each atom pair we also define the relative vector $\vec{r}_{ij} = \vec{r}_j - \vec{r}_i$ to encode local geometry.

Message Passing

Message passing is a method by which an atom node v_i collects information from its neighboring nodes v_j (Schütt, Unke, and Gastegger 2021). In the t -th layer, during message passing, each node v_i aggregates messages from the scalar features $x_j^{(t)}$ and vector features $\vec{x}_j^{(t)}$ of its neighboring nodes v_j . This aggregation forms intermediate scalar and vector variables m_i and \vec{m}_i , respectively, calculated as follows:

$$m_i = \sum_{v_j \in \mathcal{N}(v_i)} \phi_h(\mathbf{x}_j^{(t)}) \circ \gamma_h(\lambda(\|\vec{r}_{ji}^{(t)}\|)) \quad (1)$$

$$\begin{aligned} \vec{m}_i = & \sum_{v_j \in \mathcal{N}(v_i)} \phi_u(\mathbf{x}_j^{(t)}) \circ \gamma_u(\lambda(\|\vec{r}_{ji}^{(t)}\|)) \circ \vec{x}_j^{(t)} \\ & + \phi_v(\mathbf{x}_j^{(t)}) \circ \gamma_v(\lambda(\|\vec{r}_{ji}^{(t)}\|)) \circ \frac{\vec{r}_{ji}^{(t)}}{\|\vec{r}_{ji}^{(t)}\|} \end{aligned} \quad (2)$$

Here, $\mathcal{N}(v_i)$ denotes the set of neighboring nodes to v_i . The functions $\phi_h, \phi_u, \phi_v : \mathbb{R}^F \rightarrow \mathbb{R}^F$ are multilayer perceptrons (MLPs), λ represents K Gaussian radial basis functions (RBF) (Schütt et al. 2017) used to expand bond distances, and $\gamma_h, \gamma_u, \gamma_v : \mathbb{R}^K \rightarrow \mathbb{R}^F$ are also MLPs.

Self-Updating

Self-updating leverages each node’s internal context to fuse scalar and vector features. This process combines the F

scalar and F vector elements within \mathbf{m}_i and $\vec{\mathbf{m}}_i$ to generate updated scalar $x_i^{(t+1)}$ and vector $\vec{x}_i^{(t+1)}$ as follows:

$$x_i^{(t+1)} = \phi_{s1}(\mathbf{m}_i \oplus \|\mathbf{U}\vec{\mathbf{m}}_i\|) + \tanh(\phi_{s2}(\mathbf{m}_i \oplus \|\mathbf{U}\vec{\mathbf{m}}_i\|)) \mathbf{m}_i \quad (3)$$

$$\vec{x}_i^{(t+1)} = \phi_v(\mathbf{m}_i \oplus \|\mathbf{U}\vec{\mathbf{m}}_i\|) \circ (\mathbf{V}\vec{\mathbf{m}}_i) \quad (4)$$

where \oplus denotes concatenation, $\phi_{s1}, \phi_{s2}, \phi_v : \mathbb{R}^{2F} \rightarrow \mathbb{R}^F$ are MLPs, and $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{F \times F}$ are trainable matrices. The $\tanh(\cdot)$ function serves as a gate, regulating how much information from \mathbf{m}_i is preserved.

Atom Position Updating

Atom position updating refines the atomic positions to mirror the structure optimization process. Initially, we compute the coefficients α_i^f for each vector $\vec{x}_i^f \in \vec{\mathbf{x}}_i$ (i.e., $\vec{x}_i^f \in \mathbb{R}^{1 \times 3}$ represents one of the F vectors in $\vec{\mathbf{x}}_i$):

$$p_i = g(\mathbf{x}_i) \quad (5)$$

$$\alpha_i^f = \frac{\exp(p_i^f)}{\sum_k \exp(p_i^k)} \quad (6)$$

where $g : \mathbb{R}^F \rightarrow \mathbb{R}^F$ is an MLP. The positions of the atoms are updated as follows:

$$\Delta \vec{\mathbf{r}}_i = \mathbf{W} \vec{\mathbf{x}}_i + \sum_f \alpha_i^f \vec{x}_i^f \quad (7)$$

$$\vec{\mathbf{r}}_i^{(t+1)} = \vec{\mathbf{r}}_i^{(t)} + \Delta \vec{\mathbf{r}}_i \quad (8)$$

where $\mathbf{W} \in \mathbb{R}^{1 \times F}$ is a trainable matrix.

Lattice Updating

In most crystal GNNs the lattice vectors appear only implicitly: they are not represented as learnable entities, nor are they updated during message passing. To overcome this limitation, we use lattice nodes to represent the lattice vectors. These nodes carry their own features and are refined layer-by-layer alongside the atom nodes.

Specifically, lattice updating begins by first updating the lattice node representations, followed by predicting the displacement vectors and applying these to adjust the lattice vectors. Particularly, we update the lattice node representations $\mathbf{y}_c^{(t)}$ and $\vec{\mathbf{y}}_c^{(t)}$ to intermediate states \mathbf{n}_c and $\vec{\mathbf{n}}_c$ using the following equations:

$$\mathbf{n}_c = \gamma \left(\left(\frac{1}{M} \sum_{v_j \in \mathcal{V}^a} \mathbf{x}_j^{(t)} \right) \oplus \mathbf{y}_c^{(t)} \right) \quad (9)$$

$$\vec{\mathbf{n}}_c = \mathbf{W} \left(\left(\frac{1}{M} \sum_{v_j \in \mathcal{V}^a} \vec{x}_j^{(t)} \right) + \vec{\mathbf{y}}_c^{(t)} \right) \quad (10)$$

where $\gamma : \mathbb{R}^{2F} \rightarrow \mathbb{R}^F$ is an MLP, and $\mathbf{W} \in \mathbb{R}^{F \times F}$ is a trainable matrix.

Next, the updated lattice node representations $\mathbf{y}_c^{(t+1)}$ and $\vec{\mathbf{y}}_c^{(t+1)}$ are computed as:

$$\mathbf{y}_c^{(t+1)} = \phi_{s1}(\mathbf{n}_c \oplus \|\mathbf{U}\vec{\mathbf{n}}_c\|) + \tanh(\phi_{s2}(\mathbf{n}_c \oplus \|\mathbf{U}\vec{\mathbf{n}}_c\|)) \mathbf{n}_c \quad (11)$$

$$\vec{\mathbf{y}}_c^{(t+1)} = \phi_v(\mathbf{n}_c \oplus \|\mathbf{U}\vec{\mathbf{n}}_c\|) \circ (\mathbf{V}\vec{\mathbf{n}}_c) + \vec{\mathbf{n}}_c \quad (12)$$

where $\phi_{s1}, \phi_{s2}, \phi_v : \mathbb{R}^{2F} \rightarrow \mathbb{R}^F$ are MLPs, and $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{F \times F}$ are trainable matrices.

Finally, the lattice coordinates are updated as:

$$\Delta \vec{\mathbf{l}}_c = \mathbf{W}_p \vec{\mathbf{y}}_c^{(t+1)} \quad (13)$$

$$\vec{\mathbf{l}}_c^{(t+1)} = \vec{\mathbf{l}}_c^{(t)} + \Delta \vec{\mathbf{l}}_c \quad (14)$$

where $\mathbf{W}_p \in \mathbb{R}^{1 \times F}$ is a trainable matrix, and $\Delta \vec{\mathbf{l}}_c$ represents the predicted displacement vector (lattice deformation).

Lattice-Atom Interaction

The lattice updating step described above transfers information only from atom nodes to lattice nodes; it does not allow information to flow back from the lattice to the atoms. To remedy this, we introduce a lattice-atom interaction module that broadcasts lattice information to every atom. For each lattice node $v_c \in \mathcal{V}^l$, we first compute intermediate atom node features \mathbf{n}_i and $\vec{\mathbf{n}}_i$ by incorporating the relative distance between atom node $v_i \in \mathcal{V}^a$ and lattice node $v_c \in \mathcal{V}^l$:

$$\mathbf{n}_i = \phi_h(\mathbf{x}_i^{(t)}) \circ \gamma_h \left(\lambda \left(\|\vec{\mathbf{r}}_i^{(t)} - \vec{\mathbf{l}}_c^{(t)}\| \right) \right) + \mathbf{x}_i^{(t)} \quad (15)$$

$$\begin{aligned} \vec{\mathbf{n}}_i = & \phi_u(\mathbf{x}_i^{(t)}) \circ \gamma_u \left(\lambda \left(\|\vec{\mathbf{r}}_i^{(t)} - \vec{\mathbf{l}}_c^{(t)}\| \right) \right) \circ \vec{\mathbf{x}}_i^{(t)} \\ & + \phi_v(\mathbf{x}_i^{(t)}) \circ \gamma_v \left(\lambda \left(\|\vec{\mathbf{r}}_i^{(t)} - \vec{\mathbf{l}}_c^{(t)}\| \right) \right) \circ \frac{(\vec{\mathbf{r}}_i^{(t)} - \vec{\mathbf{l}}_c^{(t)})}{\|\vec{\mathbf{r}}_i^{(t)} - \vec{\mathbf{l}}_c^{(t)}\|} \end{aligned} \quad (16)$$

Here, $\phi_h, \phi_u, \phi_v : \mathbb{R}^F \rightarrow \mathbb{R}^F$ are MLPs, λ represents a set of RBF used to expand bond distances, and $\gamma_h, \gamma_u, \gamma_v : \mathbb{R}^{2F} \rightarrow \mathbb{R}^F$ are also MLPs. We apply this update once per lattice node $c = 1, 2, 3$. Finally, we fuse the intermediate features \mathbf{n}_i and $\vec{\mathbf{n}}_i$ with the current lattice node feature $\mathbf{y}_c^{(t)}, \vec{\mathbf{y}}_c^{(t)}$ to obtain the next-layer atom representations:

$$\mathbf{x}_i^{(t+1)} = f \left(\mathbf{n}_i \oplus \mathbf{y}_c^{(t)} \right) + \mathbf{n}_i \quad (17)$$

$$\vec{\mathbf{x}}_i^{(t+1)} = \mathbf{W}_v \left(\vec{\mathbf{n}}_i + \vec{\mathbf{y}}_c^{(t)} \right) + \vec{\mathbf{n}}_i \quad (18)$$

where $f : \mathbb{R}^{2F} \rightarrow \mathbb{R}^F$ is an MLP and $\mathbf{W}_v \in \mathbb{R}^{F \times F}$ is a trainable matrix. Again, we apply this update once per lattice node $c = 1, 2, 3$.

Layer-Wise Supervision

If supervision is applied only to the final output, the network may simply copy its intermediate representations forward and perform one large correction in the last layer. Such abrupt ‘‘jumps’’ are hard to learn. Instead, E³Relax attaches a loss to every layer so that each depth must contribute a small, physically meaningful correction. In other words, the model is trained to progressively approach the same optimal configuration across all depths. The overall loss is defined as:

$$\mathcal{L} = \sum_{t=1}^T \alpha_t \left(\sum_{v_i \in \mathcal{V}^a} |\vec{\mathbf{r}}_i^{(t)} - \tilde{\mathbf{r}}_i| + \sum_{v_c \in \mathcal{V}^l} |\vec{\mathbf{l}}_c^{(t)} - \tilde{\mathbf{l}}_c| \right) \quad (19)$$

where α_t controls each layer’s contribution, and all α_t are set to 1 by default.

Equivariance Analysis

In E³Relax, all non-linear activations (MLPs, gating functions, etc.) are applied exclusively to scalar feature channels (e.g., x_j), while every operation on vector features (e.g., \vec{x}_j) is either a scaling (i.e., \odot) or linear combinations (e.g., $W\vec{x}_j$). By construction, following the equivariant message-passing framework of PAINN (Schütt, Unke, and Gastegger 2021), these components guarantee exact SE(3)-equivariance throughout the network.

Implementation Details

The E³Relax model is implemented using PyTorch and experiments are conducted on an NVIDIA L40S GPU with 48 GB of memory. We employ the AdamW optimizer with an initial learning rate of 0.0001 to update the model parameters. Besides, we incorporate a learning rate decay strategy, which reduces the learning rate if no improvement is observed in a specific performance metric over a period of 5 epochs.

Experiments

Datasets

We mainly use four datasets in this study: Materials Project (MP) (Jain et al. 2013), X-Mn-O (X = Mg, Ca, Ba, Sr) (Kim et al. 2023), C2DB (Gjerding et al. 2021; Lyngby and Thygesen 2022), and Layered van der Waals (vdW) crystals (Yang et al. 2024). Each data point comprises a pair of structures, i.e., one unrelaxed and one DFT-relaxed. These datasets include both 2D and 3D materials and represent a wide range of compositions and structural complexities, making them well-suited benchmarks for evaluating structure optimization models. Detailed descriptions of these datasets are provided in Appendix B.

Baselines

Consistent with the protocol used in DeepRelax (Yang et al. 2024), we compare E³Relax with SOTA iteration-free equivariant models, including DeepRelax (Yang et al. 2024), EquiformerV2 (Liao et al. 2024), HEGNN (Cen et al. 2024), GotenNet (Aykent and Xia 2025), PAINN (Schütt, Unke, and Gastegger 2021), and EGNN (Satorras, Hoogeboom, and Welling 2021). Each baseline is evaluated using the authors’ original implementations and default hyperparameter settings. All models are evaluated on identical training, validation, and testing sets. The Dummy model, which simply returns the input initial structure as the output, served as a control in our assessments. Detailed information on baseline implementations can be found in Appendix C.

Performance Indicators

We use three metrics to quantify the discrepancy between predicted and DFT-relaxed structures: mean absolute error (MAE) of atomic coordinates, cell shape deviation, and MAE of cell volume. The coordinate MAE evaluates positional errors, while cell shape deviation and volume MAE assess lattice geometry errors. Detailed definitions and computation procedures are provided in Appendix D.

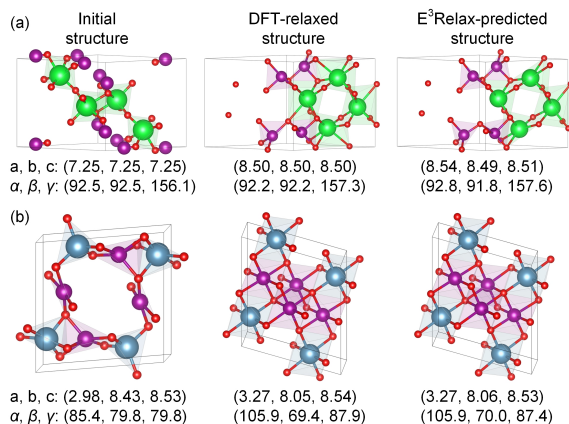


Figure 3: Two crystal structures from the X-Mn-O dataset optimized with E³Relax: (a) Mn₄O₈Sr₄ and (b) Ca₄Mn₄O₈. a , b , and c are lattice constants in angstroms (Å), and α , β , and γ are angles in degrees (°).

Results on 3D Materials Datasets

We first evaluate E³Relax on the two 3D materials datasets. As shown in Table 1, E³Relax consistently surpasses previous SOTA models across all evaluation metrics. For a qualitative assessment of our model, we select two structures from X-Mn-O dataset, Mn₄O₈Sr₄ and Ca₄Mn₄O₈, and optimize them with E³Relax (Figure 3). As can be seen, the E³Relax-predicted structures are highly consistent with the DFT-relaxed ones. Further analysis and visualizations are provided in Appendices E and F.

Results on 2D Materials Datasets

We further evaluate the model on 2D materials using the C2DB and layered vdW datasets. As reported in Table 2, E³Relax maintains strong performance and continues to outperform or remain competitive with existing approaches. Additional results and visualizations are available in Appendices E and F.

Ablation Study

The effectiveness of E³Relax is attributed to three key strategies: First, its dual-node representation explicitly captures both atomic displacements and lattice deformations. Second, the prediction process is designed as a series of refinements of atomic coordinates and lattice vectors. Third, incorporating self-connecting edges into the graph representation enhances the model’s ability to capture long-range periodicity. To evaluate the necessity of these approaches, we compared E³Relax against three variants:

- **w/o lattice nodes:** Here, lattice nodes are removed, and invariant scalar features x_i are employed for predicting relaxed lattice vectors, following the procedure outlined in DeepRelax (Yang et al. 2024).
- **w/o refinement:** This variant omits the progressive refinement process, updating atomic coordinates and lattice vectors only in the final layer without layer-wise supervision.

Model	MP			X-Mn-O		
	MAE Coords.↓	Shape Dev.↓	MAE Vol.↓	MAE Coords.↓	Shape Dev.↓	MAE Vol.↓
Dummy	0.095	11.502	27.006	0.314	13.913	32.839
PAINN	0.088	4.775	9.343	0.159	3.784	3.803
EGNN	0.086	4.834	9.253	0.166	3.814	4.208
HEGNN	0.091	4.987	10.382	0.191	3.862	4.421
GotenNet	0.078	<u>4.111</u>	<u>7.456</u>	0.160	3.669	3.559
EquiformerV2	0.069	4.329	8.539	<u>0.115</u>	3.558	3.713
DeepRelax	<u>0.066</u>	4.554	9.611	0.116	<u>3.530</u>	<u>3.442</u>
E ³ Relax	0.057	4.020	7.417	0.105	3.447	3.369
Improvement	13.64%	2.21%	0.52%	8.70%	2.35%	2.12%

Table 1: Comparison of E³Relax with baseline models on the two 3D materials datasets (MP and X-Mn-O). Metrics are MAE of atomic coordinates (Å), cell shape deviation (Å), and MAE of cell volume (Å³) between ML-relaxed and DFT-relaxed structures. The “Improvement” rows report the percentage gain of E³Relax over the second-best model for each metric. The best results are in bold, and the second-best are underlined.

Model	C2DB			Layered vdW		
	MAE Coords.↓	Shape Dev.↓	MAE Vol.↓	MAE Coords.↓	Shape Dev.↓	MAE Vol.↓
Dummy	0.268	10.615	149.648	0.103	0.172	2.991
PAINN	0.226	5.952	61.897	0.035	0.294	2.782
EGNN	0.232	6.065	67.892	0.033	<u>0.098</u>	1.636
HEGNN	0.244	6.329	71.625	0.049	0.312	4.989
GotenNet	0.208	5.704	55.800	<u>0.028</u>	0.204	<u>1.006</u>
EquiformerV2	<u>0.188</u>	<u>5.660</u>	60.435	0.034	0.228	3.617
DeepRelax	0.196	5.752	60.216	0.052	0.340	1.166
E ³ Relax	0.171	5.430	<u>56.909</u>	0.025	0.087	0.871
Improvement	9.04%	4.07%	-1.99%	10.71%	11.22%	13.42%

Table 2: Comparison of E³Relax with baseline models on the two 2D materials datasets (C2DB and Layered vdW).

- **w/o self-connecting edges:** Omits the use of self-connecting edges.

Table 3 presents the experimental results on the MP and X-Mn-O datasets, showing that E³Relax significantly outperforms the three variants. Notably, introducing lattice nodes with equivariant lattice-atom interactions yields substantial performance gains, underscoring the critical role of equivariance in accurately modeling coupled atomic-lattice dynamics.

We also conduct a sensitivity analysis of key hyperparameters, including the number of layers (T), hidden dimension size (F), and layer weights (α_t), on the X-Mn-O dataset. The results confirm that E³Relax is robust to variations in these hyperparameters, with performance remaining stable across different configurations (see Appendix G).

Comparison with Iterative ML Models

To benchmark E³Relax against iterative ML-based potentials, we use a MoS₂ dataset with point defects containing energy, force, and stress labels. This dataset, compiled by Huang et al. (Huang et al. 2023), comprises 5933 unique low-defect configurations in an 8 × 8 supercell. We split the data into 4746 training, 593 validation, and 594 test samples and train two state-of-the-art interatomic potentials,

M3GNet (Chen and Ong 2022) and CHGNet (Deng et al. 2023) (details in Appendix H), using the same splits.

Because all structures share identical lattice parameters before and after optimization, we report only the coordinate MAE. As shown in Figure 4, E³Relax notably outperforms both M3GNet and CHGNet. We also observe that the iterative methods exhibit larger standard deviations in their coordinate errors, likely due to error accumulation over iterative optimization steps.

DFT Validation

To further validate E³Relax, we compute total energies for three sets of structures, unrelaxed, DFT-relaxed, and E³Relax-predicted, to determine whether our predictions are energetically competitive. Detailed DFT parameters are provided in Appendix I. For this analysis, we randomly sampled 100 test-set structures each from the X-Mn-O (3D materials) and C2DB (2D materials) datasets. Figures 5(a) and (b) compare the energy distributions among the three types of structures: the median and spread of the E³Relax-predicted structures closely match those of the fully DFT-relaxed structures, demonstrating our model’s ability to identify near-ground-state configurations.

We further assess how well E³Relax predictions accel-

Model	MP			X–Mn–O		
	MAE Coords.↓	Shape Dev.↓	MAE Vol.↓	MAE Coords.↓	Shape Dev.↓	MAE Vol.↓
Dummy	0.095	11.502	27.006	0.314	13.913	32.839
w/o lattice nodes	0.075	4.389	7.964	0.158	3.576	3.634
w/o refinement	0.075	4.565	8.465	0.146	3.581	3.747
w/o self-connecting edges	0.058	4.114	7.553	0.144	3.839	3.494
E ³ Relax	0.057	4.020	7.417	0.105	3.447	3.369

Table 3: Ablation study on the two 3D materials datasets (MP and X–Mn–O).

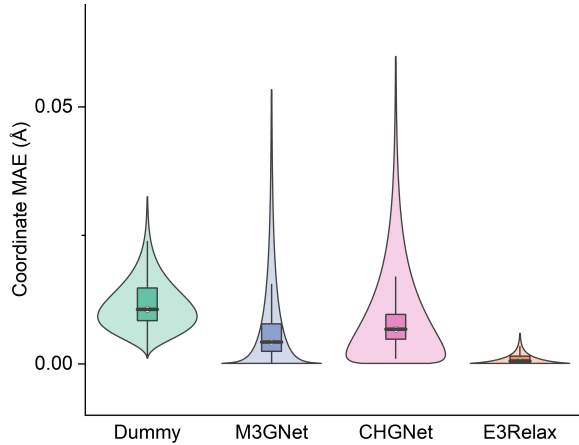


Figure 4: Comparison of coordinate MAE (Å) between E³Relax and two ML-potential models using violin plots.

erate DFT optimizations by measuring the residual number of ionic steps required when using our predicted structures as starting points. We performed this test on the layered vdW crystal dataset and on the MoS₂ point-defect dataset (for which we randomly sampled 20 test structures to limit computational cost). Figures 5(c) and (d) show that, on average, initializing DFT optimization from E³Relax outputs substantially reduces the number of ionic steps needed to converge.

Complexity Analysis

For a crystal containing N atoms with K neighbours each and a hidden feature size F , lightweight equivariant GNNs such as PAINN or EGNN incur a per-layer cost of $\mathcal{O}(NK F^2)$. The extra cost of E³Relax mainly comes from two additional modules: the lattice updating block, costing $\mathcal{O}(3NF^2 + F^2)$, and the lattice–atom interaction block, costing $\mathcal{O}(3NF^2)$. Combining these with the base message-passing and self-update operations yields a total per-layer complexity of $\mathcal{O}(N(K + 6)F^2)$. When $K \approx 50$ (typical setting), the extra $6NF^2$ overhead represents only about a 12% increase in compute.

We benchmarked inference times on an NVIDIA L40S GPU for two iterative relaxers (CHGNet and M3GNet), a two-stage single-shot model (DeepRelax), and E³Relax. Averaged over structures in the point defect dataset, CHGNet required 3.72 ± 5.13 s, M3GNet 6.64 ± 8.01 s, DeepRelax

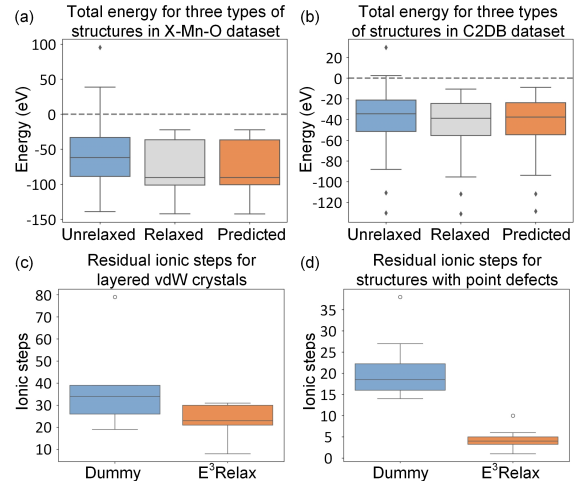


Figure 5: Box plots of total energy distributions and residual ionic steps for E³Relax predictions. (a) and (b) compare total energies of unrelaxed, DFT-relaxed, and E³Relax-predicted structures on the X–Mn–O and C2DB datasets, respectively. (c) and (d) report the number of residual ionic steps required to converge DFT optimizations when initialized from unrelaxed and E³Relax-predicted structures for the layered vdW crystal dataset and the MoS₂ defect dataset, respectively.

0.47 ± 0.04 s, and E³Relax only 0.021 ± 0.001 s per structure. This speed-up stems from E³Relax’s iteration-free, single-step architecture, which eliminates the repeated optimization or multi-stage calls inherent in the other methods.

Conclusion

We have presented E³Relax, an iteration-free, equivariant graph neural network that directly predicts relaxed crystal structures in a single end-to-end step. By promoting both atoms and lattice vectors to learnable graph nodes with dual scalar-vector features, E³Relax captures local displacements and global lattice deformations in a unified, symmetry-preserving framework. Across four benchmark datasets, including both 2D and 3D materials, E³Relax consistently outperforms SOTA iteration-free models. DFT validation further confirms that its predictions lie close to true ground-state energies and substantially reduce the number of ionic steps required for full optimization in DFT calculations.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant No. 62506143), the Natural Science Foundation of Guangdong Province (Grant No. 2025A1515011487), Ministry of Education, Singapore, Tier 1 (Grant No. A-8001194-00-00), and Tier 2 (Grant No. A-8001872-00-00).

References

- Ayken, S.; and Xia, T. 2025. Gotennet: Rethinking efficient 3d equivariant graph neural networks. In *The Thirteenth International Conference on Learning Representations*.
- Batatia, I.; Kovacs, D. P.; Simm, G.; Ortner, C.; and Csányi, G. 2022. MACE: Higher order equivariant message passing neural networks for fast and accurate force fields. *Advances in Neural Information Processing Systems*, 35: 11423–11436.
- Batzner, S.; Musaelian, A.; Sun, L.; Geiger, M.; Mailoa, J. P.; Kornbluth, M.; Molinari, N.; Smidt, T. E.; and Kozinsky, B. 2022. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1): 2453.
- Cen, J.; Li, A.; Lin, N.; Ren, Y.; Wang, Z.; and Huang, W. 2024. Are high-degree representations really unnecessary in equivariant graph neural networks? *Advances in Neural Information Processing Systems*, 37: 26238–26266.
- Chen, C.; and Ong, S. P. 2022. A universal graph deep learning interatomic potential for the periodic table. *Nature Computational Science*, 2(11): 718–728.
- Choudhary, K.; and DeCost, B. 2021. Atomistic line graph neural network for improved materials property predictions. *npj Computational Materials*, 7(1): 185.
- Deng, B.; Zhong, P.; Jun, K.; Riebesell, J.; Han, K.; Bartel, C. J.; and Ceder, G. 2023. CHGNet as a pretrained universal neural network potential for charge-informed atomistic modelling. *Nature Machine Intelligence*, 1–11.
- Gasteiger, J.; Becker, F.; and Günnemann, S. 2021. Gemnet: Universal directional graph neural networks for molecules. *Advances in Neural Information Processing Systems*, 34: 6790–6802.
- Gasteiger, J.; Groß, J.; and Günnemann, S. 2020. Directional Message Passing for Molecular Graphs. In *International Conference on Learning Representations (ICLR)*.
- Gibson, J.; Hire, A.; and Hennig, R. G. 2022. Data-augmentation for graph neural network learning of the relaxed energies of unrelaxed structures. *npj Computational Materials*, 8(1): 211.
- Gjerding, M. N.; Taghizadeh, A.; Rasmussen, A.; Ali, S.; Bertoldo, F.; Deilmann, T.; Knøsgaard, N. R.; Kruse, M.; Larsen, A. H.; Manti, S.; et al. 2021. Recent progress of the computational 2D materials database (C2DB). *2D Materials*, 8(4): 044002.
- Huang, P.; Lukin, R.; Faleev, M.; Kazeev, N.; Al-Maeni, A. R.; Andreeva, D. V.; Ustyuzhanin, A.; Tormasov, A.; Castro Neto, A.; and Novoselov, K. S. 2023. Unveiling the complex structure-property correlation of defects in 2D materials based on high throughput datasets. *npj 2D Materials and Applications*, 7(1): 6.
- Jain, A.; Ong, S. P.; Hautier, G.; Chen, W.; Richards, W. D.; Dacek, S.; Cholia, S.; Gunter, D.; Skinner, D.; Ceder, G.; et al. 2013. Commentary: The Materials Project: A materials genome approach to accelerating materials innovation. *APL materials*, 1(1).
- Jiang, C.; Marianetti, C. A.; Khafizov, M.; and Hurley, D. H. 2024. Machine learning potential assisted exploration of complex defect potential energy surfaces. *npj Computational Materials*, 10(1): 21.
- Kim, S.; Noh, J.; Jin, T.; Lee, J.; and Jung, Y. 2023. A structure translation model for crystal compounds. *npj Computational Materials*, 9(1): 142.
- Li, Y.; Zhang, X.; Liu, M.; and Shen, L. 2025. A Critical Review of Machine Learning Interatomic Potentials and Hamiltonian. *Journal of Materials Informatics*, 5(4): 43.
- Liao, Y.-L.; Wood, B.; Das, A.; and Smidt, T. 2024. EquiformerV2: Improved Equivariant Transformer for Scaling to Higher-Degree Representations. In *International Conference on Learning Representations (ICLR)*.
- Lyngby, P.; and Thygesen, K. S. 2022. Data-driven discovery of 2D materials by deep generative models. *npj Computational Materials*, 8(1): 232.
- Mosquera-Lois, I.; Kavanagh, S. R.; Ganose, A. M.; and Walsh, A. 2024. Machine-learning structural reconstructions for accelerated point defect calculations. *npj Computational Materials*, 10(1): 121.
- Omeo, S. S.; Wei, L.; Hu, M.; and Hu, J. 2024. Crystal structure prediction using neural network potential and age-fitness Pareto genetic algorithm. *Journal of Materials Informatics*, 4(1): 2.
- Park, Y.; Kim, J.; Hwang, S.; and Han, S. 2024. Scalable parallel algorithm for graph neural network interatomic potentials in molecular dynamics simulations. *Journal of chemical theory and computation*, 20(11): 4857–4868.
- Satorras, V. G.; Hoogeboom, E.; and Welling, M. 2021. E (n) equivariant graph neural networks. In *International conference on machine learning*, 9323–9332. PMLR.
- Schütt, K.; Kindermans, P.-J.; Sauceda Felix, H. E.; Chmiela, S.; Tkatchenko, A.; and Müller, K.-R. 2017. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30.
- Schütt, K.; Unke, O.; and Gastegger, M. 2021. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *International Conference on Machine Learning*, 9377–9388. PMLR.
- Wang, C.; Hu, S.; Tan, G.; and Jia, W. 2025. ELoRA: Low-Rank Adaptation for Equivariant GNNs. In *Forty-second International Conference on Machine Learning*.
- Xie, T.; and Grossman, J. C. 2018. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Physical review letters*, 120(14): 145301.

- Xu, Z.; Yu, H.; Bohde, M.; and Ji, S. 2024. Equivariant Graph Network Approximations of High-Degree Polynomials for Force Field Prediction. *Transactions on Machine Learning Research*.
- Yan, K.; Liu, Y.; Lin, Y.; and Ji, S. 2022. Periodic graph transformers for crystal material property prediction. *Advances in Neural Information Processing Systems*, 35: 15066–15080.
- Yang, Z.; Liu, X.; Zhang, X.; Huang, P.; Novoselov, K. S.; and Shen, L. 2025a. Modeling crystal defects using defect informed neural networks. *npj Computational Materials*, 11(1): 229.
- Yang, Z.; Wang, X.; Li, Y.; Lv, Q.; Chen, C. Y.-C.; and Shen, L. 2025b. Efficient equivariant model for machine learning interatomic potentials. *npj Computational Materials*, 11(1): 49.
- Yang, Z.; Zhao, Y.-M.; Wang, X.; Liu, X.; Zhang, X.; Li, Y.; Lv, Q.; Chen, C. Y.-C.; and Shen, L. 2024. Scalable crystal structure relaxation using an iteration-free deep generative model with uncertainty quantification. *Nature Communications*, 15(1): 8148.
- Yoon, J.; and Ulissi, Z. W. 2020. Differentiable optimization for the prediction of ground state structures (DOGSS). *Physical Review Letters*, 125(17): 173001.