Equivariant Diffusion for Crystal Structure Prediction

Peijia Lin [1] **Pin Chen** [1 2] **Rui Jiao** [3 4] **Qing Mo** [2] **Jianhuan Cen** [1] **Wenbing Huang** [5 6] **Yang Liu** [3 4] **Dan Huang** [1] **Yutong Lu** [1 2]

Abstract

In addressing the challenge of Crystal Structure Prediction (CSP), symmetry-aware deep learning models, particularly diffusion models, have been extensively studied, which treat CSP as a conditional generation task. However, ensuring permutation, rotation, and periodic translation equivariance during diffusion process remains incompletely addressed. In this work, we propose EquiCSP, a novel equivariant diffusion-based generative model. We not only address the overlooked issue of lattice permutation equivariance in existing models, but also develop a unique noising algorithm that rigorously maintains periodic translation equivariance throughout both training and inference processes. Our experiments indicate that EquiCSP significantly surpasses existing models in terms of generating accurate structures and demonstrates faster convergence during the training process. Code is available at https://github.com/EmperorJia/EquiCSP.

1. Introduction

Crystal structure prediction (CSP) seeks the atomic arrangement with the lowest energy for given chemical compositions and conditions (Desiraju, 2002), focusing on finding the global minimum of the potential energy surface. This task, while conceptually straightforward, is a significant challenge in physics, chemistry, and materials science due to the complexity of the potential energy landscape and the exponential increase in possible structures with more atoms in a unit cell (Oganov et al., 2019).

Traditional CSP methods primarily employ Density Functional Theory (DFT) (Kohn & Sham, 1965) for iterative energy calculations, integrating optimization algorithms like genetic algorithm (Oganov & Glass, 2006; Oganov et al., 2011) and particle swarm optimization (Wang et al., 2010a; 2012) to navigate the energy landscape for stable states. However, the time-intensive nature of DFT calculations makes these traditional CSP approaches notably inefficient.

Recent advances have seen a shift towards deep generative models, which learn distributions directly from datasets of stable structures (Court et al., 2020; Yang et al., 2021), with diffusion models, a subset of these models, gaining prominence in crystal generation (Xie et al., 2021; Jiao et al., 2023). Diffusion models are lauded for their superior interpretability and performance, owing to their inherent physical explainability. However, developing diffusion models for CSP involves addressing specific challenges. Physically, E(3) transformations, including translation, rotation, and reflection of crystal coordinates, do not change physical laws, necessitating E(3) invariant sample generation in the model. Typical diffusion models like denoising diffusion probabilistic models (DDPMs) (Sohl-Dickstein et al., 2015; Vignac et al., 2023) and score-based generative models with stochastic differential equations (SDEs) (Song et al., 2020), initially used in computer vision, require E(3) equivariance when adapted to molecular graph domains (Luo et al., 2021), and for crystals, additional consideration of periodic invariance is needed (Jiao et al., 2023).

In this study, we introduce EquiCSP, an equivariant[1] diffusion method to address CSP. We track the impact of lattice permutation in crystals on diffusion models used for CSP and propose corresponding solutions. EquiCSP entails that during generation and training, any permutation of lattice parameters results in a corresponding equivariant transformation of the atomic fractional coordinates. Furthermore, we propose a specialized diffusion noising algorithm in

[1]School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China [2]National Supercomputer Center in Guangzhou, China [3]Dept. of Comp. Sci. Tech., Institute for AI, Tsinghua University, Beijing, China [4]Institute for AIR, Tsinghua University, Beijing, China [5]Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China [6]Beijing Key Laboratory of Big Data Management and Analysis Methods, Beijing, China. Correspondence to: Yutong Lu <luyutong@mail.sysu.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

[1]This paper mainly focuses on lattice permutation equivariance and periodic translation equivariance, which are unique to crystallographic data. Traditional rotation equivariance is addressed using invariant representations.

SDEs, meticulously designed to preserve periodic translation equivariance consistently during both training and inference stages.

To summarize, our contributions in this work are as follows:

1. To our knowledge, we are the first to address lattice permutation equivariance in diffusion models, aiming to address that any permutation of lattice parameters during training and generation corresponds to an equivalent transformation in atomic fractional coordinates, achieved through simple and efficient loss functions rather than encoding the equivariance directly into the neural networks.
2. In addition, we propose a novel diffusion noising method, named Periodic CoM-free Noising, to make the widely used Score-Matching method in SDEs achieving the periodic translation equivariance of crystal generation.
3. We validate EquiCSP's effectiveness in CSP tasks, demonstrating its superior performance over existing learning methods (e.g. CDVAE (Xie et al., 2021) and DiffCSP (Jiao et al., 2023)). Furthermore, we enhance EquiCSP for ab initio generation, proving its enhanced performance comparing to similar methods.

2. Related Works

Crystal Structure Prediction. Traditional computational methods, such as DFT combined with optimization algorithms, are used to search for local minima on the potential energy surface (Pickard & Needs, 2011; Yamashita et al., 2018; Wang et al., 2010b; Zhang et al., 2017). Despite their accuracy, these methods are computationally demanding. Recently, machine learning has emerged as an alternative, using crystal databases to predict energy more efficiently than DFT (Jacobsen et al., 2018; Podryabinkin et al., 2019; Cheng et al., 2022). Another approach employs deep generative models to directly learn stable structures, representing crystals with 3D voxels (Court et al., 2020; Hoffmann et al., 2019; Noh et al., 2019), distance matrices (Yang et al., 2021; Hu et al., 2020; 2021), or 3D coordinates (Nouira et al., 2018; Kim et al., 2020; Ren et al., 2021). However, these methods often overlook the complete symmetries in crystal structures.

Equivariant Graph Neural Networks. E(3) symmetric, geometrically equivariant Graph Neural Networks (GNNs) are effective for representing physical objects and have excelled in modeling 3D structures (Schütt et al., 2018; Thomas et al., 2018; Fuchs et al., 2020; Satorras et al., 2021; Thölke & De Fabritiis, 2021), as evidenced in applications like the open catalyst project (Chanussot et al., 2021; Tran et al., 2022). To accommodate periodic materials, multi-graph edge construction (Xie & Grossman, 2018; Yan et al., 2022) and Fourier transforms to fractional coordinates (Jiao et al., 2023) were proposed to represent periodicity. In our work, we utilize Fourier transforms to achieve periodic transition invariance and constrain the additional lattice permutation invariance.

Diffusion Generative Models. Rooted in non-equilibrium thermodynamics theory (Sohl-Dickstein et al., 2015), diffusion models establish a link between data and prior distributions through forward and backward Markov chains (Ho et al., 2020), achieving significant advancements in image generation (Rombach et al., 2021; Ramesh et al., 2022). When integrated with equivariant GNNs, these models efficiently generate samples from invariant distributions, proving effective in tasks such as conformation generation (Xu et al., 2021; Shi et al., 2021), ab initio molecule design (Hoogeboom et al., 2022), and protein generation (Luo et al., 2022). DiffCSP distinguishes itself by simultaneously generating lattice and atom coordinates for crystals, utilizing a periodic-E(3)-equivariant denoising model (Jiao et al., 2023). However, it has yet to fully realize E(3) equivariance based on periodic graph symmetry during its diffusion training process.

3. Preliminaries

Crystal structures. A 3D crystal structure is depicted as an endlessly repeating pattern of atoms in three-dimensional space, with the basic repeating entity known as a 'unit cell'. This unit cell is defined by a triplet $\mathcal{M} = (\boldsymbol{A}, \boldsymbol{X}, \boldsymbol{L})$, where $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n] \in \mathbb{R}^{h \times n}$ symbolizes the one-hot encoded representations of atom types, $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{3 \times n}$ comprises the atoms' Cartesian coordinates and $\boldsymbol{L} = [\boldsymbol{l}_1, \boldsymbol{l}_2, \boldsymbol{l}_3] \in \mathbb{R}^{3 \times 3}$ represents the lattice matrix that indicates the repeating parameters of the unit cell. We represent periodic crystal structure as:

$$\{(\boldsymbol{a}_i', \boldsymbol{x}_i') | \boldsymbol{a}_i' = \boldsymbol{a}_i, \boldsymbol{x}_i' = \boldsymbol{x}_i + \boldsymbol{L}\boldsymbol{k}, \forall \boldsymbol{k} \in \mathbb{Z}^{3 \times 1}\}, \quad (1)$$

where the j-th element of the integral vector $\boldsymbol{k}$ denotes the integral 3D translation in units of $\boldsymbol{l}_j$.

Fractional coordinate system. In crystallography, the fractional coordinate system is often used to represent the periodic nature of crystal structures (Nouira et al., 2018; Kim et al., 2020; Ren et al., 2021; Hofmann & Apostolakis, 2003). This system employs lattice vectors $(\boldsymbol{l}_1, \boldsymbol{l}_2, \boldsymbol{l}_3)$ as coordinate bases, distinguishing it from the Cartesian system with its three orthogonal bases. A point in the fractional coordinate system, denoted by the vector $\boldsymbol{f} = [f_1, f_2, f_3]^\top \in [0, 1)^3$, corresponds to a Cartesian vector $\boldsymbol{x} = \sum_{i=1}^{3} f_i \boldsymbol{l}_i$. All atomic coordinates in a cell compose $\boldsymbol{F} \in [0, 1)^{3 \times n}$. This representation inherently maintains invariance to rotational and reflective transformations of the crystal structure. As described in (Mardia et al., 2000), periodic data on each

lattice base can be visualized as points on a circle, measured by angle value as depicted in Figure 1 (e).

Lattice parameters In crystallography, the lattice matrix $\boldsymbol{L}$ can be converted to an invariant representations with three lattice lengths $\boldsymbol{l} = [l_1, l_2, l_3]^\top$, where $l_i = \|\boldsymbol{l}_i\|_2$, and three lattice angles $\boldsymbol{\phi} = [\phi_{23}, \phi_{13}, \phi_{12}]$, where ϕ_{ij} is the angle between $\boldsymbol{l}_i$ and $\boldsymbol{l}_j$ (Hofmann & Apostolakis, 2003; Luo et al., 2023). This paper employs lattice parameters $\boldsymbol{C} = [\boldsymbol{l}, \boldsymbol{\phi}] \in \mathbb{R}^{3\times 2}$ instead of lattice matrix and represents the crystal by $\mathcal{M} = (\boldsymbol{A}, \boldsymbol{F}, \boldsymbol{C})$.

Task definition. The CSP task entails predicting the lattice parameters $\boldsymbol{C}$ and the fractional matrix $\boldsymbol{F}$ for each unit cell, based on its chemical composition $\boldsymbol{A}$. Specifically, this involves learning the conditional distribution $p(\boldsymbol{C}, \boldsymbol{F} \mid \boldsymbol{A})$.

4. Methods

This section initially outlines the symmetries inherent in crystal geometry, subsequently provides an overview of EquiCSP, and then introduces the joint equivariant diffusion process applied to $\boldsymbol{C}$ and $\boldsymbol{F}$, followed by the architecture of the denoising model.

4.1. Symmetries of Crystal Structure Distribution

The primary challenge of CSP lies in capturing the distribution symmetries of crystal structures. To tackle this, we define four key symmetries as representations within the distribution $p(\boldsymbol{C}, \boldsymbol{F} \mid \boldsymbol{A})$: composition permutation invariance, O(3) invariance, periodic translation invariance and lattice permutation invariance. Detailed definitions are provided as follows.

Definition 4.1 (Composition Permutation Invariance)**.** For any permutation $\boldsymbol{P} \in \mathrm{S}_n$, $p(\boldsymbol{C}, \boldsymbol{F} \mid \boldsymbol{A}) = p(\boldsymbol{C}, \boldsymbol{F}\boldsymbol{P} \mid \boldsymbol{A}\boldsymbol{P})$, *i.e.*, changing the order of atoms will not change the distribution, where S_n represents the set of permutation matrices with dimensions $n \times n$.

Definition 4.2 (O(3) Invariance)**.** Given an transformation matrix $\boldsymbol{Q} \in \mathbb{R}^{3\times 3}$ where $\boldsymbol{Q}$ is any O(3) group element operated on $\boldsymbol{L}$, the condition $p(\boldsymbol{C}(\boldsymbol{Q}\boldsymbol{L}), \boldsymbol{F} \mid \boldsymbol{A}) = p(\boldsymbol{C}(\boldsymbol{L}), \boldsymbol{F} \mid \boldsymbol{A})$ holds, indicating that the distribution remains invariant under any rotation or reflection applied to $\boldsymbol{L}$, where $\boldsymbol{C}(\cdot)$ is the function that translates a lattice matrix to lattice parameters.

Definition 4.3 (Lattice Permutation Invariance)**.** For any permutation $\boldsymbol{P} \in \mathrm{S}_3$, $p(\boldsymbol{C}, \boldsymbol{F} \mid \boldsymbol{A}) = p(\boldsymbol{P}\boldsymbol{C}, \boldsymbol{P}\boldsymbol{F} \mid \boldsymbol{A})$, *i.e.*, changing the lattice base order will not change the distribution.

Definition 4.4 (Periodic Translation Invariance)**.** For any translation $\mathbf{t} \in \mathbb{R}^{3\times 1}$, $p(\boldsymbol{C}, w(\boldsymbol{F} + \mathbf{t}\mathbf{1}^\top) \mid \boldsymbol{A}) = p(\boldsymbol{C}, \boldsymbol{F} \mid \boldsymbol{A})$, where the function $w(\boldsymbol{F}) = \boldsymbol{F} - \lfloor \boldsymbol{F} \rfloor \in [0, 1)^{3\times n}$ returns the fractional part of each element in $\boldsymbol{F}$, and $\mathbf{1} \in \mathbb{R}^{3\times 1}$ is a vector with all elements set to one. It explains that any periodic translation of $\boldsymbol{F}$ will not change the distribution.

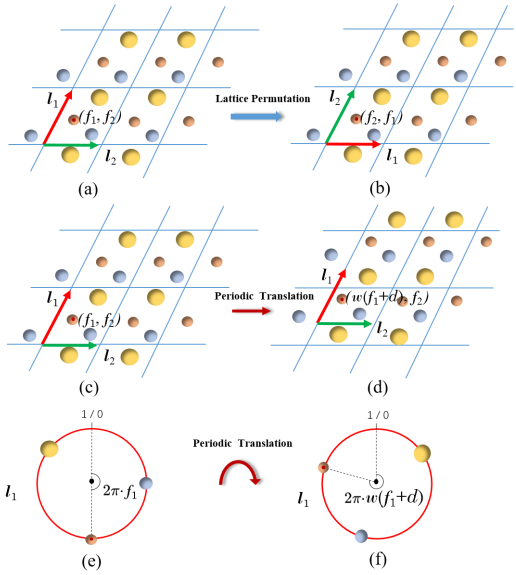

Figure 1. (a)→(b): The lattice permutation of the lattice bases $\boldsymbol{l}_1, \boldsymbol{l}_2$. (c)→(d): The periodic translation of the fractional coordinates $\boldsymbol{f}_1, \boldsymbol{f}_2$. (e)→(f): The schematic diagram of the period translation represented as points on a circle. Both cases do not change the crystal structure. Here, the 2D crystal is used for better illustration.

Composition permutation invariance in generation is effectively achieved by GNNs as the foundational architecture (Kipf & Welling, 2016). According to previous work (Jiao et al., 2023), employing the fractional system handles the O(3) invariance of crystals by ensuring O(3) invariance with respect to orthogonal transformations on the lattice matrix. Previous work (Luo et al., 2023) further address the O(3) invariance of the lattice matrix by substituting it with lattice parameters $\boldsymbol{C}$, as $\boldsymbol{C}(\boldsymbol{Q}\boldsymbol{L}) = \boldsymbol{C}(\boldsymbol{L})$ always holds for arbitrary $\boldsymbol{Q} \in \mathrm{O}(3)$. Consequently, our representation of crystals using both the fractional system and lattice parameters naturally satisfies O(3) invariance. Thus, we mainly focus on the lattice permutation and periodic translation invarance as shown in Figure 1. For better demonstration, we utilize representation method in (Mardia et al., 2000) to show the periodic translation invariance. For details on the representation of lattice bases as circles in Figure 1 (e) and (f), see Appendix B.1.

Comparing with other symmetry awareness generation method. We notice that previous approaches (Xie et al., 2021; Luo et al., 2023; Jiao et al., 2023) ignore the lattice permutation invariance, both for CSP task and for ab initio generation task. The ab initio generation method SymMat (Luo et al., 2023) directly generates lattice parameters $\boldsymbol{C}$ using a variational autoencoders from rand noise $\boldsymbol{\epsilon}$.

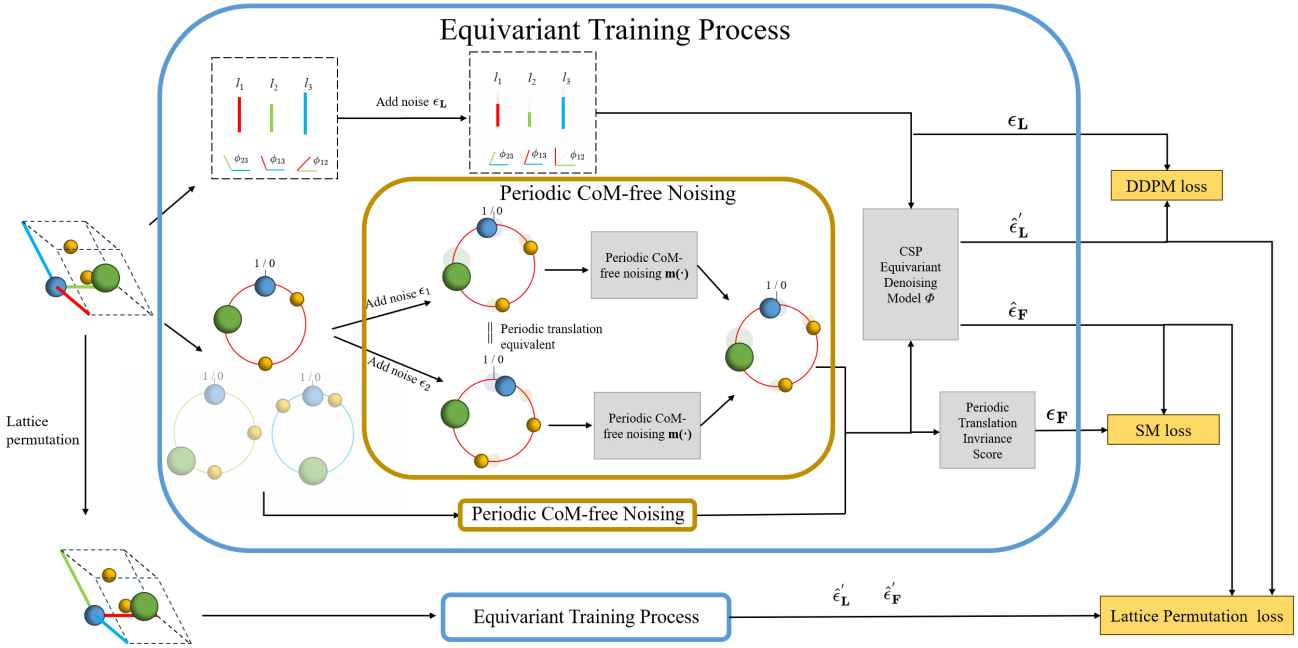

Figure 2. Overview of training process in EquiCSP.

However, it doesn't guarantee that the marginal distribution satisfies $p(\boldsymbol{C}) = p(\boldsymbol{PC})$ for any $\boldsymbol{P} \in \mathrm{S}_3$, which means that it doesn't guarantee the lattice permutation invariance. DiffCSP (Jiao et al., 2023) generates lattice matrix by diffusion model, however, as discussed in Section 4.3, its diffusion method lacks lattice permutation equivariance, impacting the final lattice distribution not invariant. Our method is the first to realize this symmetry of crystal structure, and we will ablate the benefit in Section 5.2.

4.2. An Overview of EquiCSP

In our work, we implement EquiCSP by concurrently diffusing the $\boldsymbol{C}$ and $\boldsymbol{F}$ within the framework of DiffCSP (Jiao et al., 2023). For a given atomic composition $\boldsymbol{A}$, the intermediate states of $\boldsymbol{C}$ and $\boldsymbol{F}$ at any time step t (where $0 \leq t \leq T$) are represented by $\mathcal{M}_t$. EquiCSP orchestrates two distinct Markov processes: a forward diffusion that incrementally introduces noise into $\mathcal{M}_0$, and a backward generation process that strategically samples from the prior distribution $\mathcal{M}_T$ to reconstruct the initial data $\mathcal{M}_0$. The implementation specifics are summarized in Algorithms 1 and 2.

In light of the symmetry discussed in Section 4.1, the distribution restored from $\mathcal{M}_T$ must meet invariance. This requirement is achieved if the prior distribution $p(\mathcal{M}_T)$ exhibits invariance and the Markov transition $p(\mathcal{M}_{t-1}|\mathcal{M}_t)$ is equivariant, as established in previous literature (Xu et al., 2021). An equivariant transition implies $p(g \cdot \mathcal{M}_{t-1}|g \cdot \mathcal{M}_t) = p(\mathcal{M}_{t-1}|\mathcal{M}_t)$ for any transformation g acting on $\mathcal{M}$, as defined in Definitions 4.3-4.4. Further explanations on how diffusion processes are applied to $\boldsymbol{C}$ and $\boldsymbol{F}$ are detailed subsequently.

4.3. Diffusion on Lattice Parameters

Given that $\boldsymbol{C}$ is a continuous variable with lattice lengths $\boldsymbol{l} > 0$ and lattice angles $\boldsymbol{\phi} \in (0, \pi)^3$, we exploit Denoising Diffusion Probabilistic Model (DDPM) (Ho et al., 2020) with prepossessing of $\boldsymbol{C}$ to accomplish the generation. As detailed in Appendix B.2, such preprocessing projects the definition domain of $\boldsymbol{C}$ onto $\mathbb{R}^{3\times 2}$, and hereinafter the notation $\boldsymbol{C}$ refers to the projected lattice parameters.

4.3.1. Generation

We define the generation process that progressively diffuses the Normal prior $p(\boldsymbol{C}_T)$ towards stable crystal lattice distribution $p(\boldsymbol{C}_0)$ by:

$$p(\boldsymbol{C}_{t-1}|\mathcal{M}_t) = \mathcal{N}(\boldsymbol{C}_{t-1}|\mu(\mathcal{M}_t), \sigma^2(\mathcal{M}_t)\boldsymbol{I}), \quad (2)$$

where $\mu(\mathcal{M}_t) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{C}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}}(\mathcal{M}_t, t)\right), \sigma^2(\mathcal{M}_t) = \beta_t \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}$. The denoising term $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}}(\mathcal{M}_t, t) \in \mathbb{R}^{3\times 2}$ is predicted by the neural network model $\phi(\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}, t)$ detailed in Section 4.5.

As the prior distribution $p(\boldsymbol{C}_T) = \mathcal{N}(0, \boldsymbol{I})$ is already lattice permutation invariant, we require the generation process in Eq. (2) to be lattice permutation equivariant, which is formally stated below, and give a proof in Appendix A.1.

Proposition 4.5. *The marginal distribution $p(\boldsymbol{C}_0)$ by Eq. (2) is lattice permutation invariant if $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}}(\mathcal{M}_t, t)$ is lattice permutation equivariant, namely $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}}(\boldsymbol{PC}_t, \boldsymbol{PF}_t, \boldsymbol{A}, t) = \boldsymbol{P}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}}(\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}, t), \forall \boldsymbol{P} \in \mathrm{S}_3$.*

4.3.2. Training

We define the forward process as one that gradually diffuses $\boldsymbol{C}_0$ towards a Normal prior, represented by $p(\boldsymbol{C}_T) = \mathcal{N}(0, \boldsymbol{I})$. This process is defined through the conditional probability $q(\boldsymbol{C}_t|\boldsymbol{C}_{t-1})$, which is formulated based on the initial distribution:

$$q(\boldsymbol{C}_t|\boldsymbol{C}_0) = \mathcal{N}\Big(\boldsymbol{C}_t|\sqrt{\bar{\alpha}_t}\boldsymbol{C}_0, (1-\bar{\alpha}_t)\boldsymbol{I}\Big), \tag{3}$$

where $\beta_t \in (0,1)$ controls the variance, and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_t = \prod_{s=1}^{t}(1-\beta_t)$ is valued in accordance to the cosine scheduler (Nichol & Dhariwal, 2021).

To train the denoising model ϕ, we initiate by sampling $\boldsymbol{\epsilon_L} \sim \mathcal{N}(0, \boldsymbol{I})$ and reparameterize $\boldsymbol{C}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{C}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon_L}$ based on Eq. (3). The training goal is then established by minimizing the ℓ_2 loss between $\boldsymbol{\epsilon_L}$ and its estimate $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}}$:

$$\mathcal{L}_{\boldsymbol{C}} = \mathbb{E}_{\boldsymbol{\epsilon_L}\sim\mathcal{N}(0,\boldsymbol{I}),t\sim\mathcal{U}(1,T)}[\|\boldsymbol{\epsilon_L} - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}}(\mathcal{M}_t, t)\|_2^2]. \tag{4}$$

To satisfy proposition 4.5, we introduce an additional loss value as a penalty term during the training process, detailed as follows:

$$\mathcal{L}_{p\boldsymbol{C}} = \mathbb{E}[\|\hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}}(\boldsymbol{P}\boldsymbol{C}_t, \boldsymbol{P}\boldsymbol{F}_t, \boldsymbol{A}, t) - \boldsymbol{P}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}}(\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}, t)\|_2^2], \tag{5}$$

where the expectation is taken with respect $t \sim \mathcal{U}(1,T)$ and $\boldsymbol{P} \sim \mathcal{U}(\mathrm{S}_3)$. After sufficient training, the generation process will satisfy lattice permutation invariance as stated in Proposition 4.5. Our ablation experiments demonstrate the significant performance of this method, and the learning curve in Appendix D shows that the difficulty of learning is greatly reduced compared with DiffCSP (Jiao et al., 2023).

Comparing with the Method of Encoding the Equivariance to Denoising Model. We discover that the Frame Average (FA) method (Puny et al., 2021), employing a unified, hard-constraint approach for equivariant neural networks, also satisfies Proposition 4.5. We provide implementation details in Appendix B.3. However, our experimental findings in Section 5.2 reveal that the computational burden of finite group operations required by FA renders it impractical for iterative models like diffusion models. In contrast, our method significantly enhances computational efficiency and accuracy by simply incorporating additional loss values during training.

4.4. Diffusion on Fractional Coordinates

Combining Score-Matching (SM) based framework with Wrapped Normal (WN) distribution (SMWN), as proposed in (Jiao et al., 2023), for generating fractional coordinates proves advantageous due to the periodicity and [0,1) constraint of these coordinates. Based on SMWN method, we propose an innovative noising algorithm to meet periodic translation equivariance, detailed as follows:

4.4.1. Generation

In the generation process, we first initialize $\boldsymbol{F}_T$ from the uniform distribution $\mathcal{U}(0,1)$, which is periodic translation invariant. With the denoising term $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{F}}(\mathcal{M}_t, t)$ predicted by $\phi(\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}, t)$ to model the data score $\nabla_{\boldsymbol{F}_t} \log p(\boldsymbol{F}_t)$, we combine the ancestral predictor with the Langevin corrector used in DiffCSP (Jiao et al., 2023) to sample $\boldsymbol{F}_0$. Specifically, this method can be simply viewed as progressively sampling from the wrapped normal distribution $p(\boldsymbol{F}_{t-1}|\mathcal{M}_t)$ at each time step t, where the mean of the wrapped normal is a function of $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{F}}$, with the detailed formula provided in Eq. (29) of the Appendix A.2. To ensure that $p(\boldsymbol{F}_{t-1}|\mathcal{M}_t)$ satisfies periodic translation equivariance, in accordance with (Jiao et al., 2023), $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{F}}$ must meet the periodic translation invariance:

$$\hat{\boldsymbol{\epsilon}}_{\boldsymbol{F}}(\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}, t) = \hat{\boldsymbol{\epsilon}}_{\boldsymbol{F}}(\boldsymbol{C}_t, w(\boldsymbol{F}_t + \mathbf{t}\mathbf{1}^\top), \boldsymbol{A}, t), \tag{6}$$

where $\forall \mathbf{t} \in \mathbb{R}^3$ and the truncation function $w(\cdot)$ is already defined in Definition 4.4. We will ensure that the model output conforms to this property in Section 4.5 to guarantee the equivariance of generation.

Similarly, the data score $\nabla_{\boldsymbol{F}_t} \log p(\boldsymbol{F}_t)$ must adhere to periodic translation invariance. The accurate estimation of this score, set as the training target for $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{F}}(\mathcal{M}_t, t)$, represents the primary challenge we will tackle in Section 4.4.2.

In addition, we require the generation process to be lattice permutation equivariant, which is formally stated below, provided a proof in Appendix A.2:

Proposition 4.6. *The marginal distribution $p(\boldsymbol{F}_0)$ is lattice permutation invariant if $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{F}}(\mathcal{M}_t, t)$ is lattice permutation equivariant, namely $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{F}}(\boldsymbol{P}\boldsymbol{C}_t, \boldsymbol{P}\boldsymbol{F}_t, \boldsymbol{A}, t) = \boldsymbol{P}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{F}}(\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}, t), \forall \boldsymbol{P} \in \mathrm{S}_3$.*

4.4.2. Training

During the forward process, SMWN samples each column of $\boldsymbol{\epsilon} \in \mathbb{R}^{3\times n}$ from wrapped normal distribution $\mathcal{N}_w(0, \sigma_t \boldsymbol{I})$, and then acquire $\boldsymbol{F}_t = w(\boldsymbol{F}_0 + \boldsymbol{\epsilon})$, where $\mathcal{N}_w(0, \sigma_t^2 \boldsymbol{I})$ denotes the probability density function(PDF) of WN distribution with mean 0, variance σ_t^2 and period 1, σ_t is the noise magnitude level and $\sigma_1 < \sigma_2 < \ldots \sigma_T$. According to the feature of WN, if σ_T is sufficiently large, $p(\boldsymbol{F}_T)$ approaches a uniform distribution $\mathcal{U}(0,1)$ which is desirable for generation. Our training target is:

$$\hat{\boldsymbol{\epsilon}}_{\boldsymbol{F}}(\mathcal{M}_t, t) \to \nabla_{\boldsymbol{F}_t} \log q(\boldsymbol{F}_t). \tag{7}$$

The pivotal challenge is how to accurately obtain the score matrix $\nabla_{\boldsymbol{F}_t} \log q(\boldsymbol{F}_t)$ to maintain the periodic translation invariance, a feature not guaranteed by the conventional diffusion framework. For instance, DiffCSP (Jiao et al., 2023)

employs the ordinary denoising score matching (Vincent, 2011) training objective to estimate the score:

$$\mathcal{L}_{\boldsymbol{F}} = \mathbb{E}_{\boldsymbol{F}_0 \sim q(\boldsymbol{F}_0), \boldsymbol{F}_t \sim q(\boldsymbol{F}_t|\boldsymbol{F}_0), t \sim \mathcal{U}(1,T)} \left[\lambda_t \|\nabla_{\boldsymbol{F}_t} \log q(\boldsymbol{F}_t|\boldsymbol{F}_0) - \tilde{\boldsymbol{S}}\|_2^2\right], \tag{8}$$

where $\tilde{\boldsymbol{S}}$ is the estimate of $\nabla_{\boldsymbol{F}_t} \log q(\boldsymbol{F}_t)$ and λ_t is the weight of loss. The core issue arises from the dataset distribution $q(\boldsymbol{F}_0)$, which typically does not exhibit the same periodic translation invariance as the ground truth distribution $p(\boldsymbol{F}_0)$ because the dataset usually does not contain all the samples $w(\boldsymbol{F}_0 + \mathbf{t}\mathbf{1}^\top)$. Consequently, $\tilde{\boldsymbol{S}}$ cannot be guaranteed to be periodic translation invariant. For illustration, consider a dataset with only one sample $\tilde{\boldsymbol{F}}_0$, the estimate of score will be:

$$\begin{aligned} \tilde{\boldsymbol{S}} &= \nabla_{\boldsymbol{F}_t} \log q(\boldsymbol{F}_t|\boldsymbol{F}_0 = \tilde{\boldsymbol{F}}_0) \\ &= \nabla_{\boldsymbol{F}_t} \log \mathcal{N}_w(\boldsymbol{F}_t|\tilde{\boldsymbol{F}}_0, \sigma_t^2 \boldsymbol{I}) \\ &= \nabla_{\boldsymbol{\epsilon}} \log \mathcal{N}_w(\boldsymbol{\epsilon}|0, \sigma_t^2 \boldsymbol{I}), \end{aligned} \tag{9}$$

and obviously

$$\begin{aligned} &\nabla_{\boldsymbol{\epsilon}} \log \mathcal{N}_w(\boldsymbol{\epsilon}|0, \sigma_t^2 \boldsymbol{I}) \\ &\neq \nabla_{w(\boldsymbol{\epsilon}+\mathbf{t}\mathbf{1}^\top)} \log \mathcal{N}_w\big(w(\boldsymbol{\epsilon} + \mathbf{t}\mathbf{1}^\top)|0, \sigma_t^2 \boldsymbol{I}\big), \end{aligned} \tag{10}$$

which means the predicted score not invariant. Numerous studies (Luo et al., 2023; 2021; Niu et al., 2020; Jin et al., 2023) have also indicated that for ensuring equivariance, the score matrix should be determined more cautiously.

A potential solution to this issue is to augment the dataset using periodic translation operations to better align $q(\boldsymbol{F}_0)$ with the invariant distribution $p(\boldsymbol{F}_0)$. However, this approach demands a significant amount of training time due to the periodic translation group being a Lie group with infinitely many elements.

We propose 'Periodic CoM-free Noising', a new noising method that ensures the noise added to $q(\boldsymbol{F}_0)$ results in periodic translation invariant score as closely aligned as possible to the score achieved by the original noise added to $p(\boldsymbol{F}_0)$. The method is based on the following statements: $\nabla_{\boldsymbol{F}_t} \log q(\boldsymbol{F}_t)$ is periodic translation invariant if $\nabla_{\boldsymbol{F}_t} \log q(\boldsymbol{F}_t|\boldsymbol{F}_0)$ is periodic translation invariant:

$$\begin{aligned} &\nabla_{\boldsymbol{F}_t} \log q(\boldsymbol{F}_t|\boldsymbol{F}_0) \\ &= \nabla_{w(\boldsymbol{F}_t+\mathbf{t}\mathbf{1}^\top)} \log q(w(\boldsymbol{F}_t + \mathbf{t}\mathbf{1}^\top)|\boldsymbol{F}_0), \end{aligned} \tag{11}$$

where $\forall \mathbf{t} \in \mathbb{R}^3$.

The noising method is equivalent to operating on $\nabla_{\boldsymbol{F}_t} \log q(\boldsymbol{F}_t|\boldsymbol{F}_0)$. Therefore, we first focus on achieving $\nabla_{\boldsymbol{F}_t} \log q(\boldsymbol{F}_t|\boldsymbol{F}_0)$ that meets the periodic translation invariance, followed by adjusting the score numerically for more accurate training results.

Periodic CoM-free Noising. In order to satisfy Eq.(11), we adopt a parameterization scheme for $\nabla_{\boldsymbol{F}_t} \log q(\boldsymbol{F}_t|\boldsymbol{F}_0)$ as follows:

$$\begin{aligned} \nabla_{\boldsymbol{F}_t} \log q(\boldsymbol{F}_t|\boldsymbol{F}_0) &= \nabla_{\bar{\boldsymbol{F}}} \log \mathcal{N}_w(\bar{\boldsymbol{F}}|\boldsymbol{F}_0, \sigma_t^2 \mathbf{I}) \\ &= \nabla_{\bar{\boldsymbol{\epsilon}}} \log \mathcal{N}_w(\bar{\boldsymbol{\epsilon}}|0, \sigma_t^2 \mathbf{I}), \end{aligned} \tag{12}$$

where

$$\bar{\boldsymbol{F}} = w(\boldsymbol{F}_0 + \bar{\boldsymbol{\epsilon}}), \tag{13}$$

$$\bar{\boldsymbol{\epsilon}} = \boldsymbol{m}(\boldsymbol{\epsilon}) = \boldsymbol{m}(w(\boldsymbol{\epsilon} + \mathbf{t}\mathbf{1}^\top)), \forall \mathbf{t} \in \mathbb{R}^3. \tag{14}$$

Here, we introduce a noise conversion function $\boldsymbol{m}(\cdot)$ to map all the fractional coordinate matrices that are periodic translation equivalent with $\boldsymbol{F}_t$ to a unique matrix $\bar{\boldsymbol{F}}$. This addresses the requirement of periodic translation invariance of score. Consequently, we can employ the ordinary score calculation method, specifically the score of anisotropic WN here, to compute $\nabla_{\bar{\boldsymbol{F}}} \log q(\bar{\boldsymbol{F}}|\boldsymbol{F}_0)$ as a substitute for the required score.

The key of Periodic CoM-free Noising is to design the specific function $\boldsymbol{m} : w(\boldsymbol{\epsilon} + \mathbf{t}\mathbf{1}^\top) \to \bar{\boldsymbol{\epsilon}}$. We note that the CoM-free systems in molecular conformation generation (Xu et al., 2022) solve similar problem in translation invariance. However, the Center of Mass(CoM) of periodic data cannot be simple computed as mean value of data (Bai & Breen, 2008). Similar to the idea of CoM-free systems, we utilize the concept of "mean angle" from (Mardia et al., 2000) to construct $\boldsymbol{m}(\cdot)$ as a periodic CoM-free function. Specifically, we denote $\boldsymbol{\epsilon} = [\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2, \ldots, \boldsymbol{\epsilon}_n]$ and fomulate:

$$\begin{cases} \boldsymbol{m}(\boldsymbol{\epsilon}) = w\big(\boldsymbol{\epsilon} - \dfrac{\operatorname{atan2}(\bar{\boldsymbol{y}}(\boldsymbol{\epsilon}), \bar{\boldsymbol{x}}(\boldsymbol{\epsilon}))}{2\pi} \mathbf{1}^\top\big), \\ \bar{\boldsymbol{y}}(\boldsymbol{\epsilon}) = \dfrac{1}{n} \displaystyle\sum_{i=0}^{n} \sin(2\pi \boldsymbol{\epsilon}_i), \\ \bar{\boldsymbol{x}}(\boldsymbol{\epsilon}) = \dfrac{1}{n} \displaystyle\sum_{j=0}^{n} \cos(2\pi \boldsymbol{\epsilon}_i). \end{cases} \tag{15}$$

Intuitively, as shown in Figure 3, the function transform periodic data of each lattice axis to angle data on a circle, and then subtract all the data by the periodic CoM. Consequently, it maps all equivalent periodic data to the same representation, and addresses the periodic translation invariance. We provide a proof in Appendix A.3.

After implementing the algorithm, we substituted the $\nabla_{\boldsymbol{F}_t} \log q(\boldsymbol{F}_t|\boldsymbol{F}_0)$ in the denoising score matching training objective, as defined in Eq.(8), with Eq.(12). This change led to significant performance improvements in our ablation study and excellent training convergence demonstrated in Appendix D. However, with $\bar{\boldsymbol{F}} = [\bar{\boldsymbol{f}}_1, \bar{\boldsymbol{f}}_2, \ldots, \bar{\boldsymbol{f}}_n]$ and $\boldsymbol{F}_t = [\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_n]$, we identified two points that still require enhancement: **1.** The marginal distribution[2] $q(\bar{\boldsymbol{f}}_i|\boldsymbol{F}_0)$

[2]We focus on evaluating the score of marginal distribution

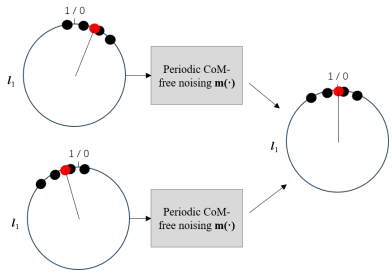

Figure 3. The illustration of periodic translation invariance with periodic CoM-free function.

does not simply satisfy $\mathcal{N}_w(0, \sigma_t^2 \boldsymbol{I})$, so it is necessary to re-evaluate its distribution to recalculate $\nabla_{\bar{\boldsymbol{F}}} \log q(\bar{\boldsymbol{F}}|\boldsymbol{F}_0)$. **2.** The generation process is designed for non periodic CoM-free system, while our method now simply use periodic CoM-free score $\nabla_{\bar{\boldsymbol{F}}} \log q(\bar{\boldsymbol{F}})$ to replace corresponding $\nabla_{\boldsymbol{F}_t} \log q(\boldsymbol{F}_t)$. A more rigorous implementation of probabilistic modeling is warranted to establish a refined connection between the score. We provide solutions below.

Von Mises Simulation. We denote $\bar{\boldsymbol{\epsilon}} = [\bar{\boldsymbol{\epsilon}}_1, \bar{\boldsymbol{\epsilon}}_2, \dots, \bar{\boldsymbol{\epsilon}}_n]$. To evaluate the probability density function (PDF) $q(\bar{\boldsymbol{f}}_i|\boldsymbol{F}_0)$, we recognize that since $\bar{\boldsymbol{F}} = w(\boldsymbol{F}_0 + \bar{\boldsymbol{\epsilon}})$, the task can be reframed as estimating the marginal PDF of $\bar{\boldsymbol{\epsilon}} = \boldsymbol{m}(\boldsymbol{\epsilon})$, specifically $p(\bar{\boldsymbol{\epsilon}}_i)$, where $\boldsymbol{\epsilon} \sim \mathcal{N}_w(0, \sigma_t^2 \boldsymbol{I})$. However, directly obtaining the formula of $p(\bar{\boldsymbol{\epsilon}}_i)$ is challenging due to the complexity of $\boldsymbol{m}(\cdot)$ and WN. For simplicity, we utilize the Von Mises distribution (Gatto & Jammalamadaka, 2007) which is commonly used in dealing with circular distribution problems to simulate the $p(\bar{\boldsymbol{\epsilon}}_i)$, and use the Monte Carlo method to obtain its parameters. More details is list in Appendix C.1. Consequently, $\nabla_{\bar{\boldsymbol{F}}} \log q(\bar{\boldsymbol{F}}|\boldsymbol{F}_0) = [\bar{\boldsymbol{c}}_1, \bar{\boldsymbol{c}}_2 \dots \bar{\boldsymbol{c}}_n]$ can be expressed using the formula of score of Von Mises:

$$\begin{aligned} \bar{\boldsymbol{c}}_i &= \nabla_{\bar{\boldsymbol{f}}_i} \log q(\bar{\boldsymbol{f}}_i|\boldsymbol{F}_0) = \nabla_{\bar{\boldsymbol{\epsilon}}_i} \log p(\bar{\boldsymbol{\epsilon}}_i) \\ &= -2\pi \cdot \kappa(n, \sigma_t) \cdot \sin(2\pi \bar{\boldsymbol{\epsilon}}_i), \end{aligned} \tag{16}$$

where $\kappa(n, \sigma_t)$ is the parameter of Von Mises distribution obtained by Monte Carlo method. Consequently, using the revised $\nabla_{\bar{\boldsymbol{F}}} \log q(\bar{\boldsymbol{F}}|\boldsymbol{F}_0)$ allows for a more accurate estimation of the invariant score $\nabla_{\bar{\boldsymbol{F}}} \log q(\bar{\boldsymbol{F}}) = [\bar{\boldsymbol{s}}_1, \bar{\boldsymbol{s}}_2, \dots \bar{\boldsymbol{s}}_n]$.

Probabilistic Modeling Process. For the second point, we propose a novel probabilistic modeling process inspired by (Luo et al., 2023). Denoting the score of non periodic CoM-free system as $\nabla_{\boldsymbol{F}_t} \log q(\boldsymbol{F}_t) = [\boldsymbol{s}_1, \boldsymbol{s}_2 \dots \boldsymbol{s}_n]$, we consider $\boldsymbol{s}_i$ as a function of all the corresponding CoM-free data namely $\{\bar{\boldsymbol{f}}_1, \bar{\boldsymbol{f}}_2, \dots, \bar{\boldsymbol{f}}_n\}$ aiming to address periodic translation invariance. From the chain rule of derivatives, we can approximate the score $\nabla_{\boldsymbol{F}_t} \log q(\boldsymbol{F}_t)$ by the score of CoM-free system namely $\nabla_{\bar{\boldsymbol{F}}} \log q(\bar{\boldsymbol{F}})$:

$$\begin{aligned} \boldsymbol{s}_j &= \sum_{i=0}^{n} \nabla_{\bar{\boldsymbol{f}}_i} \log q(\bar{\boldsymbol{f}}_i) \cdot \nabla_{\boldsymbol{f}_j} \bar{\boldsymbol{f}}_i, \\ &= \sum_{i=0}^{n} \bar{\boldsymbol{s}}_i \cdot \nabla_{\boldsymbol{f}_j} \bar{\boldsymbol{f}}_i. \end{aligned} \tag{17}$$

We previously established that $\bar{\boldsymbol{s}}_i$ is invariant, but $\nabla_{\boldsymbol{f}_j} \bar{\boldsymbol{f}}_i$ might destroy the periodic translation invariance. Fortunately, we can first transform $\nabla_{\boldsymbol{f}_j} \bar{\boldsymbol{f}}_i$ to the j-colomn of $\nabla_{\boldsymbol{\epsilon}} \bar{\boldsymbol{\epsilon}}_i$ by their definition, and then strictly prove the following statements in Appendix A.4:

Proposition 4.7. *$\nabla_{\boldsymbol{\epsilon}} \bar{\boldsymbol{\epsilon}}_i = \nabla_{\boldsymbol{\epsilon}} \big(\boldsymbol{m}(\boldsymbol{\epsilon})[:, i]\big)$ is periodic translation invariance, where $\boldsymbol{m}(\boldsymbol{\epsilon})[:, i]$ is the i-th column of $\boldsymbol{m}(\boldsymbol{\epsilon})$. In other words, $\nabla_{\boldsymbol{\epsilon}} \big(\boldsymbol{m}(\boldsymbol{\epsilon})[:, i]\big) = \nabla_{w(\boldsymbol{\epsilon} + \mathbf{t}\mathbf{1}^\top)} \big(\boldsymbol{m}(w(\boldsymbol{\epsilon} + \mathbf{t}\mathbf{1}^\top))[:, i]\big)$. Thus $\nabla_{\boldsymbol{F}_t} \log q(\boldsymbol{F}_t)$ by Eq.(17) is periodic translation invariance.*

While $\boldsymbol{m}(\bar{\boldsymbol{\epsilon}}) = \boldsymbol{m}(\boldsymbol{\epsilon})$ holds, we can reformulate Eq.(17) using Proposition 4.7 as:

$$\boldsymbol{s}_j = \sum_{i=0}^{n} \bar{\boldsymbol{s}}_i \cdot \nabla_{\bar{\boldsymbol{\epsilon}}_j} \big(\boldsymbol{m}(\bar{\boldsymbol{\epsilon}})[:, i]\big), \tag{18}$$

where $\nabla_{\bar{\boldsymbol{\epsilon}}_j} \big(\boldsymbol{m}(\bar{\boldsymbol{\epsilon}})[:, i]\big)$ is the j-column of $\nabla_{\bar{\boldsymbol{\epsilon}}} \big(\boldsymbol{m}(\bar{\boldsymbol{\epsilon}})[:, i]\big)$. This indicates that to compute the adjusted score $\nabla_{\boldsymbol{F}_t} \log q(\boldsymbol{F}_t)$, an additional invariant parameter $\bar{\boldsymbol{\epsilon}}$ is required. This parameter can be predicted by the model ϕ.

Put things together. Finally, we consolidate our approach to construct the training objective to approximate $\nabla_{\bar{\boldsymbol{F}}} \log q(\bar{\boldsymbol{F}})$ and the expected $\bar{\boldsymbol{\epsilon}}$:

$$\begin{aligned} \mathcal{L}_s = \mathbb{E}_{\boldsymbol{F}_0 \sim q(\boldsymbol{F}_0), \boldsymbol{\epsilon} \sim \mathcal{N}_w(0, \sigma_t \boldsymbol{I}), t \sim \mathcal{U}(1, T)} \\ [\|\nabla_{\bar{\boldsymbol{F}}} \log q(\bar{\boldsymbol{F}}|\boldsymbol{F}_0) - s_\theta(\mathcal{M}_t, t)\|_2^2], \end{aligned} \tag{19}$$

$$\begin{aligned} \mathcal{L}_F = \mathbb{E}_{\boldsymbol{F}_0 \sim q(\boldsymbol{F}_0), \boldsymbol{\epsilon} \sim \mathcal{N}_w(0, \sigma_t \boldsymbol{I}), t \sim \mathcal{U}(1, T)} \\ [\|\bar{\boldsymbol{\epsilon}} - F_\theta(\mathcal{M}_t, t)\|_2^2], \end{aligned} \tag{20}$$

where $s_\theta(\mathcal{M}_t, t)$ and $F_\theta(\mathcal{M}_t, t)$ are directly predicted by the model ϕ, and $\nabla_{\bar{\boldsymbol{F}}} \log q(\bar{\boldsymbol{F}}|\boldsymbol{F}_0)$ is calculated by Eq.(16), meanwhile $\bar{\boldsymbol{\epsilon}} = \boldsymbol{m}(\boldsymbol{\epsilon})$. And we can finally derive $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{F}}$ in Section 4.4.1 by replacing $\nabla_{\bar{\boldsymbol{F}}} \log q(\bar{\boldsymbol{F}})$ with $s_\theta(\mathcal{M}_t, t)$ and $\bar{\boldsymbol{\epsilon}}$ with $F_\theta(\mathcal{M}_t, t)$ in Eq.(18).

In addition, to satisfy Proposition 4.6, we design the permutation loss similar to the method in Section 4.3:

$$\mathcal{L}_{ps} = \mathbb{E}[\|s_\theta(\boldsymbol{P}\boldsymbol{C}_t, \boldsymbol{P}\boldsymbol{F}_t, \boldsymbol{A}, t) - \boldsymbol{P} s_\theta(\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}, t)\|_2^2], \tag{21}$$

$$\mathcal{L}_{pF} = \mathbb{E}[\|F_\theta(\boldsymbol{P}\boldsymbol{C}_t, \boldsymbol{P}\boldsymbol{F}_t, \boldsymbol{A}, t) - \boldsymbol{P} F_\theta(\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}, t)\|_2^2], \tag{22}$$

where the expectation is taken with respect $t \sim \mathcal{U}(1, T)$ and $\boldsymbol{P} \sim \mathcal{U}(\mathrm{S}_3)$.

instead of the joint distribution to better align with the origin noise addition techniques employed in SDEs.

Table 1. Results on stable structure prediction task. The results of baseline methods are from Jiao (Jiao et al., 2023)

	Perov-5		MP-20		MPTS-52	
	Match rate↑	RMSE↓	Match rate↑	RMSE↓	Match rate↑	RMSE↓
RS	36.56	0.0886	11.49	0.2822	2.68	0.3444
BO	55.09	0.2037	12.68	0.2816	6.69	0.3444
PSO	21.88	0.0844	4.35	0.1670	1.09	0.2390
P-cG-SchNet	48.22	0.4179	15.39	0.3762	3.67	0.4115
CDVAE	45.31	0.1138	33.90	0.1045	5.34	0.2106
DiffCSP	52.02	0.0760	51.49	0.0631	12.19	0.1786
EquiCSP	**52.02**	**0.0707**	**57.59**	**0.0510**	**14.85**	**0.1169**

4.5. The Architecture of the Denoising Model

In this subsection, we outline the specific design of the denoising model $\phi(\mathcal{M}_t)$, focusing on how it computes the three denoising terms: $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}}, s_\theta, F_\theta$. For simplicity, the subscript t is omitted in this discussion.

The model begins by integrating the atom embeddings $f_{\text{atom}}(\boldsymbol{A})$ with sinusoidal time embeddings $f_{\text{time}}(t)$ to generate the initial node features $\boldsymbol{H} = \varphi_{\text{in}}(f_{\text{atom}}(\boldsymbol{A}), f_{\text{time}}(t))$. We then describe the message passing mechanism from node j to node i in the l-th layer of the network.

$$\boldsymbol{m}_{ij}^{(l)} = \varphi_m(\boldsymbol{h}_i^{(l-1)}, \boldsymbol{h}_j^{(l-1)}, \boldsymbol{C}, \psi_{\text{FT}}(\boldsymbol{f}_j - \boldsymbol{f}_i)) \tag{23}$$

$$\boldsymbol{h}_i^{(l)} = \boldsymbol{h}_i^{(l-1)} + \varphi_h(\boldsymbol{h}_i^{(l-1)}, \sum_{j=1}^{n} \boldsymbol{m}_{ij}^{(l)}), \tag{24}$$

where φ_m and φ_h are MLPs, and The function $\psi_{\text{FT}} : (-1,1)^3 \to [-1,1]^{3\times K}$ is Fourier Transformation of the relative fractional coordinate $\boldsymbol{f}_j - \boldsymbol{f}_i$ to address periodic translation invariance according to (Jiao et al., 2023).

After S layers of message passing, we get the graph-level denoising term as:

$$\hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}} = \varphi_{\boldsymbol{L}}\Big(\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{h}_i^{(S)}\Big), \tag{25}$$

and the node-level denoising terms as

$$s_\theta[:,i], F_\theta[:,i] = \varphi_s(\boldsymbol{h}_i^{(S)}), \varphi_F(\boldsymbol{h}_i^{(S)}) \tag{26}$$

where $\varphi_{\boldsymbol{L}}, \varphi_s, \varphi_F$ are MLPs.

5. Experiments

In this section, we evaluate the performance of EquiCSP on diverse tasks, by showing the capability of generating high-quality structures of different crystals in Section 5.1. Ablations in Section 5.2 show the necessity of each designed component. We further exhibit the capability of EquiCSP in the ab initio generation task in Appendix E.

5.1. Stable Structure Prediction Results

Datasets. Experiments are carried out on three datasets, each varying in complexity. The **Perov-5** dataset (Castelli et al., 2012a;b) comprises 18,928 perovskite materials, characterized by their analogous structural configurations. Notably, each structure within this dataset features a unit cell containing 5 atoms. The dataset **MP-20** comprises 45,231 stable inorganic materials curated from the Material Projects(Jain et al., 2013). This dataset predominantly includes materials that are experimentally generated and contain no more than 20 atoms per unit cell. In addition, **MPTS-52** represents a more challenging extension of MP-20, encompassing 40,476 structures with up to 52 atoms per cell. These structures are organized based on the earliest year of publication in the literature. For datasets such as Perov-5, and MP-20, we adhere to a 60-20-20 split for training, validation, and testing, respectively, aligning with the methodology of Jiao et al. (2023). Conversely, for MPTS-52, we allocate 27,380 entries for training, 5,000 for validation, and 8,096 for testing, arranged in chronological order.

Baselines. This study contrasts two categories of preceding research. The initial category adopts a predict-optimize approach, initially training a property predictor, followed by employing optimization algorithms for identifying optimal structures. Following Cheng et al. (2022), we use MEGNet (Chen et al., 2019) for formation energy prediction. For optimization, we select Random Search (**RS**), Bayesian Optimization (**BO**), and Particle Swarm Optimization (**PSO**), each conducted over 5,000 iterations. The second category revolves around deep generative models. In line with modifications by Xie et al. (2021), we employ cG-SchNet (Gebauer et al., 2022), integrating SchNet (Schütt et al., 2018) as its

core and incorporating ground-truth lattice initialization to encode periodicity, resulting in the **P-cG-SchNet** model. Another baseline, **CDVAE** (Xie et al., 2021), which is a VAE-based approach for crystal generation, predicts lattice and initial composition, and then optimizes atom types and coordinates using annealed Langevin dynamics. Following the method by (Jiao et al., 2023),we adapt CDVAE for the CSP task. **DiffCSP** (Jiao et al., 2023), a diffusion method, learns stable structure distributions, incorporating translation, rotation, and periodicity, effectively modeling material systems.

Evaluation metrics. Adhering to established protocols (Xie et al., 2021), we assess performance by comparing predicted candidates against ground-truth structures. For each test set structure, we generate one samples with identical composition, considering a match if any sample aligns with the ground truth under pymatgen's StructureMatcher class metrics (Ong et al., 2013), with ltol=0.3, angle_tol=10, stol=0.5. The **Match rate** reflects the ratio of matched structures in the test set. **RMSE** is computed between the ground truth and the closest matching candidate, normalized by $\sqrt[3]{V/n}$ where V represents the lattice volume, and averaged across matched structures.

Results. Table 1 presents the following key insights: **1.** Optimization approaches exhibit low Match rates, indicating the challenging nature of pinpointing optimal structures within the expansive search space. **2.** Our method outperforms other generative approaches, which underscore our method's effectiveness in incorporating symmetry awareness during training and inference. **3.** Across datasets ranging from Perov-5 to MPTS-52, all techniques experience a drop in performance with increasing atoms per cell. Despite this, our approach consistently surpasses the performance of other methods. In particular, our method significantly improved the RMSE metric, indicating that our equivariant diffusion approach effectively reduces the redundancy in the solution space, allowing the model to better learn the distribution characteristics of crystal data.

5.2. Ablation Studies

In Table 2, we conduct an ablation study on each component of EquiCSP, exploring the following aspects. **1**. To verify the necessity of lattice permutation equivariance in the generation procedure, we conduct experiments by removing the loss component of lattice permutation. Result indicates that 3.77% decrease in match rate and 6.67% increase in RMSE, both of these performance metrics have deteriorated. In addition, we also compared the performance of the Frame Average method, which is also lower than that of our proposed method. **2**. Without applying periodic CoM-free noising, we observe a significant deterioration in the performance metrics, with the match rate dropping from 57.59% to 52.31% and the RMSE increasing from 0.0510 to 0.0594. This substantial change indicates that our methodology has effectively captured the characteristics of crystalline periodic translation during training, leading to a notable impact on the evaluation metrics. **3**. To further investigate the importance of the Von Mises and Probalistic Model components in periodic translation, we conducted ablation studies on these two modules separately. Even without utilizing these two components, we observed performance improvements compared to scenarios where the noising method $\boldsymbol{m}(\cdot)$ was not used. This indicates that our approach of modifying the noising method to ensure the score is periodic translation invariant is valid. And we observed that removing any or all of the Von Mises and Probalistic Model components will reduce the performance of the model in terms of match rate and RMSE, indicating that both components play a positive role in performance.

Table 2. Ablation studies of EquiCSP model on MP-20.

Method	Performance	
	Match rate↑	RMSE↓
EquiCSP	**57.59**	**0.0510**
w/o Lattice Permutation Equivariance		
w/o permutation loss	55.42	0.0544
$w/$ Frame Average	55.92	0.0578
w/o Periodic CoM − free Noising		
w/o $\boldsymbol{m}(\cdot)$	52.31	0.0594
$w/$ Partial Periodic CoM − free Noising		
w/o Probalistic Model & w/o Von Mises	54.77	0.0578
$w/$ Von Mises & w/o Probalistic Model	57.03	0.0537
$w/$ Probalistic Model & w/o Von Mises	56.32	0.0525

6. Conclusion

In summary, we introduce EquiCSP, a novel equivariant diffusion generative model for Crystal Structure Prediction task. We addresses a previously unacknowledged challenge in current models: lattice permutation equivariance. During the diffusion phase, when lattice parameters undergo permutation, the fractional coordinates of atoms experience an equivariant transformation, ensuring consistency and preserving structural integrity. Furthermore, we have devised an innovative noising algorithm that meticulously preserves periodic translation equivariance throughout both the inference and training phases. Experimental results unequivocally demonstrate that EquiCSP outperforms existing CSP methods, achieving superior in generating high quilty structures.

Acknowledgements

This research was jointly supported by the following project: the National Key RD Program of China (2021YFB0301300); the Major Program of Guangdong

Basic and Applied Research (2019B030302002); Guangdong Province Special Support Program for Cultivating High-Level Talents (2021TQ06X160); Pazhou Lab Research Project (PZL2023KF0001); the Fundamental Research Funds for the Central Universities, Sun Yat-sen University (23xkjc016); the National Science and Technology Major Project under Grant 2020AAA0107300; the National Natural Science Foundation of China (No. 61925601, No. 62376276); Beijing Nova Program (No. 20230484278); Alibaba Damo Research Fund.

Impact Statement

The development of equivariant diffusion models for CSP holds transformative potential across a broad range of scientific disciplines, including materials science, chemistry, and physics. By accurately predicting the arrangement of atoms in crystal structures, this technology can significantly expedite the discovery of novel materials. This, in turn, has far-reaching implications for various applications such as renewable energy, pharmaceuticals, electronics, and more, where innovative materials can lead to advancements in efficiency, efficacy, and sustainability. Moreover, the ability to predict crystal structures from their elemental compositions could reduce the need for expensive and time-consuming physical experiments, making research more accessible and accelerating the pace of innovation.

References

Bai, L. and Breen, D. Calculating center of mass in an unbounded 2d environment. *Journal of Graphics Tools*, 13(4):53–60, 2008.

Castelli, I. E., Landis, D. D., Thygesen, K. S., Dahl, S., Chorkendorff, I., Jaramillo, T. F., and Jacobsen, K. W. New cubic perovskites for one-and two-photon water splitting using the computational materials repository. *Energy & Environmental Science*, 5(10):9034–9043, 2012a.

Castelli, I. E., Olsen, T., Datta, S., Landis, D. D., Dahl, S., Thygesen, K. S., and Jacobsen, K. W. Computational screening of perovskite metal oxides for optimal solar light capture. *Energy & Environmental Science*, 5(2): 5814–5819, 2012b.

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. Open catalyst 2020 (oc20) dataset and community challenges. *ACS Catalysis*, 2021. doi: 10.1021/acscatal.0c04525.

Chen, C., Ye, W., Zuo, Y., Zheng, C., and Ong, S. P. Graph networks as a universal machine learning framework for molecules and crystals. *Chemistry of Materials*, 31(9): 3564–3572, 2019.

Cheng, G., Gong, X.-G., and Yin, W.-J. Crystal structure prediction by combining graph network and optimization algorithm. *Nature communications*, 13(1):1–8, 2022.

Court, C. J., Yildirim, B., Jain, A., and Cole, J. M. 3-d inorganic crystal structure generation and property prediction via representation learning. *Journal of chemical information and modeling*, 60(10):4518–4535, 2020.

Davies, D. W., Butler, K. T., Jackson, A. J., Skelton, J. M., Morita, K., and Walsh, A. Smact: Semiconducting materials by analogy and chemical theory. *Journal of Open Source Software*, 4(38):1361, 2019.

Desiraju, G. R. Cryptic crystallography. *Nature materials*, 1(2):77–79, 2002.

Fuchs, F., Worrall, D. E., Fischer, V., and Welling, M. Se(3)-transformers: 3d roto-translation equivariant attention networks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Gatto, R. and Jammalamadaka, S. R. The generalized von mises distribution. *Statistical Methodology*, 4(3):341–353, 2007.

Gebauer, N., Gastegger, M., and Schütt, K. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 7566–7578. Curran Associates, Inc., 2019.

Gebauer, N. W., Gastegger, M., Hessmann, S. S., Müller, K.-R., and Schütt, K. T. Inverse design of 3d molecular structures with conditional generative neural networks. *Nature communications*, 13(1):1–11, 2022.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Hoffmann, J., Maestrati, L., Sawada, Y., Tang, J., Sellier, J. M., and Bengio, Y. Data-driven approach to encoding and decoding 3-d crystal structures. *arXiv preprint arXiv:1909.00949*, 2019.

Hofmann, D. W. and Apostolakis, J. Crystal structure prediction by data mining. *Journal of Molecular Structure*, 647(1-3):17–39, 2003.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, pp. 8867–8887. PMLR, 2022.

Hu, J., Yang, W., and Dilanga Siriwardane, E. M. Distance matrix-based crystal structure prediction using evolutionary algorithms. *The Journal of Physical Chemistry A*, 124 (51):10909–10919, 2020.

Hu, J., Yang, W., Dong, R., Li, Y., Li, X., Li, S., and Siriwardane, E. M. Contact map based crystal structure prediction using global optimization. *CrystEngComm*, 23 (8):1765–1776, 2021.

Jacobsen, T., Jørgensen, M., and Hammer, B. On-the-fly machine learning of atomic potential in density functional theory structure optimization. *Physical review letters*, 120 (2):026102, 2018.

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):011002, 2013.

Jiao, R., Huang, W., Lin, P., Han, J., Chen, P., Lu, Y., and Liu, Y. Crystal structure prediction by joint equivariant diffusion. *arXiv preprint arXiv:2309.04475*, 2023.

Jin, W., Chen, X., Vetticaden, A., Sarzikova, S., Raychowdhury, R., Uhler, C., and Hacohen, N. Dsmbind: Se (3) denoising score matching for unsupervised binding energy prediction and nanobody design. *bioRxiv*, pp. 2023–12, 2023.

Kim, S., Noh, J., Gu, G. H., Aspuru-Guzik, A., and Jung, Y. Generative adversarial networks for crystal structure prediction. *ACS central science*, 6(8):1412–1420, 2020.

Kipf, T. N. and Welling, M. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

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

Luo, S., Shi, C., Xu, M., and Tang, J. Predicting molecular conformation via dynamic graph score matching. *Advances in Neural Information Processing Systems*, 34: 19784–19795, 2021.

Luo, S., Su, Y., Peng, X., Wang, S., Peng, J., and Ma, J. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022.

Luo, Y., Liu, C., and Ji, S. Towards symmetry-aware generation of periodic materials. *arXiv preprint arXiv:2307.02707*, 2023.

Mardia, K. V., Jupp, P. E., and Mardia, K. *Directional statistics*, volume 2. Wiley Online Library, 2000.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

Niu, C., Song, Y., Song, J., Zhao, S., Grover, A., and Ermon, S. Permutation invariant graph generation via score-based generative modeling. In *International Conference on Artificial Intelligence and Statistics*, pp. 4474–4484. PMLR, 2020.

Noh, J., Kim, J., Stein, H. S., Sanchez-Lengeling, B., Gregoire, J. M., Aspuru-Guzik, A., and Jung, Y. Inverse design of solid-state materials via a continuous representation. *Matter*, 1(5):1370–1384, 2019.

Nouira, A., Sokolovska, N., and Crivello, J.-C. Crystalgan: learning to discover crystallographic structures with generative adversarial networks. *arXiv preprint arXiv:1810.11203*, 2018.

Oganov, A., Lyakhov, A., and Valle, M. How evolutionary crystal structure prediction works—and why. *Accounts of chemical research*, 44:227–37, 03 2011. doi: 10.1021/ar1001318.

Oganov, A. R. and Glass, C. W. Crystal structure prediction using ab initio evolutionary techniques: Principles and applications. *Journal of Chemical Physics*, 124(24):201–419, 2006.

Oganov, A. R., Pickard, C. J., Zhu, Q., and Needs, R. J. Structure prediction drives materials discovery. *Nature Reviews Materials*, 4(5):331–348, 2019.

Ong, S. P., Richards, W. D., Jain, A., Hautier, G., Kocher, M., Cholia, S., Gunter, D., Chevrier, V. L., Persson, K. A., and Ceder, G. Python materials genomics (pymatgen): A robust, open-source python library for materials analysis. *Computational Materials Science*, 68:314–319, 2013.

Pickard, C. J. Airss data for carbon at 10gpa and the c+n+h+o system at 1gpa, 2020.

Pickard, C. J. and Needs, R. Ab initio random structure searching. *Journal of Physics: Condensed Matter*, 23(5): 053201, 2011.

Podryabinkin, E. V., Tikhonov, E. V., Shapeev, A. V., and Oganov, A. R. Accelerating crystal structure prediction by machine-learning interatomic potentials with active learning. *Physical Review B*, 99(6):064114, 2019.

Puny, O., Atzmon, M., Ben-Hamu, H., Misra, I., Grover, A., Smith, E. J., and Lipman, Y. Frame averaging for invariant and equivariant network design. *arXiv preprint arXiv:2110.03336*, 2021.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Ren, Z., Tian, S. I. P., Noh, J., Oviedo, F., Xing, G., Li, J., Liang, Q., Zhu, R., Aberle, A. G., Sun, S., Wang, X., Liu, Y., Li, Q., Jayavelu, S., Hippalgaonkar, K., Jung, Y., and Buonassisi, T. An invertible crystallographic representation for general inverse design of inorganic crystals with targeted properties. *Matter*, 2021. ISSN 2590-2385. doi: https://doi.org/10.1016/j.matt.2021.11.032.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models, 2021.

Satorras, V. G., Hoogeboom, E., and Welling, M. E (n) equivariant graph neural networks. In *International Conference on Machine Learning*, pp. 9323–9332. PMLR, 2021.

Schütt, K. T., Sauceda, H. E., Kindermans, P.-J., Tkatchenko, A., and Müller, K.-R. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24):241722, 2018.

Shi, C., Luo, S., Xu, M., and Tang, J. Learning gradient fields for molecular conformation generation. In *International Conference on Machine Learning*, pp. 9558–9568. PMLR, 2021.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Thölke, P. and De Fabritiis, G. Equivariant transformers for neural network based molecular potentials. In *International Conference on Learning Representations*, 2021.

Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., and Riley, P. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

Tran, R., Lan, J., Shuaibi, M., Goyal, S., Wood, B. M., Das, A., Heras-Domingo, J., Kolluru, A., Rizvi, A., Shoghi, N., et al. The open catalyst 2022 (oc22) dataset and challenges for oxide electrocatalysis. *arXiv preprint arXiv:2206.08917*, 2022.

Vignac, C., Krawczuk, I., Siraudin, A., Wang, B., Cevher, V., and Frossard, P. Digress: Discrete denoising diffusion for graph generation. In *The Eleventh International Conference on Learning Representations*, 2023.

Vincent, P. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Wang, Y., Lv, J., Zhu, L., and Ma, Y. Crystal structure prediction via particle swarm optimization. *Physics*, 82 (9):7174–7182, 2010a.

Wang, Y., Lv, J., Zhu, L., and Ma, Y. Crystal structure prediction via particle-swarm optimization. *Physical Review B*, 82(9):094116, 2010b.

Wang, Y., Lv, J., Zhu, L., and Ma, Y. Calypso: A method for crystal structure prediction. *Computer Physics Communications*, 183(10):2063–2070, 2012. ISSN 0010-4655. doi: https://doi.org/10.1016/j.cpc.2012.05.008. URL https://www.sciencedirect.com/science/article/pii/S0010465512001762.

Ward, L., Agrawal, A., Choudhary, A., and Wolverton, C. A general-purpose machine learning framework for predicting properties of inorganic materials. *npj Computational Materials*, 2(1):1–7, 2016.

Xie, T. and Grossman, J. C. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Phys. Rev. Lett.*, 120:145301, Apr 2018. doi: 10.1103/PhysRevLett.120.145301.

Xie, T., Fu, X., Ganea, O.-E., Barzilay, R., and Jaakkola, T. S. Crystal diffusion variational autoencoder for periodic material generation. In *International Conference on Learning Representations*, 2021.

Xu, M., Yu, L., Song, Y., Shi, C., Ermon, S., and Tang, J. Geodiff: A geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations*, 2021.

Xu, M., Yu, L., Song, Y., Shi, C., Ermon, S., and Tang, J. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.

Yamashita, T., Sato, N., Kino, H., Miyake, T., Tsuda, K., and Oguchi, T. Crystal structure prediction accelerated by bayesian optimization. *Physical Review Materials*, 2 (1):013803, 2018.

Yan, K., Liu, Y., Lin, Y., and Ji, S. Periodic graph transformers for crystal material property prediction. In *The 36th Annual Conference on Neural Information Processing Systems*, 2022.

Yang, W., Siriwardane, E. M. D., Dong, R., Li, Y., and Hu, J. Crystal structure prediction of materials with high symmetry using differential evolution. *Journal of Physics: Condensed Matter*, 33(45):455902, 2021.

Zhang, Y., Wang, H., Wang, Y., Zhang, L., and Ma, Y. Computer-assisted inverse design of inorganic electrides. *Physical Review X*, 7(1):011017, 2017.

Zimmermann, N. E. and Jain, A. Local structure order parameters and site fingerprints for quantification of coordination environment and crystal structure similarity. *RSC advances*, 10(10):6063–6081, 2020.

A. Theoretical Analysis.

A.1. Proof of Proposition 4.5

We first introduce the following definition to describe the equivariance and invariance from the perspective of distributions.

Definition A.1. We call a distribution $p(x)$ is G-invariant if for any transformation g in the group G, $p(g \cdot x) = p(x)$, and a conditional distribution $p(x|c)$ is G-equivariant if $p(g \cdot x|g \cdot c) = p(x|c), \forall g \in G$.

We then provide the following lemma to capture the symmetry of the generation process.

Lemma A.2 (Xu et al. (2021))**.** *Consider the generation Markov process $p(x_0) = p(x_T) \int p(x_{0:T-1}|x_t) dx_{1:T}$. If the prior distribution $p(x_T)$ is G-invariant and the Markov transitions $p(x_{t-1}|x_t), 0 < t \leq T$ are G-equivariant, the marginal distribution $p(x_0)$ is also G-invariant.*

The proposition Proposition 4.5 is rewritten and proved as follows.

Proof. Consider the transition probability in Eq. (2), we have

$$p(\boldsymbol{C}_{t-1}|\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}) = \mathcal{N}(\boldsymbol{C}_{t-1}|a_t(\boldsymbol{C}_t - b_t \hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}}(\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}, t)), \sigma_t^2 \boldsymbol{I}),$$

where $a_t = \frac{1}{\sqrt{\alpha_t}}, b_t = \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}, \sigma_t^2 = \beta_t \cdot \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}$ for simplicity, and $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}}(\mathcal{M}_t, t)$ is completed as $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}}(\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}, t)$.

As the denoising term $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}}(\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}, t)$ is lattice permutation equivariant, we have $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}}(\boldsymbol{P}\boldsymbol{C}_t, \boldsymbol{P}\boldsymbol{F}_t, \boldsymbol{A}, t) = \boldsymbol{P}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}}(\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}, t)$ for any permutation matrix $\boldsymbol{P} \in S_3, \boldsymbol{P}^\top \boldsymbol{P} = \boldsymbol{I}$.

For the variable $\boldsymbol{C} \sim \mathcal{N}(\bar{\boldsymbol{C}}, \sigma^2 \boldsymbol{I})$, we have $\boldsymbol{P}\boldsymbol{C} \sim \mathcal{N}(\boldsymbol{P}\bar{\boldsymbol{C}}, \boldsymbol{P}(\sigma^2 \boldsymbol{I})\boldsymbol{P}^\top) = \mathcal{N}(\boldsymbol{P}\bar{\boldsymbol{C}}, \sigma^2 \boldsymbol{I})$. That is,

$$\mathcal{N}(\boldsymbol{C}|\bar{\boldsymbol{C}}, \sigma^2 \boldsymbol{I}) = \mathcal{N}(\boldsymbol{P}\boldsymbol{C}|\boldsymbol{P}\bar{\boldsymbol{C}}, \sigma^2 \boldsymbol{I}). \tag{27}$$

For the transition probability $p(\boldsymbol{C}_{t-1}|\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A})$, we have

$$\begin{aligned}
p(\boldsymbol{P}\boldsymbol{C}_{t-1}|\boldsymbol{P}\boldsymbol{C}_t, \boldsymbol{P}\boldsymbol{F}_t, \boldsymbol{A}) &= \mathcal{N}(\boldsymbol{P}\boldsymbol{C}_{t-1}|a_t(\boldsymbol{P}\boldsymbol{C}_t - b_t \hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}}(\boldsymbol{P}\boldsymbol{C}_t, \boldsymbol{P}\boldsymbol{F}_t, \boldsymbol{A}, t)), \sigma_t^2 \boldsymbol{I}) \\
&= \mathcal{N}(\boldsymbol{P}\boldsymbol{C}_{t-1}|a_t(\boldsymbol{P}\boldsymbol{C}_t - b_t \boldsymbol{P}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}}(\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}, t)), \sigma_t^2 \boldsymbol{I}) \quad \text{(lattice permutation equivariant } \hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}}) \\
&= \mathcal{N}(\boldsymbol{P}\boldsymbol{C}_{t-1}|\boldsymbol{P}\Big(a_t(\boldsymbol{C}_t - b_t \hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}}(\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}, t))\Big), \sigma_t^2 \boldsymbol{I}) \\
&= \mathcal{N}(\boldsymbol{C}_{t-1}|a_t(\boldsymbol{C}_t - b_t \hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}}(\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}, t)), \sigma_t^2 \boldsymbol{I}) \quad \text{(Eq. (27))} \\
&= p(\boldsymbol{C}_{t-1}|\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}).
\end{aligned}$$

As the transition is lattice permutation equivariant and the prior distribution $\mathcal{N}(0, \boldsymbol{I})$ is lattice permutation invariant, we prove that the the marginal distribution $p(\boldsymbol{C}_0)$ is lattice permutation invariant based on lemma A.2. □

A.2. Proof of Proposition 4.6

Let $\mathcal{N}_w(\mu, \sigma^2 \boldsymbol{I})$ denote the wrapped normal distribution with mean μ, variance σ^2 and period 1. We first provide the following lemma.

Lemma A.3. *If the denoising term $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{F}}(\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}, t)$ is lattice permutation equivariant, and the transition probabilty can be formulated as $p(\boldsymbol{F}_{t-1}|\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}) = \mathcal{N}_w(\boldsymbol{F}_{t-1}|\boldsymbol{F}_t + u_t \hat{\boldsymbol{\epsilon}}_{\boldsymbol{F}}(\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}, t), v_t^2 \boldsymbol{I})$, where u_t, v_t are functions of t, the transition is lattice permutation equivariant.*

Proof. For the variable $\boldsymbol{F} \sim \mathcal{N}_w(\bar{\boldsymbol{F}}, v_t^2 \boldsymbol{I})$ and $\boldsymbol{P} \in S_3$, we have $\boldsymbol{P}\boldsymbol{F} \sim \mathcal{N}_w(\boldsymbol{P}\bar{\boldsymbol{F}}, \boldsymbol{P}(v_t^2 \boldsymbol{I})\boldsymbol{P}^\top) = \mathcal{N}_w(\boldsymbol{P}\bar{\boldsymbol{F}}, v_t^2 \boldsymbol{I})$. That is,

$$\mathcal{N}_w(\boldsymbol{F}|\bar{\boldsymbol{F}}, v_t^2 \boldsymbol{I}) = \mathcal{N}_w(\boldsymbol{P}\boldsymbol{F}|\boldsymbol{P}\bar{\boldsymbol{F}}, v_t^2 \boldsymbol{I}). \tag{28}$$

For the transition probability $p(\boldsymbol{F}_{t-1}|\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A})$, we have

$$\begin{aligned}
p(\boldsymbol{P}\boldsymbol{F}_{t-1}|\boldsymbol{P}\boldsymbol{C}_t, \boldsymbol{P}\boldsymbol{F}_t, \boldsymbol{A}) &= \mathcal{N}_w(\boldsymbol{P}\boldsymbol{F}_{t-1}|\boldsymbol{P}\boldsymbol{F}_t + u_t\hat{\boldsymbol{\epsilon}}_{\boldsymbol{F}}(\boldsymbol{P}\boldsymbol{C}_t, \boldsymbol{P}\boldsymbol{F}_t, \boldsymbol{A}, t), v_t^2\boldsymbol{I}) \\
&= \mathcal{N}_w(\boldsymbol{P}\boldsymbol{F}_{t-1}|\boldsymbol{P}\boldsymbol{F}_t + u_t\boldsymbol{P}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{F}}(\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}, t), v_t^2\boldsymbol{I}) \qquad \text{(lattice permutation equivariant } \hat{\boldsymbol{\epsilon}}_{\boldsymbol{F}}) \\
&= \mathcal{N}_w(\boldsymbol{P}\boldsymbol{F}_{t-1}|\boldsymbol{P}\Big(\boldsymbol{F}_t + u_t\hat{\boldsymbol{\epsilon}}_{\boldsymbol{F}}(\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}, t)\Big), v_t^2\boldsymbol{I}) \\
&= \mathcal{N}_w(\boldsymbol{F}_{t-1}|\boldsymbol{F}_t + u_t\hat{\boldsymbol{\epsilon}}_{\boldsymbol{F}}(\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}, t), v_t^2\boldsymbol{I}) \qquad \text{(Eq. (28))} \\
&= p(\boldsymbol{F}_{t-1}|\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}).
\end{aligned}$$

□

The transition probability of the fractional coordinates during the Predictor-Corrector sampling can be formulated as

$$\begin{aligned}
p(\boldsymbol{F}_{t-1}|\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}) &= p_P(\boldsymbol{F}_{t-\frac{1}{2}}|\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A})p_C(\boldsymbol{F}_{t-1}|\boldsymbol{C}_{t-1}, \boldsymbol{F}_{t-\frac{1}{2}}, \boldsymbol{A}), \\
p_P(\boldsymbol{F}_{t-\frac{1}{2}}|\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}) &= \mathcal{N}_w(\boldsymbol{F}_{t-\frac{1}{2}}|\boldsymbol{F}_t + (\sigma_t^2 - \sigma_{t-1}^2)\hat{\boldsymbol{\epsilon}}_{\boldsymbol{F}}(\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}, t), \frac{\sigma_{t-1}^2(\sigma_t^2 - \sigma_{t-1}^2)}{\sigma_t^2}\boldsymbol{I}), \\
p_C(\boldsymbol{F}_{t-1}|\boldsymbol{C}_{t-1}, \boldsymbol{F}_{t-\frac{1}{2}}, \boldsymbol{A}) &= \mathcal{N}_w(\boldsymbol{F}_{t-\frac{1}{2}}|\boldsymbol{F}_t + \gamma\frac{\sigma_{t-1}}{\sigma_1}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{F}}(\boldsymbol{C}_{t-1}, \boldsymbol{F}_{t-\frac{1}{2}}, \boldsymbol{A}, t-1), 2\gamma\frac{\sigma_{t-1}}{\sigma_1}\boldsymbol{I}),
\end{aligned} \tag{29}$$

where p_P, p_C are the transitions of the predictor and corrector. According to lemma A.3, both of the transitions are lattice permutation equivariant. Therefore, the transition $p(\boldsymbol{F}_{t-1}|\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A})$ is lattice permutation equivariant. As the prior distribution $\mathcal{U}(0, 1)$ is lattice permutation invariant, we finally prove that the marginal distribution $p(\boldsymbol{F}_0)$ is lattice permutation invariant based on lemma A.2.

A.3. Proof of Periodic CoM-free Nosing

We prove the periodic CoM-free function $\boldsymbol{m}(\boldsymbol{\epsilon})$ constructed by Eq (15) is periodic translation invariance as Eq (14) describes in this section. Since $\boldsymbol{m}(\cdot)$ can be treated as operating on each row of $\boldsymbol{\epsilon} \in \mathbb{R}^{3\times n}$ independently, without loss of generality, we use $\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2, \ldots, \epsilon_n]$ to represent any row of the origin $\boldsymbol{\epsilon}$ for simplification. Then we can rewrite $\boldsymbol{m}$ as:

$$\begin{cases}
\boldsymbol{m}(\boldsymbol{\epsilon}) = w\big(\boldsymbol{\epsilon} - \dfrac{\operatorname{atan2}(\bar{\boldsymbol{y}}(\boldsymbol{\epsilon}), \bar{\boldsymbol{x}}(\boldsymbol{\epsilon}))}{2\pi}\big), \\
\bar{\boldsymbol{y}}(\boldsymbol{\epsilon}) = \dfrac{1}{n}\displaystyle\sum_{i=0}^{n}\sin(2\pi\epsilon_i), \\
\bar{\boldsymbol{x}}(\boldsymbol{\epsilon}) = \dfrac{1}{n}\displaystyle\sum_{i=0}^{n}\cos(2\pi\epsilon_i),
\end{cases}$$

We next prove that $\boldsymbol{m}(\boldsymbol{\epsilon}) = \boldsymbol{m}(\boldsymbol{\epsilon} + r)$ for any $r \in \mathbb{R}$.

Proof. We can rewrite $\boldsymbol{m}(\boldsymbol{\epsilon}) = \boldsymbol{m}(\boldsymbol{\epsilon} + r)$:

$$\begin{aligned}
w\big(\boldsymbol{\epsilon} - \frac{\operatorname{atan2}(\bar{\boldsymbol{y}}(\boldsymbol{\epsilon}), \bar{\boldsymbol{x}}(\boldsymbol{\epsilon}))}{2\pi}\big) &= w\big(\boldsymbol{\epsilon} + r - \frac{\operatorname{atan2}(\bar{\boldsymbol{y}}(\boldsymbol{\epsilon} + r), \bar{\boldsymbol{x}}(\boldsymbol{\epsilon} + r))}{2\pi}\big) \\
\boldsymbol{\epsilon} - \frac{\operatorname{atan2}(\bar{\boldsymbol{y}}(\boldsymbol{\epsilon}), \bar{\boldsymbol{x}}(\boldsymbol{\epsilon}))}{2\pi} &= \boldsymbol{\epsilon} + r - \frac{\operatorname{atan2}(\bar{\boldsymbol{y}}(\boldsymbol{\epsilon} + r), \bar{\boldsymbol{x}}(\boldsymbol{\epsilon} + r))}{2\pi} + d,
\end{aligned}$$

where d denotes any integer. We can redefine that $r = w(r) \in [0, 1)$ because $\bar{\boldsymbol{y}}(\boldsymbol{\epsilon} + r) = \bar{\boldsymbol{y}}(\boldsymbol{\epsilon} + w(r))$ and $\bar{\boldsymbol{x}}(\boldsymbol{\epsilon} + r) = \bar{\boldsymbol{x}}(\boldsymbol{\epsilon} + w(r))$ by the periodicity of trigonometric functions and we can merge integer part of the origin r into d since d can

be any integer. Further simplification leads to:

$$
\begin{aligned}
\frac{\operatorname{atan2}\left(\bar{\boldsymbol{y}}(\boldsymbol{\epsilon}), \bar{\boldsymbol{x}}(\boldsymbol{\epsilon})\right)}{2\pi} + r + d &= \frac{\operatorname{atan2}\left(\bar{\boldsymbol{y}}(\boldsymbol{\epsilon}+r), \bar{\boldsymbol{x}}(\boldsymbol{\epsilon}+r)\right)}{2\pi}, \\
\operatorname{atan2}\left(\bar{\boldsymbol{y}}(\boldsymbol{\epsilon}), \bar{\boldsymbol{x}}(\boldsymbol{\epsilon})\right) + 2\pi \cdot r + 2\pi \cdot d &= \operatorname{atan2}\left(\bar{\boldsymbol{y}}(\boldsymbol{\epsilon}+r), \bar{\boldsymbol{x}}(\boldsymbol{\epsilon}+r)\right), \\
\tan\left(\operatorname{atan2}\left(\bar{\boldsymbol{y}}(\boldsymbol{\epsilon}), \bar{\boldsymbol{x}}(\boldsymbol{\epsilon})\right) + 2\pi \cdot r + 2\pi \cdot d\right) &= \frac{\bar{\boldsymbol{y}}(\boldsymbol{\epsilon}+r)}{\bar{\boldsymbol{x}}(\boldsymbol{\epsilon}+r)}, && (\tan(\cdot) \text{ for both sides}) \\
\tan\left(\operatorname{atan2}\left(\bar{\boldsymbol{y}}(\boldsymbol{\epsilon}), \bar{\boldsymbol{x}}(\boldsymbol{\epsilon})\right) + 2\pi \cdot r\right) &= \frac{\bar{\boldsymbol{y}}(\boldsymbol{\epsilon}+r)}{\bar{\boldsymbol{x}}(\boldsymbol{\epsilon}+r)}, \\
\frac{\frac{\bar{\boldsymbol{y}}(\boldsymbol{\epsilon})}{\bar{\boldsymbol{x}}(\boldsymbol{\epsilon})} + \tan(2\pi \cdot r)}{1 - \frac{\bar{\boldsymbol{y}}(\boldsymbol{\epsilon})}{\bar{\boldsymbol{x}}(\boldsymbol{\epsilon})}\tan(2\pi \cdot r)} &= \frac{\bar{\boldsymbol{y}}(\boldsymbol{\epsilon}+r)}{\bar{\boldsymbol{x}}(\boldsymbol{\epsilon}+r)}, && (\tan(\cdot) \text{ addition formula}) \\
\frac{\bar{\boldsymbol{y}}(\boldsymbol{\epsilon})\cos(2\pi \cdot r) + \bar{\boldsymbol{x}}(\boldsymbol{\epsilon})\sin(2\pi \cdot r)}{\bar{\boldsymbol{x}}(\boldsymbol{\epsilon})\cos(2\pi \cdot r) - \bar{\boldsymbol{y}}(\boldsymbol{\epsilon})\sin(2\pi \cdot r)} &= \frac{\bar{\boldsymbol{y}}(\boldsymbol{\epsilon}+r)}{\bar{\boldsymbol{x}}(\boldsymbol{\epsilon}+r)},
\end{aligned}
$$

Then we can prove that:

$$
\begin{aligned}
\text{the right-hand side} &= \frac{\frac{1}{n}\sum_{i=0}^{n}\sin\left(2\pi\epsilon_i + 2\pi \cdot r\right)}{\frac{1}{n}\sum_{i=0}^{n}\cos\left(2\pi\epsilon_i + 2\pi \cdot r\right)}, \\
&= \frac{\frac{1}{n}\sum_{i=0}^{n}\left(\sin\left(2\pi\epsilon_i\right)\cos 2\pi \cdot r + \cos\left(2\pi\epsilon_i\right)\sin\left(2\pi \cdot r\right)\right)}{\frac{1}{n}\sum_{i=0}^{n}\left(\cos\left(2\pi\epsilon_i\right)\cos\left(2\pi \cdot r\right) - \sin\left(2\pi\epsilon_i\right)\sin\left(2\pi \cdot r\right)\right)} \quad (\sin(\cdot) \text{ and } \cos(\cdot) \text{ addition formula}) \\
&= \frac{\bar{\boldsymbol{y}}(\boldsymbol{\epsilon})\cos(2\pi \cdot r) + \bar{\boldsymbol{x}}(\boldsymbol{\epsilon})\sin(2\pi \cdot r)}{\bar{\boldsymbol{x}}(\boldsymbol{\epsilon})\cos(2\pi \cdot r) - \bar{\boldsymbol{y}}(\boldsymbol{\epsilon})\sin(2\pi \cdot r)} \\
&= \text{the left-hand side}
\end{aligned} \tag{30}
$$

We finally prove that $\boldsymbol{m}(\boldsymbol{\epsilon})$ is periodic translation invariance.

□

A.4. Proof of Proposition 4.7

Since $\nabla_{\boldsymbol{\epsilon}}\big(\boldsymbol{m}(\boldsymbol{\epsilon})[:, i]\big)$ can be treated as operating on each row of $\boldsymbol{\epsilon} \in \mathbb{R}^{3\times n}$ independently, without loss of generality, we use $\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2, \ldots, \epsilon_n]$ to represent any row of the origin $\boldsymbol{\epsilon}$, and use $\boldsymbol{m}_i(\boldsymbol{\epsilon})$ to represent the origin $\boldsymbol{m}(\boldsymbol{\epsilon})[:, i]$ for simplification. Then we can rewrite $\boldsymbol{m}(\boldsymbol{\epsilon})[:, i]$ as:

$$
\begin{cases}
\boldsymbol{m}_i(\boldsymbol{\epsilon}) = w\big(\epsilon_i - \dfrac{\operatorname{atan2}\left(\bar{\boldsymbol{y}}(\boldsymbol{\epsilon}), \bar{\boldsymbol{x}}(\boldsymbol{\epsilon})\right)}{2\pi}\big), \\
\bar{\boldsymbol{y}}(\boldsymbol{\epsilon}) = \dfrac{1}{n}\displaystyle\sum_{j=0}^{n}\sin\left(2\pi\epsilon_j\right), \\
\bar{\boldsymbol{x}}(\boldsymbol{\epsilon}) = \dfrac{1}{n}\displaystyle\sum_{j=0}^{n}\cos\left(2\pi\epsilon_j\right),
\end{cases} \tag{31}
$$

Our target is to prove that $\nabla_{\boldsymbol{\epsilon}}\boldsymbol{m}_i(\boldsymbol{\epsilon})$ is periodic translation invariance.

We first get the formula of $\nabla_{\boldsymbol{\epsilon}}\boldsymbol{m}_i(\boldsymbol{\epsilon})$:

$$
\frac{\partial \boldsymbol{m}_i}{\partial \epsilon_j} = \begin{cases}
-\dfrac{1}{2\pi}\left(\dfrac{\bar{x}}{\bar{x}^2 + \bar{y}^2} \cdot \dfrac{\partial \bar{y}}{\partial \epsilon_j} - \dfrac{\bar{y}}{\bar{x}^2 + \bar{y}^2} \cdot \dfrac{\partial \bar{x}}{\partial \epsilon_j}\right), \text{ if } i \neq j, \\
1 - \dfrac{1}{2\pi}\left(\dfrac{\bar{x}}{\bar{x}^2 + \bar{y}^2} \cdot \dfrac{\partial \bar{y}}{\partial \epsilon_j} - \dfrac{\bar{y}}{\bar{x}^2 + \bar{y}^2} \cdot \dfrac{\partial \bar{x}}{\partial \epsilon_j}\right), \text{ if } i = j,
\end{cases}
$$

where

$$\begin{aligned}
\bar{y} &= \bar{\boldsymbol{y}}(\boldsymbol{\epsilon}), \\
\bar{x} &= \bar{\boldsymbol{x}}(\boldsymbol{\epsilon}), \\
\frac{\partial \bar{y}}{\partial \epsilon_j} &= \frac{1}{n} \cdot 2\pi \cos(2\pi\epsilon_j), \\
\frac{\partial \bar{x}}{\partial \epsilon_j} &= -\frac{1}{n} \cdot 2\pi \sin(2\pi\epsilon_j).
\end{aligned}$$

Substituting these partial derivatives, we obtain:

$$\frac{\partial \boldsymbol{m}_i}{\partial \epsilon_j} = \begin{cases} -\dfrac{1}{n(\bar{x}^2+\bar{y}^2)} \left(\bar{x}\cos(2\pi\epsilon_j) + \bar{y}\sin(2\pi\epsilon_j)\right), \text{ if } i \neq j, \\ 1 - \dfrac{1}{n(\bar{x}^2+\bar{y}^2)} \left(\bar{x}\cos(2\pi\epsilon_j) + \bar{y}\sin(2\pi\epsilon_j)\right), \text{ if } i = j, \end{cases} \tag{32}$$

where $\bar{y} = \bar{\boldsymbol{y}}(\boldsymbol{\epsilon})$ and $\bar{x} = \bar{\boldsymbol{x}}(\boldsymbol{\epsilon})$. Thus, the gradient $\nabla_{\boldsymbol{\epsilon}} \boldsymbol{m}_i(\boldsymbol{\epsilon})$ is a vector with its jth component given by Eq.(32).

We next prove $\nabla_{\boldsymbol{\epsilon}} \boldsymbol{m}_i(\boldsymbol{\epsilon}) = \nabla_{\boldsymbol{\epsilon}+r} \boldsymbol{m}_i(\boldsymbol{\epsilon}+r)$.

Proof. By Eq.(32), we can rewrite $\nabla_{\boldsymbol{\epsilon}} \boldsymbol{m}_i(\boldsymbol{\epsilon}) = \nabla_{\boldsymbol{\epsilon}+r} \boldsymbol{m}_i(\boldsymbol{\epsilon}+r)$ as:

$$\frac{\bar{\boldsymbol{x}}(\boldsymbol{\epsilon})\cos(2\pi\epsilon_j) + \bar{\boldsymbol{y}}(\boldsymbol{\epsilon})\sin(2\pi\epsilon_j)}{\bar{\boldsymbol{x}}^2(\boldsymbol{\epsilon}) + \bar{\boldsymbol{y}}^2(\boldsymbol{\epsilon})} = \frac{\bar{\boldsymbol{x}}(\boldsymbol{\epsilon}+2\pi r)\cos(2\pi\epsilon_j + 2\pi r) + \bar{\boldsymbol{y}}(\boldsymbol{\epsilon}+2\pi r)\sin(2\pi\epsilon_j + 2\pi r)}{\bar{\boldsymbol{x}}^2(\boldsymbol{\epsilon}+2\pi r) + \bar{\boldsymbol{y}}^2(\boldsymbol{\epsilon}+2\pi r)} \tag{33}$$

Referring to Eq.(30), we have:

$$\begin{aligned}
\bar{\boldsymbol{y}}(\boldsymbol{\epsilon}+r) &= \bar{\boldsymbol{y}}(\boldsymbol{\epsilon})\cos(2\pi \cdot r) + \bar{\boldsymbol{x}}(\boldsymbol{\epsilon})\sin(2\pi \cdot r), \\
\bar{\boldsymbol{x}}(\boldsymbol{\epsilon}+r) &= \bar{\boldsymbol{x}}(\boldsymbol{\epsilon})\cos(2\pi \cdot r) - \bar{\boldsymbol{y}}(\boldsymbol{\epsilon})\sin(2\pi \cdot r).
\end{aligned}$$

The numerator of the right-hand side of Eq.(33) can be expressed as:

$$\begin{aligned}
&\bar{x}(\boldsymbol{\epsilon}+r)\cos(2\pi\epsilon_j + 2\pi r) + \bar{y}(\boldsymbol{\epsilon}+r)\sin(2\pi\epsilon_j + 2\pi r) \\
&= [\bar{x}(\boldsymbol{\epsilon})\cos(2\pi r) - \bar{y}(\boldsymbol{\epsilon})\sin(2\pi r)][\cos(2\pi\epsilon_j)\cos(2\pi r) - \sin(2\pi\epsilon_j)\sin(2\pi r)] \\
&\quad + [\bar{y}(\boldsymbol{\epsilon})\cos(2\pi r) + \bar{x}(\boldsymbol{\epsilon})\sin(2\pi r)][\sin(2\pi\epsilon_j)\cos(2\pi r) + \cos(2\pi\epsilon_j)\sin(2\pi r)] \\
&= \bar{x}(\boldsymbol{\epsilon})\cos(2\pi\epsilon_j) + \bar{y}(\boldsymbol{\epsilon})\sin(2\pi\epsilon_j),
\end{aligned}$$

where the terms involving r cancel out due to trigonometric identities.

The denominator remains invariant under the transformation due to the Pythagorean identity:

$$\begin{aligned}
\bar{x}^2(\boldsymbol{\epsilon}+r) + \bar{y}^2(\boldsymbol{\epsilon}+r) &= [\bar{x}(\boldsymbol{\epsilon})\cos(2\pi r) - \bar{y}(\boldsymbol{\epsilon})\sin(2\pi r)]^2 + [\bar{y}(\boldsymbol{\epsilon})\cos(2\pi r) + \bar{x}(\boldsymbol{\epsilon})\sin(2\pi r)]^2 \\
&= \bar{x}^2(\boldsymbol{\epsilon}) + \bar{y}^2(\boldsymbol{\epsilon}).
\end{aligned}$$

Therefore, the given statement is proven:

$$\frac{\bar{x}(\boldsymbol{\epsilon})\cos(2\pi\epsilon_j) + \bar{y}(\boldsymbol{\epsilon})\sin(2\pi\epsilon_j)}{\bar{x}^2(\boldsymbol{\epsilon}) + \bar{y}^2(\boldsymbol{\epsilon})} = \frac{\bar{x}(\boldsymbol{\epsilon}+r)\cos(2\pi\epsilon_j + 2\pi r) + \bar{y}(\boldsymbol{\epsilon}+r)\sin(2\pi\epsilon_j + 2\pi r)}{\bar{x}^2(\boldsymbol{\epsilon}+r) + \bar{y}^2(\boldsymbol{\epsilon}+r)}.$$

We finally prove that $\nabla_{\boldsymbol{\epsilon}} \boldsymbol{m}_i(\boldsymbol{\epsilon})$ is periodic translation invariance, which is equivalent to the statement that $\nabla_{\boldsymbol{\epsilon}}\big(\boldsymbol{m}(\boldsymbol{\epsilon})[:, i]\big)$ is periodic translation invariance. □

We next prove that $\nabla_{\boldsymbol{F}_t} \log q(\boldsymbol{F}_t)$ is periodic translation invariance. We can derive it as:

$$\begin{aligned}\nabla_{\boldsymbol{F}_t} \log q(\boldsymbol{F}_t) &= \sum_{i=0}^{n} \big(\nabla_{\bar{\boldsymbol{f}}_i} \log q(\bar{\boldsymbol{f}}_i)\big)\mathbf{1}^\top \odot \nabla_{\boldsymbol{F}_t} \bar{\boldsymbol{f}}_i, \\ &= \sum_{i=0}^{n} \big(\nabla_{\bar{\boldsymbol{f}}_i} \log q(\bar{\boldsymbol{f}}_i)\big)\mathbf{1}^\top \odot \nabla_{\boldsymbol{\epsilon}} \big(\boldsymbol{m}(\boldsymbol{\epsilon})[:, i]\big),\end{aligned}$$

Since $\nabla_{\bar{\boldsymbol{f}}_i} \log q(\bar{\boldsymbol{f}}_i)$ is periodic translation invariance, and $\nabla_{\boldsymbol{\epsilon}} \big(\boldsymbol{m}(\boldsymbol{\epsilon})[:, i]\big)$ is also periodic translation invariance from the above proof, we can easily get $\nabla_{\boldsymbol{F}_t} \log q(\boldsymbol{F}_t)$ is periodic translation invariance.

B. Methods

B.1. Symmetries of crystal structure distribution

We represent the fractional coordinates on a lattice base of crystal as the points on the circle in Figure 1 (e) (f). If the points rotate alongside the circle, it means that the fractional coordinates on the base undergo a periodic translation. Specifically, the geometry of $2\pi \cdot f_i$ alongside the circle is equivalent to $2\pi w(f_i + d)$. Periodic translation invariance can be explained that any rotation on the circle does not change geometry of the angle distribution.

B.2. Diffusion on Lattice Parameters

In DDPM models (Ho et al., 2020), lattice parameters typically range from $[0, +\infty)$ for lengths and $(0, \pi)$ for angles. However, DDPMs diffuse within the $(-\infty, +\infty)$ interval, potentially generating unreasonable lattice parameters during diffusion generation. To ensure generated lattice parameters are always reasonable, we apply a logarithmic transformation to lengths, as the function $\log$ maps $(0, +\infty)$ to $(-\infty, +\infty)$, perfectly aligning with our requirement. Thus, we generate $\log \boldsymbol{l}$ instead of $\boldsymbol{l}$, and convert it back using $e^{\log \boldsymbol{l}} = \boldsymbol{l}$, ensuring lengths $\boldsymbol{l}$ are always positive. For angles, we process them with $\tan(\boldsymbol{\phi} - \pi/2)$, which also maps the desired $(0, \pi)$ to $(-\infty, +\infty)$. Upon generating values for $\tan(\boldsymbol{\phi} - \pi/2)$, we retrieve angles $\boldsymbol{\phi}$ in the $(0, \pi)$ range through $\arctan(\tan(\boldsymbol{\phi} - \pi/2) + \pi/2) = \boldsymbol{\phi}$.

B.3. Frame Average Method

Frame Average Method encodes the lattice permutation equivariance to the neural network. On the context of lattice permutation group, a frame is defined as one specific order of lattice vectors. By applying a permutation matrix to both the lattice and its fractional coordinates, we are able to transform the structure into an equivalent frame, as we described in Definition 4.5 of our paper.

Specifically, we adjusted the neural network formula to:

$$\phi_{FA}(\boldsymbol{X}) = \frac{1}{6} \sum_{\boldsymbol{P} \in S_3} \boldsymbol{P}^{-1} \phi(\boldsymbol{P}\boldsymbol{X})$$

where ϕ is the neural network model in Section 4.5, $\boldsymbol{X}$ is a $3 \times N$ matrix comprising three lattices and their respective fractional coordinates, and $\boldsymbol{P}$ represents the permutation matrices for the lattices.

C. Implementation Details.

C.1. Von Mises Distribution Simulation

The Von Mises distribution(Gatto & Jammalamadaka, 2007), often referred to as the 'circular normal' distribution, is a probability distribution used for modeling angular or directional data. It is flexible and efficient to handle periodic and directional characteristics. Let $\mathcal{V}(\mu, \kappa)$ denote the Von Mises distribution with mean direction μ, concentration parameter κ and period 1. The probability density function (PDF) of $\mathcal{V}(\mu, \kappa)$ is defined as:

$$\mathcal{V}(x; \mu, \kappa) = \frac{e^{\kappa \cos(2\pi x - \mu)}}{I_0(\kappa)},$$

where μ is the mean direction of the distribution, and κ is the concentration parameter, indicating the level of concentration around the mean direction. The function $I_0(\kappa)$ is the modified Bessel function of order zero, which normalizes the distribution.

As detailed in Section 4.4.1, we employ Von Mises distribution to simulate $p(\bar{\boldsymbol{\epsilon}}_i)$ where $\bar{\boldsymbol{\epsilon}} = \boldsymbol{m}(\boldsymbol{\epsilon})$ by Eq.(15), $\boldsymbol{\epsilon} \in \mathbb{R}^{3\times n}$ and $\boldsymbol{\epsilon} \sim \mathcal{N}_w(0, \sigma_t^2 \boldsymbol{I})$ and $\bar{\boldsymbol{\epsilon}}_i$ is the i-coloumn of $\bar{\boldsymbol{\epsilon}}$. Similar to the analysis in Appendix.A.4, we can simplify the question as: using $\mathcal{V}(\mu, \kappa)$ to simulate $p(\bar{\epsilon}_i)$ where $\bar{\epsilon}_i = \boldsymbol{m}_i(\boldsymbol{\epsilon})$ by Eq.(31), $\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2, \cdots \epsilon_n]$, $i \in \mathcal{U}(1, n)$ and $\boldsymbol{\epsilon} \sim \mathcal{N}_w(0, \sigma_t^2 \boldsymbol{I})$. Since the mean of the $\boldsymbol{\epsilon} \sim \mathcal{N}_w(0, \sigma_t^2 \boldsymbol{I})$ is 0 and the function $\boldsymbol{m}_i(\boldsymbol{\epsilon})$ intuitively moves the elements of $\boldsymbol{\epsilon}$ as a whole closer to 0, we set the mean direction of $\mathcal{V}(\mu, \kappa)$ to 0 empirically. As a result, the key of simulation is to estimate the concentration parameter κ. We denotes $\mathcal{V}(0, \kappa(n, \sigma_t))$ as the target distribution since κ is relative to the size of $\boldsymbol{\epsilon}$ i.e. n and the variance of $\mathcal{N}_w(0, \sigma_t^2 \boldsymbol{I})$ i.e. σ_t.

We employ the Monte Carlo method to estimate $\kappa(n, \sigma_t)$. For each specified n and σ_t, the procedure initiates by generating samples from $\boldsymbol{\epsilon} \sim \mathcal{N}_w(0, \sigma_t^2 \boldsymbol{I})$, which are then transformed into $\bar{\epsilon}_i$ following the methodology outlined above. These transformed points are then used to compute their respective probability values according to $\mathcal{V}(0, \kappa(n, \sigma_t))$, where $\kappa(n, \sigma_t)$ is initially set to an arbitrary value. The negative log-likelihood of these probabilities serves as the loss function. To refine the estimation and ascertain the optimal $\kappa(n, \sigma_t)$ value, we utilize the minimize function from the SciPy library, ensuring an effective and precise optimization tailored to each n and σ_t configuration.

We have obtained the approximate probability density function for each element of $\bar{\boldsymbol{\epsilon}}$. As each element is considered to be independently and identically distributed, the calculation of the score for $\bar{\boldsymbol{\epsilon}}$ involves deriving the corresponding score for each individual element, i.e.$\nabla_{\bar{\epsilon}_i} \log \mathcal{V}(\bar{\epsilon}_i | 0, \kappa(n, \sigma_t))$. To derive $\nabla_{\bar{\epsilon}_i} \log \mathcal{V}(\bar{\epsilon}_i | 0, \kappa(n, \sigma_t))$, we first consider the logarithm of $\mathcal{V}(\bar{\epsilon}_i; \mu, \kappa)$:

$$
\begin{aligned}
\log \mathcal{V}(\bar{\epsilon}_i; \mu, \kappa) &= \log \left(\frac{e^{\kappa \cos(2\pi \bar{\epsilon}_i - \mu)}}{I_0(\kappa)} \right) \\
&= \kappa \cos(2\pi \bar{\epsilon}_i - \mu) - \log I_0(\kappa)
\end{aligned}
$$

Now, taking the gradient with respect to $\bar{\epsilon}_i$, we get:

$$
\begin{aligned}
\nabla_{\bar{\epsilon}_i} \log \mathcal{V}(\bar{\epsilon}_i | 0, \kappa(n, \sigma_t)) &= \frac{d}{d\bar{\epsilon}_i} \left(\kappa(n, \sigma_t) \cos(2\pi \bar{\epsilon}_i) - \log I_0(\kappa(n, \sigma_t)) \right) \\
&= -2\pi \kappa(n, \sigma_t) \sin(2\pi \bar{\epsilon}_i)
\end{aligned} \tag{34}
$$

To sum up, we have:

$$
\begin{aligned}
\nabla_{\bar{\boldsymbol{\epsilon}}_i} \log p(\bar{\boldsymbol{\epsilon}}_i) &\approx \nabla_{\bar{\boldsymbol{\epsilon}}_i} \log \mathcal{V}(\bar{\boldsymbol{\epsilon}}_i | 0, \kappa(n, \sigma_t)) \\
&= -2\pi \kappa(n, \sigma_t) \sin(2\pi \bar{\boldsymbol{\epsilon}}_i)
\end{aligned}
$$

C.2. Probabilistic Modeling Process

We simplify Eq.(18) to improve the efficiency of calculation. Since $\nabla_{\boldsymbol{F}_t} \log q(\boldsymbol{F}_t)$ can be treated as operating on each row of $\bar{\boldsymbol{\epsilon}}$ independently, without loss of generality, we use $\bar{\boldsymbol{\epsilon}} = [\bar{\epsilon}_1, \bar{\epsilon}_2, \ldots, \bar{\epsilon}_n]$ to represent any row of the origin $\bar{\boldsymbol{\epsilon}}$ and use $\boldsymbol{s}$ to represent the corresponding row of $\nabla_{\boldsymbol{F}_t} \log q(\boldsymbol{F}_t)$ for simplification, while using $\bar{\boldsymbol{s}} = [\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_n]$ to represent the corresponding row of $\nabla_{\bar{\boldsymbol{F}}} \log q(\bar{\boldsymbol{F}})$. Then we have:

$$
\boldsymbol{s} = \sum_{i=0}^{n} \left(\bar{s}_i \cdot \nabla_{\bar{\boldsymbol{\epsilon}}} \boldsymbol{m}_i(\bar{\boldsymbol{\epsilon}}) \right),
$$

where $\nabla_{\bar{\epsilon}_i} \log p(\bar{\epsilon}_i)$ can be obtained by Eq.(34) and $\nabla_{\bar{\boldsymbol{\epsilon}}} \boldsymbol{m}_i(\bar{\boldsymbol{\epsilon}})$ can be obtained by Eq.(32). Further expanding:

$$\begin{aligned}
\boldsymbol{s} &= \bar{s}_1 \cdot [1+g_1, g_2, \ldots, g_n] + \bar{s}_2 \cdot [g_1, 1+g_2, \ldots, g_n] + \ldots \\
&\quad \bar{s}_n \cdot [g_1, g_2, \ldots, 1+g_n], \\
&= \left[\bar{s}_1 + g_1 \cdot \big(\sum_{i=0}^{n} \bar{s}_i\big), \quad \ldots, \quad \bar{s}_n + g_n \cdot \big(\sum_{i=0}^{n} \bar{s}_i\big) \right], \\
&= \bar{\boldsymbol{s}} + \big(\sum_{i=0}^{n} \bar{s}_i\big) \cdot \boldsymbol{g}(\bar{\boldsymbol{\epsilon}}),
\end{aligned}$$

where

$$\begin{cases}
\boldsymbol{g}(\bar{\boldsymbol{\epsilon}}) = [g_1, g_2, \ldots, g_n], \\
\qquad = -\dfrac{\bar{x}\cos(2\pi\bar{\boldsymbol{\epsilon}}) + \bar{y}\sin(2\pi\bar{\boldsymbol{\epsilon}})}{n(\bar{x}^2 + \bar{y}^2)} \\
\quad \bar{x} = \dfrac{1}{n}\sum_{j=0}^{n} \sin\left(2\pi\bar{\epsilon}_j\right), \\
\quad \bar{y} = \dfrac{1}{n}\sum_{j=0}^{n} \cos\left(2\pi\bar{\epsilon}_j\right).
\end{cases}$$

Consequently, we have successfully formulated a more parallelization-friendly expression of s. By extrapolating this to the initial context where $\bar{\boldsymbol{\epsilon}} = [\bar{\boldsymbol{\epsilon}}_1, \bar{\boldsymbol{\epsilon}}_2, \ldots, \bar{\boldsymbol{\epsilon}}_n] \in \mathbb{R}^{3\times n}$, the ultimate expression is thus deduced:

$$\nabla_{\boldsymbol{F}_t} \log q(\boldsymbol{F}_t) = \nabla_{\bar{\boldsymbol{F}}} \log q(\bar{\boldsymbol{F}}) + \big(\sum_{i=0}^{n} \bar{\boldsymbol{s}}_i \mathbf{1}^\top \odot \boldsymbol{g}(\bar{\boldsymbol{\epsilon}}\big), \tag{35}$$

where

$$\begin{cases}
\boldsymbol{g}(\bar{\boldsymbol{\epsilon}}) = -\dfrac{1}{n(\bar{\boldsymbol{x}}^2 + \bar{\boldsymbol{y}}^2)} \odot \big(\bar{\boldsymbol{x}} \odot \cos(2\pi\bar{\boldsymbol{\epsilon}}) + \bar{\boldsymbol{y}} \odot \sin(2\pi\bar{\boldsymbol{\epsilon}})\big) \\
\quad \bar{\boldsymbol{x}} = \big(\dfrac{1}{n}\sum_{j=0}^{n} \sin\left(2\pi\bar{\boldsymbol{\epsilon}}_j\right)\big)\mathbf{1}^\top, \\
\quad \bar{\boldsymbol{y}} = \big(\dfrac{1}{n}\sum_{j=0}^{n} \cos\left(2\pi\bar{\boldsymbol{\epsilon}}_j\right)\big)\mathbf{1}^\top.
\end{cases} \tag{36}$$

C.3. Algorithms for Training and Sampling

Algorithm 1 provides a comprehensive overview of the forward diffusion process and the training procedure for the denoising model ϕ, while Algorithm 2 elucidates the backward sampling process. These algorithms can effectively preserve symmetries if ϕ is meticulously designed. It is worth mentioning that we employ the predictor-corrector sampler (Song et al., 2020) to sample $\boldsymbol{F}_0$ in Algorithm 2, where Line 8 denotes the predictor, and Lines 11-12 correspond to the corrector. The $\boldsymbol{m}(\cdot)$ denotes the function in Eq.(15). The $w(\cdot)$ denotes the truncation function. The $\boldsymbol{g}(\cdot)$ denotes the function in Eq.(36).

C.4. Hyper-parameters and Training Details.

For our EquiCSP, we employ a 4-layer setting with 256 hidden states for Perov-5 and a 6-layer setting with 512 hidden states for other datasets. The dimension of the Fourier embedding is set to $k = 256$. We utilize the cosine scheduler with $s = 0.008$ to regulate the variance of the DDPM process on $\boldsymbol{C}_t$, and an exponential scheduler with $\sigma_1 = 0.005, \sigma_T = 0.5$ to control the noise scale of the score matching process on $\boldsymbol{F}_t$. The diffusion step is set to $T = 1000$. Our model undergoes training for 3500, 4000, 1000, and 1000 epochs respectively for Perov-5, Carbon-24, MP-20, and MPTS-52 using the same optimizer and learning rate scheduler as CDVAE. For Langevin dynamics' step size γ, we apply values of $\gamma = 5 \times 10^{-7}$ for Perov-5, $\gamma = 5 \times 10^{-6}$ for MP-20, $\gamma = 1 \times 10^{-5}$ for MPTS-52; while for ab initio generation in Carbon-24 case we use $\gamma = 1 \times 10^{-5}$. All models are trained on one Nvidia A800 GPU.

Algorithm 1 Training Procedure of EquiCSP

Input: lattice parameters $\boldsymbol{C}_0$, atom types $\boldsymbol{A}$, fractional coordinates $\boldsymbol{F}_0$, denoising model ϕ, and the number of sampling steps T.
Sample $\boldsymbol{\epsilon_L} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$,$\boldsymbol{\epsilon_F} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$,$\boldsymbol{P} \sim \mathcal{U}(\mathrm{S}_3)$ and $t \sim \mathcal{U}(1, T)$.
$\bar{\boldsymbol{\epsilon}} \leftarrow \boldsymbol{m}(w(\sigma_t \boldsymbol{\epsilon}_F))$
$\boldsymbol{C}_t \leftarrow \sqrt{\bar{\alpha}_t}\boldsymbol{C}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon_L}$
$\boldsymbol{F}_t \leftarrow w(\boldsymbol{F}_0 + \bar{\boldsymbol{\epsilon}})$
$\hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}}, s_\theta, F_\theta \leftarrow \phi(\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}, t)$
$\hat{\boldsymbol{\epsilon}}'_{\boldsymbol{L}}, s'_\theta, F'_\theta \leftarrow \phi(\boldsymbol{P}\boldsymbol{C}_t, \boldsymbol{P}\boldsymbol{F}_t, \boldsymbol{A}, t)$
$\mathcal{L}_{\boldsymbol{C}} \leftarrow \|\boldsymbol{\epsilon_L} - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}}\|_2^2$
$\mathcal{L}_s \leftarrow \|(-2\pi\kappa(n, \sigma_t)\sin(2\pi\bar{\boldsymbol{\epsilon}})) - s_\theta\|_2^2$
$\mathcal{L}_F \leftarrow \|\bar{\boldsymbol{\epsilon}} - F_\theta\|_2^2$
$\mathcal{L}_{p\boldsymbol{C}} \leftarrow \|\boldsymbol{P}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}} - \hat{\boldsymbol{\epsilon}}'_{\boldsymbol{L}}|_2^2$
$\mathcal{L}_{ps} \leftarrow \|\boldsymbol{P}s_\theta - s'_\theta|_2^2$
$\mathcal{L}_{pF} \leftarrow \|\boldsymbol{P}F_\theta - F'_\theta|_2^2$
Minimize $\mathcal{L}_{\boldsymbol{C}} + \mathcal{L}_s + \mathcal{L}_F + \mathcal{L}_{p\boldsymbol{C}} + \mathcal{L}_{ps} + \mathcal{L}_{pF}$

Algorithm 2 Sampling Procedure of EquiCSP

Input: atom types $\boldsymbol{A}$, denoising model ϕ, number of sampling steps T, step size of Langevin dynamics γ.
Sample $\boldsymbol{C}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$,$\boldsymbol{F}_T \sim \mathcal{U}(0, 1)$.
for $t \leftarrow T, \cdots, 1$ **do**
  Sample $\boldsymbol{\epsilon_L}, \boldsymbol{\epsilon_F}, \boldsymbol{\epsilon}'_{\boldsymbol{F}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}}, s_\theta, F_\theta \leftarrow \phi(\boldsymbol{C}_t, \boldsymbol{F}_t, \boldsymbol{A}, t)$.
  $\boldsymbol{C}_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}}(\boldsymbol{C}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{L}}) + \sqrt{\beta_t \cdot \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}}\boldsymbol{\epsilon_L}$.
  $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{F}} = s_\theta + (\sum_{i=0}^{n} s_\theta[:, i])\mathbf{1}^\top \odot \boldsymbol{g}(F_\theta)$
  $\boldsymbol{F}_{t-\frac{1}{2}} \leftarrow w(\boldsymbol{F}_t + (\sigma_t^2 - \sigma_{t-1}^2)\hat{\boldsymbol{\epsilon}}_{\boldsymbol{F}} + \frac{\sigma_{t-1}\sqrt{\sigma_t^2 - \sigma_{t-1}^2}}{\sigma_t}\boldsymbol{\epsilon_F})$
  $_, s_\theta, F_\theta \leftarrow \phi(\boldsymbol{C}_{t-1}, \boldsymbol{F}_{t-\frac{1}{2}}, \boldsymbol{A}, t-1)$.
  $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{F}} = s_\theta + (\sum_{i=0}^{n} s_\theta[:, i])\mathbf{1}^\top \odot \boldsymbol{g}(F_\theta)$
  $d_t \leftarrow \gamma\sigma_{t-1}/\sigma_1$
  $\boldsymbol{F}_{t-1} \leftarrow w(\boldsymbol{F}_{t-\frac{1}{2}} + d_t\hat{\boldsymbol{\epsilon}}_{\boldsymbol{F}} + \sqrt{2d_t}\boldsymbol{\epsilon}'_{\boldsymbol{F}})$.
end for
Return $\boldsymbol{C}_0, \boldsymbol{F}_0$.

D. Learning Curves of Different Variants.

We plot the curves of training loss of different variants proposed in Figure 4 and 5.

E. Ab initio Structure Generation

Dataset. We conduct experiments on **Perov-5**, **Carbon-24** and **MP-20** dataset. Notably, **Carbon-24** (Pickard, 2020) encompasses 10,153 carbon materials, each containing 6 to 24 atoms per cell. Contrasting with other datasets used in Table 3, where compositions typically correspond to a single stable structure, **Carbon-24** features a wide array of structures for any given composition. This dataset allows us to evaluate the capability to generate diverse one-to-many metastable structures, reflecting the variability inherent in crystal structures.

Extending EquiCSP to Ab Initio Generation Task We utilize the approach described in Appendix G of the DiffCSP literature (Jiao et al., 2023) to extend EquiCSP to the ab initio generation task.

Baseline. Our approach is compared against four generative methods suited to this dataset. **FTCP**(Ren et al., 2021), a coordinate-based, non-E(3)-invariant method, represents crystals via a blend of real-space and Fourier-transformed properties, utilizing a CNN-VAE architecture for generation. **G-SchNet**(Gebauer et al., 2019) employs an autoregressive model for structure generation, while **P-G-SchNet** is a G-SchNet variant incorporating periodicity. **CDVAE**(Xie et al., 2021), as previously mentioned, integrates a score matching-based decoder into the VAE framework; here, its standard version is applied without modifications. **SyMat** (Luo et al., 2023) uses a variational auto-encoder for generating periodic structures, defining lattice and atom types. **DiffCSP** (Jiao et al., 2023), a diffusion method, learns stable structure distributions, incorporating translation, rotation, and periodicity, effectively modeling material systems.

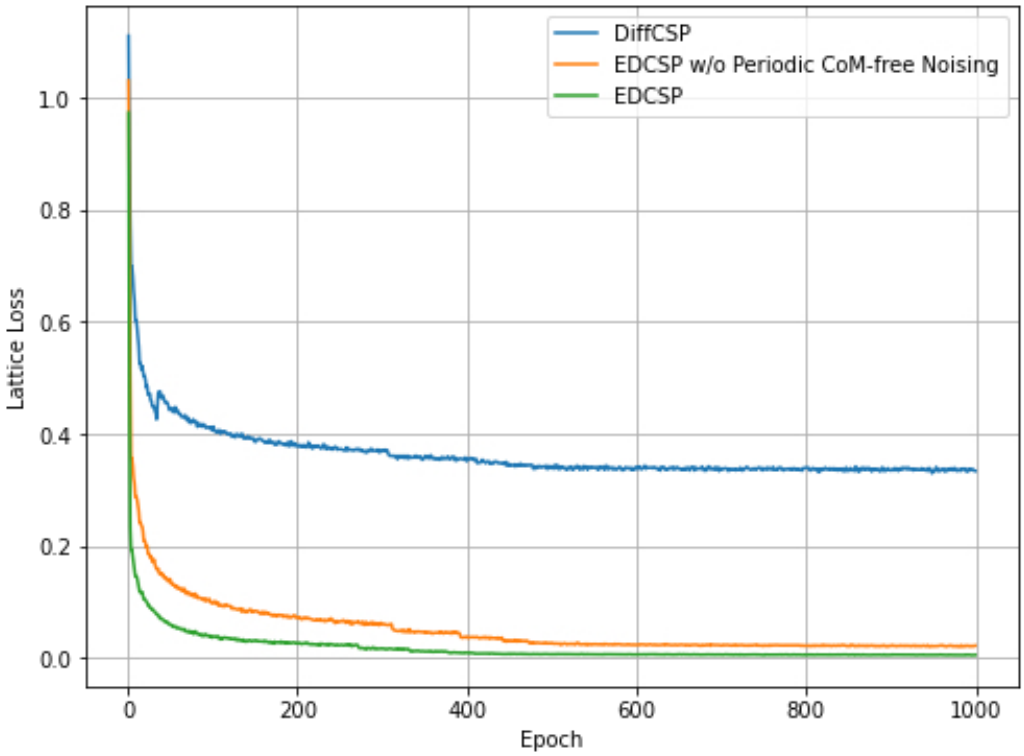

Figure 4. Learning curves of lattice loss.

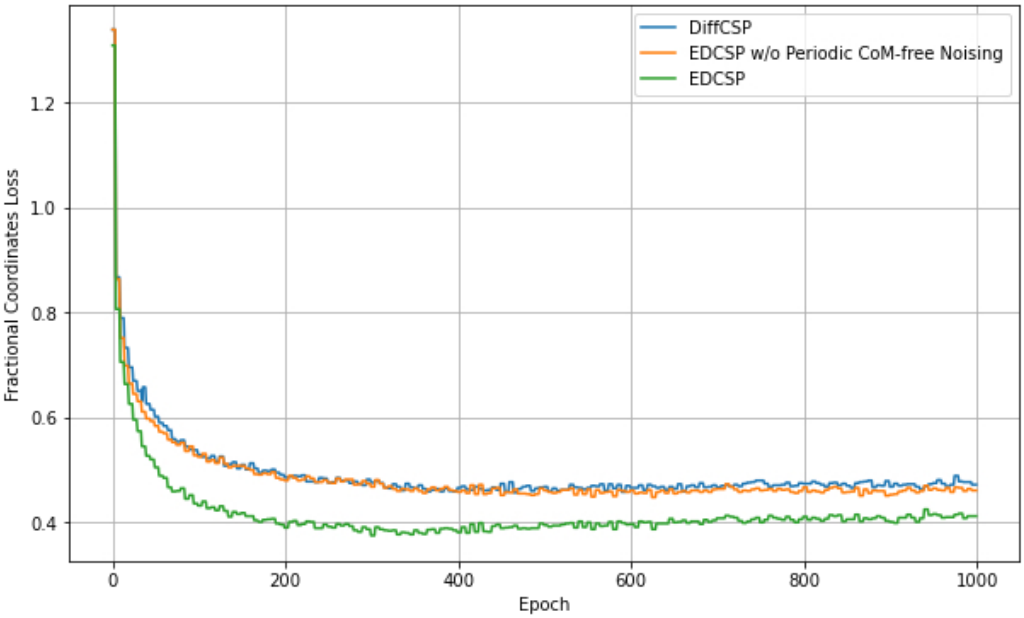

Figure 5. Leanring curves of fractional coordinates loss.

Table 3. Results on ab initio generation task. The results of baseline methods are from Jiao (Jiao et al., 2023)

Data	Method	Validity (%) ↑		Coverage (%) ↑		Property ↓		
		Struc.	Comp.	COV-R	COV-P	d_ρ	d_E	d_{elem}
Perov-5	FTCP	0.24	54.24	0.00	0.00	10.27	156.0	0.6297
	Cond-DFC-VAE	73.60	82.95	73.92	10.13	2.268	4.111	0.8373
	G-SchNet	99.92	98.79	0.18	0.23	1.625	4.746	0.0368
	P-G-SchNet	79.63	**99.13**	0.37	0.25	0.2755	1.388	0.4552
	CDVAE	**100.0**	98.59	99.45	98.46	0.1258	0.0264	0.0628
	SyMat	**100.0**	97.40	99.68	98.64	0.1893	0.2364	0.0177
	DiffCSP	**100.0**	98.85	**99.74**	98.27	**0.1110**	0.0263	**0.0128**
	EquiCSP	**100.0**	98.60	99.60	**98.76**	**0.1110**	**0.0257**	0.0503
Carbon-24	FTCP	0.08	–	0.00	0.00	5.206	19.05	–
	G-SchNet	99.94	–	0.00	0.00	0.9427	1.320	–
	P-G-SchNet	48.39	–	0.00	0.00	1.533	134.7	–
	CDVAE	**100.0**	–	99.80	83.08	0.1407	0.2850	–
	SyMat	**100.0**	–	100.0	**97.59**	0.1195	3.9576	-
	DiffCSP	**100.0**	–	**99.90**	97.27	0.0805	0.0820	–
	EquiCSP	**100.0**	–	99.75	97.12	**0.0734**	**0.0508**	–
MP-20	FTCP	1.55	48.37	4.72	0.09	23.71	160.9	0.7363
	G-SchNet	99.65	75.96	38.33	99.57	3.034	42.09	0.6411
	P-G-SchNet	77.51	76.40	41.93	99.74	4.04	2.448	0.6234
	CDVA	**100.0**	86.70	99.15	99.49	0.6875	0.2778	1.432
	SyMat	**100.0**	**88.26**	98.97	**99.97**	0.3805	0.3506	0.5067
	DiffCSP	**100.0**	83.25	**99.71**	99.76	0.3502	0.1247	**0.3398**
	EquiCSP	99.97	82.20	99.65	99.68	**0.1300**	**0.0848**	0.3978

Evaluation Metrics We assess the results using three different criteria. **Validity**: This encompasses both structural and compositional validity. Structural validity is assessed by calculating the percentage of generated structures where all pairwise distances exceed 0.5 Å, while compositional validity checks for charge neutrality using the SMACT criteria (Davies et al., 2019). **Coverage**: This metric evaluates how well the structural and compositional attributes of the generated samples $\mathcal{S}_g$ match those in the test set $\mathcal{S}_t$. It uses $d_S(\mathcal{M}_1, \mathcal{M}_2)$ and $d_C(\mathcal{M}_1, \mathcal{M}_2)$ to represent the L2 distances for CrystalNN structural fingerprints (Zimmermann & Jain, 2020) and normalized Magpie compositional fingerprints (Ward et al., 2016), respectively. Coverage Recall (COV-R) is calculated as COV-R $= \frac{1}{|\mathcal{S}_t|}|\{\mathcal{M}_i | \mathcal{M}_i \in \mathcal{S}_t, \exists \mathcal{M}_j \in \mathcal{S}_g, d_S(\mathcal{M}_i, \mathcal{M}_j) < \delta_S, d_C(\mathcal{M}_i, \mathcal{M}_j) < \delta_C\}|$, with predefined thresholds δ_S, δ_C. Coverage Precision (COV-P) is defined in a similar manner but with the roles of $\mathcal{S}_g$ and $\mathcal{S}_t$ reversed. **Property Statistics**: This includes the calculation of Wasserstein distances for three properties—density, formation energy, and elemental count—between the generated and test structures, denoted as d_ρ, d_E, and d_{elem}, respectively. The validity and coverage metrics are based on 10,000 generated samples, whereas the property statistics are derived from 1,000 samples that passed the validity check.

Results. Our method, EquiCSP, exhibits outstanding performance across multiple metrics, as detailed in Table 3. Notably, EquiCSP achieves competitive results in validity and coverage precision, underscoring the high quality of the samples it generates. Additionally, it delivers robust coverage recall, demonstrating the diversity of the structures produced. In the realm of property metrics, EquiCSP excels by significantly reducing the density distance d_ρ, influenced by the volume of the generated lattice, and the formation energy distance d_E, which relates to the atomic configuration. These achievements in minimizing key distances underscore the effectiveness of our symmetry-aware processing approach.

F. Visualizations

In this section, we present additional visualizations of the predicted structures from EquiCSP and the second best method DiffCSP in Figure 6. Our EquiCSP provides more accurate predictions compared with DiffCSP.

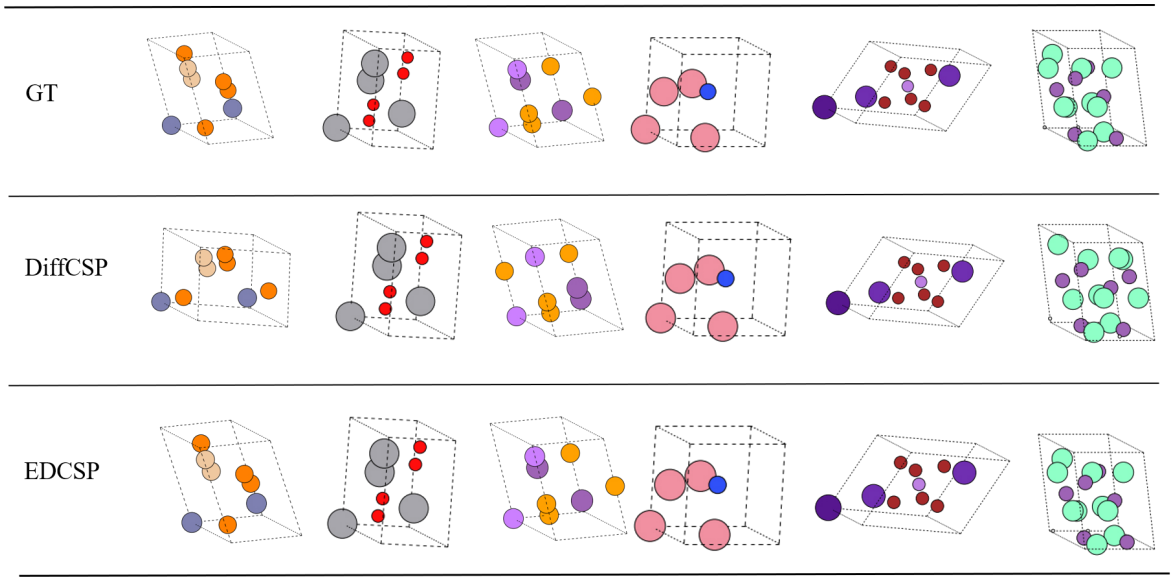

Figure 6. Additional visualizations of the predicted structures. We translate the same atom to the origin for better visualization and comparison.