OPEN MATERIALS GENERATION WITH STOCHASTIC INTERPOLANTS

Philipp Höllmer[1,*], Thomas Egg[1,*], Maya M. Martirossyan[1,*], Eric Fuemmeler[2,*], Zeren Shui[2], Amit Gupta[2], Pawan Prakash[3], Adrian Roitberg[3], Mingjie Liu[3], George Karypis[2], Mark Transtrum[4], Richard G. Hennig[3], Ellad B. Tadmor[2], Stefano Martiniani[1,†]
[1]New York University [2]University of Minnesota [3]University of Florida [4]Brigham Young University

ABSTRACT

The discovery of new materials is essential for enabling technological advancements. Computational approaches for predicting novel materials must effectively learn the manifold of stable crystal structures within an infinite design space. We introduce Open Materials Generation (OMatG), a unifying framework for the generative design and discovery of inorganic crystalline materials. OMatG employs stochastic interpolants (SI) to bridge an arbitrary base distribution to the target distribution of inorganic crystals via a broad class of tunable stochastic processes, encompassing both diffusion models and flow matching as special cases. In this work, we adapt the SI framework by integrating an equivariant graph representation of crystal structures and extending it to account for periodic boundary conditions in unit cell representations. Additionally, we couple the SI flow over spatial coordinates and lattice vectors with discrete flow matching for atomic species. We benchmark OMatG's performance on two tasks: crystal structure prediction (CSP) for specified compositions, and *de novo* generation (DNG) aimed at discovering stable, novel, and unique structures. In our ground-up implementation of OMatG, we refine and extend both CSP and DNG metrics compared to previous works. OMatG establishes a new state-of-the-art in generative modeling for materials discovery, outperforming purely flow-based and diffusion-based implementations. These results underscore the importance of designing flexible deep learning frameworks to accelerate progress in materials science. The OMatG code is available at `https://github.com/FERMat-ML/OMatG`.

1 INTRODUCTION

A core objective of materials science is the discovery of new synthesizable structures and compounds with the potential to meet critical societal demands. The development of new materials such as room-temperature superconductors (Boeri et al., 2022), high-performance alloys with exceptional mechanical properties (Gludovatz et al., 2014; 2016; George et al., 2019), advanced catalysts (Strmcnik et al., 2016; Nakaya & Furukawa, 2023), and materials for energy storage and generation (Liu et al., 2010; Snyder & Toberer, 2008) holds the potential to drive technological revolutions.

Exploring the vast compositional and structural landscape of multicomponent materials with novel properties is essential, yet exhaustive experimental screening is infeasible (Cantor, 2021). Quantum and classical molecular simulation offer a powerful alternative, enabling a more targeted and efficient exploration. In recent decades, both experimental (Potyrailo et al., 2011; Maier, 2019) and computational (Jain et al., 2011; Curtarolo et al., 2013) high-throughput pipelines have led to a proliferation of materials databases for crystal structures (Bergerhoff et al., 1983; Mehl et al., 2017) and simulations (Blaiszik et al., 2016; Vita et al., 2023; Fuemmeler et al., 2024). These advances have already facilitated the development of more accurate machine-learned interatomic potentials (Batzner et al., 2022; Batatia et al., 2022; Chen & Ong, 2022).

Still, efficiently sampling the manifold of stable materials structures under diverse constraints—such as composition and target properties—remains a major challenge. Traditional approaches to

[*]These authors contributed equally.
[†]Corresponding author.

materials discovery have relied on first-principles electronic structure methods such as DFT—or more sophisticated theory, depending on the property (Booth et al., 2013; Zaki et al., 2014; Isaacs & Marianetti, 2020)—which, while powerful and fairly accurate, are very computationally expensive. These methods include *ab initio* random structure searching (AIRSS) (Pickard & Needs, 2011) or genetic algorithms for structure and phase prediction (Tipton & Hennig, 2013), both of which have successfully predicted new crystal structures and some of which have even been experimentally realized (Oganov et al., 2019). However, the high computational cost of these approaches has limited the scope and speed of material exploration, highlighting the need for cutting-edge ML techniques to significantly accelerate the discovery of stable inorganic crystalline materials.

1.1 Related works

Recent advances in machine learning techniques have generated significant interest in applying data-driven approaches for inorganic materials discovery. Among these, Graph Networks for Materials Exploration (GNoME) has demonstrated remarkable success by coupling coarse sampling strategies for structure and composition with AIRSS that leverages a highly accurate machine-learned interatomic potential (MLIP) to predict material stability, leading to the identification of millions of new candidate crystal structures (Merchant et al., 2023). Other frameworks have approached the generation of composition and structure jointly through fully ML-based methods. Crystal Diffusion Variational Autoencoder (CDVAE) leverages variational autoencoders and a graph neural network representation to sample new crystal structures from a learned latent space (Xie et al., 2022). To date, state-of-the-art performance in both crystal structure prediction for given compositions and *de novo* generation of novel stable materials has been achieved by diffusion models such as DiffCSP (Jiao et al., 2023) and MatterGen (Zeni et al., 2025), as well as conditional flow-matching frameworks such as FlowMM (Miller et al., 2024).

While these approaches have demonstrated that ML can push the boundaries of computational materials discovery, it remains uncertain whether score-based diffusion or flow-matching represent the definitive methodological frameworks for this problem. Furthermore, the extent to which the optimal approach depends on the training data remains an open question. Thus far, each new method has typically outperformed its predecessors across datasets.

1.2 Our contribution

The work we present in this paper is the first implementation and extension of the stochastic interpolants (SIs) framework (Albergo et al., 2023) for the modeling and generation of inorganic crystalline materials. SIs are a unifying framework that encompasses both flow-based and diffusion-based methods as specific instances, while offering a more general and flexible framework for generative modeling. In this context, SIs define a stochastic process that interpolates between pairs of samples from a known base distribution and a target distribution of inorganic crystals. By learning the drift terms of either ordinary differential equations (ODEs) or stochastic differential equations (SDEs), new samples can be generated by numerically integrating these equations. The flexibility of the SI framework stems from the ability to tailor the choice of interpolants, and the incorporation of an additional random latent variable, further enhancing its expressivity. With their many degrees of freedom, SIs thus provide an ideal framework for optimizing generative models for materials design.

We implement the SI framework in the open-source Open Materials Generation (OMatG) package, released alongside this paper. OMatG allows to train and benchmark models for two materials generation tasks: *Crystal structure prediction* (CSP) which only learns to generate atomic positions and lattice vectors for a given composition, and *de novo generation* (DNG) which learns to generate both crystal structure and composition to predict novel materials. We discover that optimizing interpolation schemes for different degrees of freedom substantially improves performance across diverse datasets. As a result, our approach achieves a new state of the art—outperforming both DiffCSP (Jiao et al., 2023) and FlowMM (Miller et al., 2024)—in CSP and DNG across all evaluated datasets for both existing, revised, and new performance measures.

2 BACKGROUND

2.1 DIFFUSION MODELS

A widely used approach in generative modeling uses diffusion models (Sohl-Dickstein et al., 2015), which define a stochastic process that progressively transforms structured data into noise. A model is then trained to approximate the reverse process, enabling the generation of new samples. These models are often formulated in terms of SDEs, where the forward process follows a predefined diffusion dynamic.

Score-based diffusion models (SBDMs) are an instantiation of diffusion models that learn a score function—the gradient of the log probability density—to guide the reversal of the diffusion process via numerical integration (Song et al., 2021). SBDMs have demonstrated remarkable success in generating high-quality and novel samples across a wide range of applications where the target distribution is complex and intractable, such as photorealistic image generation (Saharia et al., 2022) and molecular conformation prediction (Corso et al., 2023).

2.2 CONDITIONAL FLOW MATCHING

Conditional flow matching (CFM) (Lipman et al., 2023) is a generative modeling technique that learns a flow which transports samples from a base distribution at time $t = 0$ to a target distribution at time $t = 1$. This process defines a probability path that describes how samples are distributed at any intermediate time $t \in [0, 1]$. The velocity field associated with this flow governs how individual samples evolve over time. CFM learns the velocity indirectly by constructing conditional vector fields that are known *a priori*. Once trained, samples drawn from the base distribution can be evolved numerically to generate new samples from the target distribution. Originally, CFM was formulated using Gaussian conditional probability paths, but Tong et al. (2024) later extended this framework to allow for arbitrary probability paths and couplings between base and target distributions. A further extension, particularly relevant to physics and chemistry, is Riemannian flow matching (RFM), which generalizes CFM to Riemannian manifolds (Chen & Lipman, 2024). This allows in particular to use the flow-matching framework for systems with periodic boundary conditions as they appear in unit cell representations of inorganic crystals (Miller et al., 2024).

3 OPEN MATERIALS GENERATION

3.1 STOCHASTIC INTERPOLANTS

SIs provide a unifying mathematical framework for generative modeling, generalizing both SBDMs and CFM (Albergo et al., 2023). The SI $x(t, x_0, x_1, z)$ bridges the base distribution ρ_0 with a target distribution ρ_1 by learning a time-dependent drift $b^\theta(t, x)$. In this work, we focus on stochastic interpolants of the form:

$$x_t \equiv x(t, x_0, x_1, z) = \alpha(t)x_0 + \beta(t)x_1 + \gamma(t)z. \tag{1}$$

Here, $t \in [0, 1]$ represents time and (x_0, x_1) are paired samples drawn from ρ_0 and ρ_1, respectively. The random variable z is drawn from a standard Gaussian $\mathcal{N}(0, \boldsymbol{I})$ independently of x_0 and x_1. The functional forms of α, β, and γ are flexible, subject to few constraints (see Appendix A.2). The inclusion of the latent variable $\gamma(t)z$ allows sampling of an ensemble of paths around the mean interpolant $I(t, x) = \alpha(t)x_0 + \beta(t)x_1$, and is theorized to improve generative modeling by promoting smoother and more regular learned flows (Albergo et al., 2023).

The time-dependent density $\rho(t)$ of the stochastic process x_t in Eq. (1) can also be realized either *via* deterministic sampling through an ODE (derived from a transport equation) or stochastic sampling through an SDE (derived from a Fokker–Planck equation) only requiring $x_0 \sim \rho_0$ (see Appendix A.1). This enables generative modeling by evolving samples from a known base distribution ρ_0 to the target distribution ρ_1. For both ODE- and SDE-based sampling, the required drift term $b^\theta(t, x) : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$ is learned by minimizing the loss function

$$\mathcal{L}_b(\theta) = \mathbb{E}_{t,z,x_0,x_1}\left[|b^\theta(t, x_t)|^2 - 2\,\partial_t x(t, x_0, x_1, z) \cdot b^\theta(t, x_t)\right], \tag{2}$$

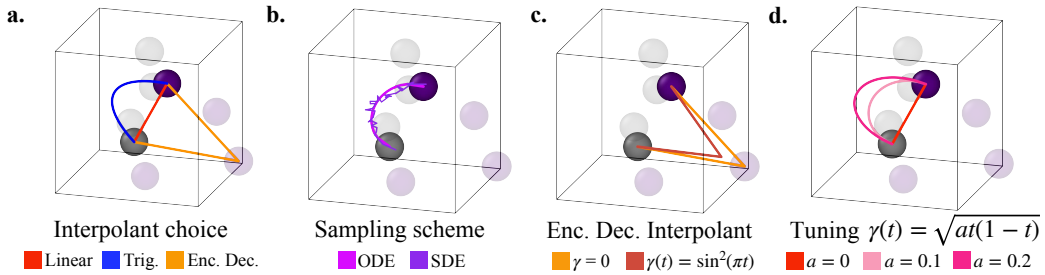

Figure 1: Visualization of the tunable components of the SI framework for bridging samples x_0 (gray particles) and x_1 (purple particles). Interpolation paths are shown only for one pair of highlighted particles. **(a)** The choice of the interpolant changes the path of the time-dependent interpolation trajectory. **(b)** During inference, the learned drift term $b^\theta(t, x)$ and denoiser $z^\theta(t, x)$ generate new samples *via* ODE or SDE integration, here for a linear interpolant with $\gamma = \sqrt{0.07t(1-t)}$. **(c)** The inclusion of a latent variable $\gamma(t)z$ changes the interpolation path. **(d)** The function $\gamma(t, a) = \sqrt{at(1-t)}$ depends on a that also influences the interpolation path.

where the expectation is taken independently over $t \sim \mathcal{U}(0, 1)$ with $\mathcal{U}(0, 1)$ the uniform distribution between 0 and 1, $z \sim \mathcal{N}(0, \boldsymbol{I})$, $x_0 \sim \rho_0$, and $x_1 \sim \rho_1$. For SDE-based sampling, an additional denoiser $z^\theta(t, x) : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$ must be learned by minimizing an additional loss

$$\mathcal{L}_z^\theta(\theta) = \mathbb{E}_{t,z,x_0,x_1}\left[|z^\theta(t, x_t)|^2 - 2\, z^\theta(t, x_t) \cdot z\right]. \tag{3}$$

The drift term, along with the denoiser in the case of SDE-based sampling, enables the generation of samples from the target distribution (Albergo et al., 2023). Note that minimizing with respect to these loss functions amounts to minimizing with respect to a mean-squared error loss function (see Appendix B.2). For ODE-based sampling, $\gamma(t) = 0$ is a possible choice. However, for SDE-based sampling, $\gamma(t) > 0$ is required for all $t \in (0, 1)$ (see Appendix A.1).

By appropriately selecting interpolation functions α, β, γ and choosing between deterministic (ODE) and stochastic (SDE) sampling schemes, the SI framework not only recovers CFM and SBDM as special cases but also enables the design of a broad class of novel generative models (see Appendix A.2 for examples). The strength of OMatG's SI implementation for materials discovery lies in its ability to tune both the interpolation and sampling schemes, as illustrated in Fig. 1 for a pair of structures sampled from ρ_0 and ρ_1. By systematically optimizing over this large design space, we achieve superior performance for CSP and DNG tasks across datasets, as discussed in Section 5.

3.2 Crystal representation and generation

A crystalline material is defined by its idealized repeat unit, or unit cell, which encodes its periodicity. In the OMatG representation, a unit cell is described by separating the material's chemical composition—given by its atomic species $\boldsymbol{A} \in \mathbb{Z}_{>0}^N$, where N is the number of atoms in the unit cell—from its structural representation—its fractional coordinates $\boldsymbol{X} \in [0, 1)^{3\times N}$ with periodic boundaries and lattice vectors $\boldsymbol{L} \in \mathbb{R}^{3\times 3}$. During training, all three components $\{\boldsymbol{A}, \boldsymbol{X}, \boldsymbol{L}\}$ are considered simultaneously. We apply the SI framework only to the continuous structural representations $\{\boldsymbol{X}, \boldsymbol{L}\}$ with loss functions defined in Eqs (2) and (3), and use discrete flow matching (DFM) on the chemical species $\boldsymbol{A}$ (Gat et al., 2024). The number of atoms N in the structure x_0 sampled from the base distribution ρ_0 is determined by the number of atoms in the corresponding structure x_1 sampled from the target distribution ρ_1.

3.2.1 Atomic coordinates

We specify the base distribution for the fractional coordinates $\boldsymbol{x} \in [0, 1)$ for all $\boldsymbol{x} \in \boldsymbol{X}$ *via* a uniform distribution (except for the score-based diffusion interpolant that requires a wrapped normal distribution $\rho_0(\boldsymbol{x})$ following the approach of Jiao et al. (2023); see Section 4.1). For treating fractional coordinates, we implement a variety of periodic interpolants that connect the base to the target data distributions. To satisfy periodic boundary conditions on the paths defined by the interpolants, we

extend the SI framework to the surface of a four-dimensional torus in this paper. Reminiscent of RFM (Chen & Lipman, 2024), the linear interpolant on the torus traverses a path equivalent to the shortest-path geodesic which is always[1] well-defined. Other interpolants, however, are more complex. In order to uniquely define them, we always define the interpolation with respect to the shortest-path geodesic. That is, for interpolation between $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ with a periodic boundary at 0 and 1, we first unwrap $\boldsymbol{x}_1$ to the periodic image $\boldsymbol{x}_1'$ which has the shortest possible distance from $\boldsymbol{x}_0$. Following this, the interpolation between $\boldsymbol{x}_0$ and $\boldsymbol{x}_1'$ is computed given a choice of interpolant, and the traversed path is wrapped back into the boundary from 0 to 1. This approach is illustrated in Appendix A.4.

3.2.2 LATTICE VECTORS

Lattice vectors $\boldsymbol{L}$ are treated with a wide range of (non-periodic) stochastic interpolants (see Section 4.1 again). To construct the base distribution, we follow Miller et al. (2024) and construct an informative base distribution $\rho_0(\boldsymbol{L})$ by combining a uniform distribution over the lattice angles with a log-normal distribution fitted to the empirical distribution of the lattice lengths in each target dataset. This choice brings the base distribution closer to the target distribution. Unlike SBDM, which requires a Gaussian base distribution, the SI framework allows such flexibility. Importantly, the model still has to learn to generate a joint, correlated distribution of lattice vectors, fractional coordinates, and atomic species.

3.2.3 ATOMIC SPECIES

The discrete nature of chemical compositions $\boldsymbol{A}$ in atomic crystals requires a specialized approach for generative modeling. To address this, we implement discrete flow matching (DFM) (Campbell et al., 2024). In our implementation of the DFM framework, each atomic species $\boldsymbol{a} \in \boldsymbol{A}$ can take values in $\{1, 2, \ldots, 100\} \cup \{M\}$; where $\{1-100\}$ are atomic element numbers and M is a masking token used during training. The base distribution is defined as $\rho_0(\boldsymbol{a}) = [M]^N$, meaning that initially all N atoms are masked. As sampling progresses, the identities of the atoms evolve *via* a continuous-time Markov Chain (CTMC), and are progressively unmasked to reveal valid atomic species. At $t = 1$, all masked tokens are replaced. To learn this process, we define a conditional flow $p_{t|1}(\boldsymbol{a}_t|\boldsymbol{a}_1)$ that linearly interpolates in time from the fully masked state $\boldsymbol{a}_0$ toward $\boldsymbol{a}_1$ and thus yields the composition $\boldsymbol{a}_t$ of the interpolated structure x_t. Based on these conditional flows, a neural network is trained to approximate the denoising distribution $p^{\theta}_{1|t}(\boldsymbol{a}_1|x_t)$, which yields the probability for the composition $\boldsymbol{a}_1$ given the entire structure x_t, by minimizing a cross-entropy loss

$$\mathcal{L}_{\mathrm{DFM}}(\theta) = \mathbb{E}_{t,x_1,x_t}\left[\log p^{\theta}_{1|t}(\boldsymbol{a}_1|x_t)\right]. \tag{4}$$

In doing this, we are able to directly construct the marginal rate matrix $R^{\theta}_t(\boldsymbol{a}_t, i)$ for the CTMC that dictates the evolution of $\boldsymbol{a}_t$ at time t to the next time step during generation (see Appendix A.5). It is important to note that the learned probability path is a function of the entire atomic configuration $\{\boldsymbol{A}, \boldsymbol{X}, \boldsymbol{L}\}$ which is necessary for the prediction of chemical composition from structure.

3.3 JOINT GENERATION WITH STOCHASTIC INTERPOLANTS

For both CSP and DNG tasks, we seek to generate samples from a joint distribution over multiple coordinates. For DNG, this joint distribution ρ_1 encompasses all elements of a crystal unit cell. For CSP we similarly model the joint distribution, ρ_1, but with atom types fixed to compositions sampled from the target dataset. For both tasks, the total loss function is formulated as a weighted sum of the individual loss functions for each variable (see Appendix B.2), and their relative weights are optimized (see Appendix B.3). We illustrate both types of models and their structure generation process in Fig. 2.

Additionally, for DNG, we consider a two-step process in which composition is learned separately from structure, as seen in Fig. 2b. In this approach, we first train a Chemical Formula Prediction (CFP) model (see Appendix B.1) to generate compositions optimized for SMACT stability (Davies et al., 2019), similarity in the distribution of N-arity of known structures, as well as uniqueness and novelty. The predicted compositions are then used as input for a pretrained CSP model, which generates the corresponding atomic configurations.

[1]The only exception being when two points are precisely half the box length apart. However, this case is not relevant for the given base distribution.

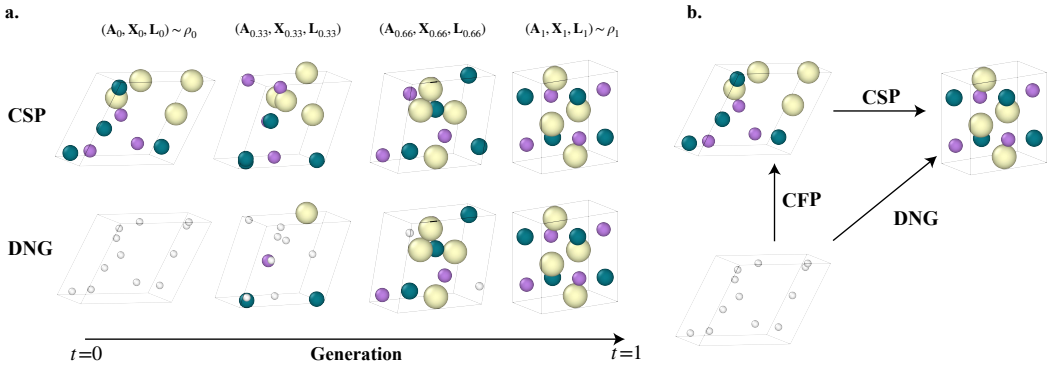

Figure 2: Illustration of CSP and DNG tasks. (**a**) For CSP, the species $\boldsymbol{A}$ are fixed with known compositions from $t = 0$. From this, we predict $\boldsymbol{X}$ and $\boldsymbol{L}$ from randomly sampled initial values. For DNG, we predict $(\boldsymbol{A}, \boldsymbol{X}, \boldsymbol{L})$ jointly. Our implementation of DFM initializes $\boldsymbol{A}$ as a sequence of masked particles that are unmasked through a series of discrete jumps to reveal a physically reasonable composition. (**b**) Two avenues for performing *de novo* generation of materials. The first uses two steps: a CFP model predicts compositions and then uses a CSP model to find accompanying stable structures. The second trains a DNG model over cell, species, and fractional coordinates jointly as shown in (a).

4 METHODOLOGY

4.1 CHOICE OF INTERPOLANT

In training OMatG, we optimize the choice of the interpolating function that is used during training for the lattice vectors and the fractional coordinates (with periodic boundary conditions). Albergo et al. (2023) introduced four interpolants of the form defined in Eq. (1), each shaping the interpolation trajectory differently (see also Appendix A.2 for further details).

The **linear** interpolant defines a constant velocity trajectory from x_0 to x_1. When combined with an ODE sampling scheme and $\gamma = 0$, this reproduces a particular choice of CFM. If the sampling scheme or γ is changed, however, the use of linear interpolants becomes distinct from CFM. The **trigonometric** interpolant prescribes trajectories with more curvature than the linear interpolant. The **encoder-decoder** interpolant first evolves samples from ρ_0 at $t = 0$ to follow an intermediate Gaussian distribution at a switch time T_{switch}, before mapping them to samples from the target distribution ρ_1 at $t = 1$. This approach has been found to interpolate more smoothly between distributions, potentially mitigating the formation of spurious features in the probability path at intermediate times (Albergo et al., 2023). Lastly, we consider the variance preserving **score-based diffusion (SBD)** interpolant. When paired with an SDE sampling scheme, this interpolant is mathematically equivalent to an SBDM, but on the continuous time interval $[0, 1]$. The SBD interpolant assumes that ρ_0 is a Gaussian, and unlike the previous three interpolants it involves no explicit latent variable; instead the $\alpha(t)x_0$ term takes on this role. We note that the trajectory of the encoder-decoder interpolant between times $t = T_{\text{switch}}$ and $t = 1$ resembles that of the SBD interpolant between times $t = 0$ and $t = 1$. For the example of using the encoder-decoder interpolant only for the coordinates, however, we emphasize that the Gaussian-distributed coordinates at $t = T_{\text{switch}}$ are conditioned on other coordinates that are partially interpolated at this point. Conversely, for SBD interpolation, the Gaussian distributed coordinates at $t = 0$ are only conditioned on other random variables since, at this point, all elements of x_0 are randomly distributed.

To investigate how different interpolants affect generative performance, we consider all interpolants outlined above for both the atomic positions $\boldsymbol{X}$ and the lattice vectors $\boldsymbol{L}$. We noted that learning accurate drifts and denoisers for the atomic positions, $\boldsymbol{X}$, is more challenging than for the other degrees of freedom. Accordingly, we optimize all hyperparameters—including the choice of interpolant for $\boldsymbol{L}$—separately for each interpolant applied to $\boldsymbol{X}$. This results in a set of experiments specific to the position interpolants, where the best performing lattice interpolant may vary.

4.2 EQUIVARIANT REPRESENTATION OF CRYSTAL STRUCTURES

Imposing inductive biases on the latent representation of the crystal structure can promote data efficiency and improve learning. The CSPNet architecture (Jiao et al., 2023), originally adopted in DiffCSP, is an equivariant graph neural network (EGNN) (Satorras et al., 2021) that produces a permutation- and rotation-equivariant, as well as translation-invariant representation of the crystal structures.

In the current OMatG implementation, we employ CSPNet as an encoder that is trained from scratch. The CSPNet architecture encodes atomic types using learnable atomic embeddings and represents fractional coordinates through sinusoidal positional encodings (see Appendix B.1). These features are processed through six layers of message-passing, after which the encoder produces the drift $b^\theta(t,x)$ of both the lattice and the fractional coordinates, as well as potentially predicting the denoiser $z^\theta(t,x)$. For DNG, the network must also predict $\log p^\theta_{1|t}(\boldsymbol{a}_1|x_t)$. The resulting outputs inherently preserve the permutation, rotational, and translational symmetries embedded in CSPNet.

The output of CSPNet is invariant with respect to translations of the fractional coordinates in the input. Thus, one should, in principle, use a representation of the fractional coordinates that does not contain any information about translations. While this is straightforward in Euclidean space by removing the mean of the coordinates of the given structure, this cannot be done with periodic boundary conditions where the mean is not uniquely defined. We follow Miller et al. (2024) and instead remove the center-of-mass motion when computing the ground-truth $\partial_t x(t, x_0, x_1, z)$ in Eq. (2).

Alternative EGNNs such as NequIP (Batzner et al., 2022), M3GNet (Chen & Ong, 2022), or MACE (Batatia et al., 2022) which have been widely used for the development of MLIPs can also serve as plug-and-play encoders within OMatG's SI framework. Integrating different architectures is a direction that we plan to explore in future iterations of the framework.

4.3 COMPARISON TO OTHER FRAMEWORKS

We compare our results to existing DiffCSP and FlowMM models. We detail in Section 5 how we improve the extant benchmarks used in the field and therefore recompute all CSP and DNG benchmarks for these models. In nearly all cases, we were able to generate structures using the DiffCSP and FlowMM source code whose metrics closely matched (within $\sim$1%) the previously reported metrics in their respective manuscripts. The observed differences can be attributed to the use of a newer version of SMACT composition rules[2] (Davies et al., 2019) and to natural fluctuations arising from model retraining.

Since the focus of this work is to assess our model's ability to learn unconstrained and unconditioned flows, we do not compare against symmetry-constrained generation methods (AI4Science et al., 2023; Cao et al., 2024; Zhu et al., 2024; Kazeev et al., 2024; Jiao et al., 2024). Symmetry constraints can be incorporated in future extensions of the flexible OMatG framework.

5 EXPERIMENTS

5.1 PERFORMANCE METRICS

We assess the performance of OMatG's and competing models using a variety of standard, refined, and contributed benchmarks. For full details of all metrics, see Appendix C. Briefly, for the CSP task, the most commonly computed metrics are the match rate and root mean square error (RMSE) between matched generated and reference structures. Match rate is defined as the percentage of generated structures that match those found in the reference set, where matching is computed according to Pymatgen's `StructureMatcher` module (Ong et al., 2013).

For the DNG task, a variety of validity, coverage, property, and stability metrics are computed. Among them, the S.U.N. (stable, unique, and novel) rate is perhaps the most important as it represents the model's ability to propose new candidate materials. The S.U.N. rate is defined as the percentage of generated structures that are *stable* with respect to a reference convex hull (within 0.1 eV/atom), are not found within the reference set (*novel*), and are not duplicated within the generated set itself

[2]The SMACT Python library updated its default oxidation states with the release of version 3.0.

Table 1: Results from crystal structure prediction. Match rate and RMSE of matched structures without (left) and with (right) filtering for structural and compositional validity are reported for all models. For OMatG's model, the choice of positional interpolant, latent variable component γ, and sampling scheme are noted.

Method	***perov-5***		***MP-20***		***MPTS-52***		***Alex-MP-20***	
	Match (%) ↑	RMSE ↓	Match (%) ↑	RMSE ↓	Match (%) ↑	RMSE ↓	Match (%) ↑	RMSE ↓
DiffCSP	53.08/51.94	**0.0774/0.0775**	57.82/52.51	**0.0627/0.0600**	15.79/14.29	**0.1533**/0.1489	-	
FlowMM	53.63/51.86	0.1025/0.0994	66.22/59.98	0.0661/0.0629	22.29/20.28	0.1541/**0.1486**	-	
OMatG	**83.06/81.27**	0.3753/0.3755	**69.83/63.75**	0.0741/0.0720	**27.38/25.15**	0.1970/0.1931	**69.44/61.38**	**0.1303/0.1289**
	SBD; ODE; $\sigma = 0.28$		Linear; ODE; $\gamma = 0$		Linear; ODE; $\gamma = 0$		Linear; ODE; $\gamma = 0$	

(*unique*). The stability-based metrics are calculated using MatterGen's code base (MatterGen, 2025), in which the machine-learned interatomic potential MatterSim (Yang et al., 2024) is utilized for structural relaxation. This requires significantly less compute time in comparison to running DFT relaxations and is shown to correlate well with DFT predictions.

5.2 Benchmarks and datasets

We use the following datasets to benchmark the OMatG model: ***perov-5*** (Castelli et al., 2012), a dataset of perovskites with $18\,928$ samples with five atoms per unit cell in which only lattice lengths and atomic types change; ***MP-20*** (Jain et al., 2013; Xie et al., 2022) from the Materials Project that contains $45\,231$ structures with a maximum of $N = 20$ atoms per unit cell, and ***MPTS-52*** (Baird et al., 2024) which is a time split of the Materials Project with $40\,476$ structures with up to $N = 52$ atoms per unit cell and is typically the most difficult to learn. We use the same 60-20-20 splits as Xie et al. (2022); Jiao et al. (2023); Miller et al. (2024). Additionally, we consider the ***Alex-MP-20*** dataset (Zeni et al., 2025), where we used an 80-10-10 split constructed from MatterGen's 90-10 split, in which we removed 10% of the training data to create a test dataset. This dataset contains $675\,204$ structures with 20 or fewer atoms per unit cell from the *Alexandria* (Schmidt et al., 2022a;b) and *MP-20* datasets. We do not include the *carbon-24* dataset (Pickard, 2020) in our results, as the match rate metric is ill-defined for this dataset; because all elements are carbon, it is not clear how many generated structures are unique and producing a structure that matches one in the reference dataset is trivial.[3]

5.3 Results

We report performance for the CSP task in Tab. 1 and for the DNG task in Tabs 2 and 3 for DiffCSP, FlowMM, and OMatG models. For the CSP task, OMatG significantly outperforms previous approaches on all datasets. We emphasize the particularly strong performance of the SBD and trigonometric positional interpolants with ODE sampling schemes in achieving high match rates for the *perov-5* dataset as shown in Tabs 1 and 5. Likewise for DNG metrics, OMatG shows state-of-the-art performance for multiple positional interpolants. Figure 3 compares the distributions of the average energies above the hull for generated structures, exhibiting OMatG's superior performance for the generation of stable structures. OMatG consistently produces lower energy structures compared to previous models, and they are also generated closer to their relaxed configuration. This, together with high novelty rates, begets improved S.U.N. rates.

[3]Previous papers (Xie et al., 2022; Jiao et al., 2023; Miller et al., 2024) report match rate for *carbon-24*, but they do not compare the generated structure to the entirety of the reference dataset; their results suggest the match tolerance is larger than the differences between the *carbon-24* structures.

[4]We do not bold any values in the structural validity category as the CDVAE model reports the state-of-the-art with 100% structural validity. For the Wasserstein distances of the density and Nary distributions, we only bold values lower than 0.075 and 0.079 respectively, as these were the values reported by FlowMM for their model with 500 integration steps (not included in this table).

[5]OMatG's performance is comparable to the 22.56% S.U.N. Rate and 0.11Å RMSD reported by the MatterGen-MP model when trained on *MP-20* Zeni et al. (2025), though we cannot directly compare to these benchmarks as those structures are relaxed using DFT.

Table 2: Results from *de novo* generation of 10,000 structures with models trained on the *MP-20* dataset. The integration steps for OMatG is chosen based on best overall performance. For OMatG's model, the choice of positional interpolant, latent variable component γ, and sampling scheme are noted. Best scores in each category are bolded.[4]

Method	**Integration steps**	**Validity** (% ↑)			**Coverage** (% ↑)		**Property** (↓)		
		Structural	Composition	Combined	Recall	Precision	wdist (ρ)	wdist (Nary)	wdist ($\langle CN \rangle$)
DiffCSP	1000	99.91	82.68	82.65	**99.67**	**99.63**	0.3133	0.3193	0.3053
FlowMM	1000	92.26	83.11	76.94	99.34	99.02	1.0712	0.1130	0.4405
OMatG									
Linear (SDE); $\gamma = \sqrt{0.018t\,(1-t)}$	710	99.04	**83.40**	**83.40**	99.47	98.81	0.2583	**0.0418**	0.4066
Trig (ODE); $\gamma = \sqrt{0.027t\,(1-t)}$	680	95.05	82.84	82.84	99.33	94.75	**0.0607**	**0.0172**	**0.1650**
Enc-Dec (ODE); $\gamma = \sin^2 \pi t$	840	97.25	**86.35**	**84.19**	99.62	99.61	0.1155	**0.0553**	**0.0465**
SBD (SDE); $\sigma = 7.14$	870	93.38	80.66	80.66	98.95	92.76	0.1865	**0.0768**	**0.1637**
CFP + CSP (Linear, $\gamma = 0$)	130+210	97.95	79.68	78.21	**99.67**	99.50	0.5614	0.2008	0.6256

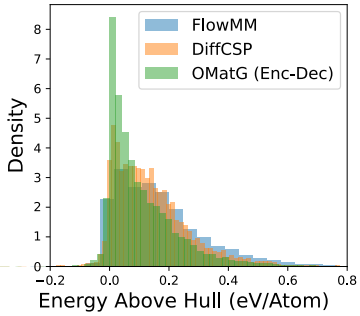

Figure 3: Histogram of the computed energies above the convex hull for structures generated by FlowMM, DiffCSP, and OMatG (Enc-Dec interpolant). The OMatG model consistently produces lower energy structures compared to FlowMM and DiffCSP. See Appendix C.3 for calculation details.

Method	$\langle E \rangle / N$ (↓) above hull	RMSD (Å, ↓)	Novelty Rate (%, ↑)	Stability Rate (%, ↑)	S.U.N. Rate (%, ↑)
DiffCSP	0.1984	1.295	72.73	43.04	19.00
FlowMM	0.2509	0.651	72.76	37.47	13.86
OMatG					
Linear	**0.1823**	**0.615**	72.00	**45.00**	**22.07**
Trig	**0.1857**	0.657	65.35	**51.40**	**19.96**
Enc-Dec	**0.1699**	**0.390**	54.97	**58.56**	17.59
SBD	0.2189	0.763	**75.80**	42.60	**22.10**
CFP + CSP	0.2340	**0.488**	**75.85**	42.21	**20.50**

Table 3: Stability (defined as $\leq$ 0.1 eV/atom above hull), uniqueness, and novelty results from *de novo* generation on the MP-20 dataset computed for the same models as in Tab. 2. All evaluations are performed with the MatterGen code base (MatterGen, 2025) with respect to the included reference *Alex-MP-20* dataset. The average RMSD is between the generated and the relaxed structures, and the average energy above hull is reported in units of eV/atom.[5]

OMatG also outperforms FlowMM in settings where large language models are used as base distributions Sriram et al. (2024), with results shown in Appendix D.

6 Discussion and Conclusions

We demonstrate OMatG's capabilities for learning the manifold of stable materials and generating novel structures. The match rates for the CSP task on the *perov-5* dataset achieved using the SBD and trigonometric positional interpolants, as shown in Tabs 1 and 5, greatly surpass (by 1.6×) the match rates of previous models. We also outperform previous models' match rate for CSP on the *MP-20* dataset with the linear (both ODE and SDE sampling schema) and the trigonometric positional interpolants; similarly, the fraction of stable, novel, and unique structures is highest across almost all positional interpolants. Our results and hyperparameters are further discussed in Appendix B.3.

In conclusion, we show state-of-the-art performance of OMatG across nearly all benchmarks for both crystal structure prediction as well as *de novo* generation of stable, novel, and unique structures. In our discussion in Appendix B.3, we highlight several ways in which hyperparameters and interpolation paths influence learning, though a more systematic theoretical analysis is needed to fully understand why specific choices affect model performance. Future work can also extend the flexibility of OMatG to additional interpolating functions. We underscore the importance of flexible ML frameworks like OMatG, which can adapt to different types of materials datasets by optimizing the generative model accordingly. Our work represents a key step forward in applications of machine-learning methods to materials discovery.

ACKNOWLEDGEMENTS

The authors thank Shenglong Wang at NYU IT High Performance Computing for his resourcefulness and valuable support. The authors acknowledge funding from NSF Grant OAC-2311632. P. H. and S. M. also acknowledge support from the Simons Center for Computational Physical Chemistry (Simons Foundation grant 839534, MT). The authors gratefully acknowledge the computational resources and consultation support that have contributed to the research results reported in this publication, provided by: IT High Performance Computing at New York University; the Empire AI Consortium; the Minnesota Supercomputing Institute (http://www.msi.umn.edu) at the University of Minnesota; UFIT Research Computing (http://www.rc.ufl.edu) and the NVIDIA AI Technology Center at the University of Florida in part through the AI and Complex Computational Research Award; Drexel University through NSF Grant OAC-2320600.

REFERENCES

Mila AI4Science, Alex Hernandez-Garcia, Alexandre Duval, Alexandra Volokhova, Yoshua Bengio, Divya Sharma, Pierre Luc Carrier, Yasmine Benabed, Michał Koziarski, and Victor Schmidt. Crystal-GFN: Sampling crystals with desirable properties and constraints, December 2023. URL http://arxiv.org/abs/2310.04925.

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic Interpolants: A Unifying Framework for Flows and Diffusions, November 2023. URL http://arxiv.org/abs/2303.08797.

Michael S. Albergo, Mark Goldstein, Nicholas M. Boffi, Rajesh Ranganath, and Eric Vanden-Eijnden. Stochastic interpolants with data-dependent couplings, September 2024. URL http://arxiv.org/abs/2310.03725.

Sterling G. Baird, Hasan M. Sayeed, Joseph Montoya, and Taylor D. Sparks. Matbench-genmetrics: A Python library for benchmarking crystal structure generative models using time-based splits of Materials Project structures. *Journal of Open Source Software*, 9(97):5618, May 2024. ISSN 2475-9066. doi: 10.21105/joss.05618. URL https://joss.theoj.org/papers/10.21105/joss.05618.

Ilyes Batatia, David P. Kovacs, Gregor Simm, Christoph Ortner, and Gabor Csanyi. MACE: Higher Order Equivariant Message Passing Neural Networks for Fast and Accurate Force Fields. *Advances in Neural Information Processing Systems*, 35:11423–11436, December 2022. URL https://arxiv.org/abs/2206.07697.

Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P. Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E. Smidt, and Boris Kozinsky. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*, 13(1):2453, May 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-29939-5. URL https://www.nature.com/articles/s41467-022-29939-5.

G. Bergerhoff, R. Hundt, R. Sievers, and I. D. Brown. The inorganic crystal structure data base. *J. Chem. Inf. Comput. Sci.*, 23(2):66–69, May 1983. ISSN 0095-2338. doi: 10.1021/ci00038a003. URL https://doi.org/10.1021/ci00038a003.

J. Bergstra, D. Yamins, and D. D. Cox. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. *TProc. of the 30th International Conference on Machine Learning (ICML 2013)*, pp. pp. I–115 to I–123, 2013.

B. Blaiszik, K. Chard, J. Pruyne, R. Ananthakrishnan, S. Tuecke, and I. Foster. The Materials Data Facility: Data Services to Advance Materials Science Research. *JOM*, 68(8):2045–2052, August 2016. ISSN 1543-1851. doi: 10.1007/s11837-016-2001-3. URL https://doi.org/10.1007/s11837-016-2001-3.

Lilia Boeri, Richard Hennig, Peter Hirschfeld, Gianni Profeta, Antonio Sanna, Eva Zurek, Warren E Pickett, Maximilian Amsler, Ranga Dias, Mikhail I Eremets, Christoph Heil, Russell J Hemley, Hanyu Liu, Yanming Ma, Carlo Pierleoni, Aleksey N Kolmogorov, Nikita

Rybin, Dmitry Novoselov, Vladimir Anisimov, Artem R Oganov, Chris J Pickard, Tiange Bi, Ryotaro Arita, Ion Errea, Camilla Pellegrini, Ryan Requist, E K U Gross, Elena Roxana Margine, Stephen R Xie, Yundi Quan, Ajinkya Hire, Laura Fanfarillo, G R Stewart, J J Hamlin, Valentin Stanev, Renato S Gonnelli, Erik Piatti, Davide Romanin, Dario Daghero, and Roser Valenti. The 2021 room-temperature superconductivity roadmap. *J. Phys.: Condens. Matter*, 34(18):183002, March 2022. ISSN 0953-8984. doi: 10.1088/1361-648X/ac2864. URL https://dx.doi.org/10.1088/1361-648X/ac2864.

George H Booth, Andreas Grüneis, Georg Kresse, and Ali Alavi. Towards an exact description of electronic wavefunctions in real solids. *Nature*, 493(7432):365–370, 2013.

Avishek Joey Bose, Tara Akhound-Sadegh, Guillaume Huguet, Kilian Fatras, Jarrid Rector-Brooks, Cheng-Hao Liu, Andrei Cristian Nica, Maksym Korablyov, Michael Bronstein, and Alexander Tong. SE(3)-Stochastic Flow Matching for Protein Backbone Generation, April 2024. URL http://arxiv.org/abs/2310.02391.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative Flows on Discrete State-Spaces: Enabling Multimodal Flows with Applications to Protein Co-Design, June 2024. URL http://arxiv.org/abs/2402.04997.

B. Cantor. Multicomponent high-entropy Cantor alloys. *Progress in Materials Science*, 120:100754, July 2021. ISSN 0079-6425. doi: 10.1016/j.pmatsci.2020.100754. URL https://www.sciencedirect.com/science/article/pii/S0079642520301183.

Zhendong Cao, Xiaoshan Luo, Jian Lv, and Lei Wang. Space Group Informed Transformer for Crystalline Materials Generation, March 2024. URL https://arxiv.org/abs/2403.15734v2.

Ivano E. Castelli, David D. Landis, Kristian S. Thygesen, Søren Dahl, Ib Chorkendorff, Thomas F. Jaramillo, and Karsten W. Jacobsen. New cubic perovskites for one- and two-photon water splitting using the computational materials repository. *Energy & Environmental Science*, 5(10): 9034–9043, September 2012. ISSN 1754-5706. doi: 10.1039/C2EE22341D. URL https://pubs.rsc.org/en/content/articlelanding/2012/ee/c2ee22341d.

Chi Chen and Shyue Ping Ong. A universal graph deep learning interatomic potential for the periodic table. *Nature Computational Science*, 2(11):718–728, November 2022. ISSN 2662-8457. doi: 10.1038/s43588-022-00349-3. URL https://www.nature.com/articles/s43588-022-00349-3.

Ricky T. Q. Chen and Yaron Lipman. Flow Matching on General Geometries, February 2024. URL http://arxiv.org/abs/2302.03660.

Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking, February 2023.

Stefano Curtarolo, Gus L. W. Hart, Marco Buongiorno Nardelli, Natalio Mingo, Stefano Sanvito, and Ohad Levy. The high-throughput highway to computational materials design. *Nat. Mater.*, 12(3):191–201, March 2013. ISSN 1476-4660. doi: 10.1038/nmat3568. URL https://www.nature.com/articles/nmat3568.

Daniel W. Davies, Keith T. Butler, Adam J. Jackson, Jonathan M. Skelton, Kazuki Morita, and Aron Walsh. SMACT: Semiconducting Materials by Analogy and Chemical Theory. *Journal of Open Source Software*, 4(38):1361, June 2019. ISSN 2475-9066. doi: 10.21105/joss.01361. URL https://joss.theoj.org/papers/10.21105/joss.01361.

Eric Fuemmeler, Gregory Wolfe, Amit Gupta, Joshua A Vita, Ellad B Tadmor, and Stefano Martiniani. Advancing the colabfit exchange towards a web-scale data source for machine learning interatomic potentials. In *AI for Accelerated Materials Design-NeurIPS 2024*, 2024.

Alex M. Ganose, Hrushikesh Sahasrabuddhe, Mark Asta, Kevin Beck, Tathagata Biswas, Alexander Bonkowski, Joana Bustamante, Xin Chen, Yuan Chiang, Daryl Chrzan, Jacob Clary, Orion Cohen, Christina Ertural, Max Gallant, Janine George, Sophie Gerits, Rhys Goodall, Rishabh Guha, Geoffroy Hautier, Matthew Horton, Aaron Kaplan, Ryan Kingsbury, Matthew Kuner,

Bryant Li, Xavier Linn, Matthew McDermott, Rohith Srinivaas Mohanakrishnan, Aakash Naik, Jeffrey Neaton, Kristin Persson, Guido Petretto, Thomas Purcell, Francesco Ricci, Benjamin Rich, Janosh Riebesell, Gian-Marco Rignanese, Andrew Rosen, Matthias Scheffler, Jonathan Schmidt, Jimmy-Xuan Shen, Andrei Sobolev, Ravishankar Sundararaman, Cooper Tezak, Victor Trinquet, Joel Varley, Derek Vigil-Fowler, Duo Wang, David Waroquiers, Mingjian Wen, Han Yang, Hui Zheng, Jiongzhi Zheng, Zhuoying Zhu, and Anubhav Jain. Atomate2: Modular workflows for materials science. *ChemRxiv*, 2025. URL https://chemrxiv.org/engage/chemrxiv/article-details/678e76a16dde43c9085c75e9.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete Flow Matching, July 2024. URL http://arxiv.org/abs/2407.15595.

Easo P. George, Dierk Raabe, and Robert O. Ritchie. High-entropy alloys. *Nat. Rev. Mater.*, 4 (8):515–534, August 2019. ISSN 2058-8437. doi: 10.1038/s41578-019-0121-4. URL https://www.nature.com/articles/s41578-019-0121-4.

Bernd Gludovatz, Anton Hohenwarter, Dhiraj Catoor, Edwin H. Chang, Easo P. George, and Robert O. Ritchie. A fracture-resistant high-entropy alloy for cryogenic applications. *Science*, 345(6201): 1153–1158, September 2014. doi: 10.1126/science.1254581. URL https://www.science.org/doi/10.1126/science.1254581.

Bernd Gludovatz, Anton Hohenwarter, Keli V. S. Thurston, Hongbin Bei, Zhenggang Wu, Easo P. George, and Robert O. Ritchie. Exceptional damage-tolerance of a medium-entropy alloy CrCoNi at cryogenic temperatures. *Nat. Commun.*, 7(1):10602, February 2016. ISSN 2041-1723. doi: 10.1038/ncomms10602. URL https://www.nature.com/articles/ncomms10602.

R. W. Grosse-Kunstleve, N. K. Sauter, and P. D. Adams. Numerically stable algorithms for the computation of reduced unit cells. *Acta Cryst A*, 60(1):1–6, January 2004. ISSN 0108-7673. doi: 10.1107/S010876730302186X. URL //journals.iucr.org/paper?sh5006.

Nate Gruver, Anuroop Sriram, Andrea Madotto, Andrew Gordon Wilson, C. Lawrence Zitnick, and Zachary Ulissi. Fine-Tuned Language Models Generate Stable Inorganic Materials as Text, February 2024. URL http://arxiv.org/abs/2402.04379.

Ask Hjorth Larsen, Jens Jørgen Mortensen, Jakob Blomqvist, Ivano E Castelli, Rune Christensen, Marcin Dułak, Jesper Friis, Michael N Groves, Bjørk Hammer, Cory Hargus, Eric D Hermes, Paul C Jennings, Peter Bjerre Jensen, James Kermode, John R Kitchin, Esben Leonhard Kolsbjerg, Joseph Kubal, Kristen Kaasbjerg, Steen Lysgaard, Jón Bergmann Maronsson, Tristan Maxson, Thomas Olsen, Lars Pastewka, Andrew Peterson, Carsten Rostgaard, Jakob Schiøtz, Ole Schütt, Mikkel Strange, Kristian S Thygesen, Tejs Vegge, Lasse Vilhelmsen, Michael Walter, Zhenhua Zeng, and Karsten W Jacobsen. The atomic simulation environment—a Python library for working with atoms. *J. Phys.: Condens. Matter*, 29(27):273002, June 2017. ISSN 0953-8984. doi: 10.1088/1361-648X/aa680e. URL https://dx.doi.org/10.1088/1361-648X/aa680e.

Eric B. Isaacs and Chris A. Marianetti. Compositional phase stability of correlated electron materials within DFT+DMFT. *Phys. Rev. B*, 102:045146, Jul 2020. doi: 10.1103/PhysRevB.102.045146. URL https://link.aps.org/doi/10.1103/PhysRevB.102.045146.

Anubhav Jain, Geoffroy Hautier, Charles Moore, Shyue Ong, Chris Fischer, Tim Mueller, Kristin Persson, and Gerbrand Ceder. A high-throughput infrastructure for density functional theory calculations. *Comput. Mater. Sci.*, 50:2295–2310, June 2011. doi: 10.1016/j.commatsci.2011.02.023.

Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, and Kristin A. Persson. Commentary: The Materials Project: A materials genome approach to accelerating materials innovation. *APL Materials*, 1(1):011002, July 2013. doi: 10.1063/1.4812323. URL https://aip.scitation.org/doi/10.1063/1.4812323.

Rui Jiao, Wenbing Huang, Peijia Lin, Jiaqi Han, Pin Chen, Yutong Lu, and Yang Liu. Crystal Structure Prediction by Joint Equivariant Diffusion, July 2023. URL https://arxiv.org/abs/2309.04475v2.

Rui Jiao, Wenbing Huang, Yu Liu, Deli Zhao, and Yang Liu. Space Group Constrained Crystal Generation, April 2024. URL http://arxiv.org/abs/2402.03992.

Nikita Kazeev, Ruiming Zhu, Ignat Romanov, Andrey E. Ustyuzhanin, Shuya Yamazaki, Wei Nong, and Kedar Hippalgaonkar. WyckoffTransformer: Generation of Symmetric Crystals. In *AI for Accelerated Materials Design - NeurIPS 2024*, November 2024. URL https://openreview.net/forum?id=Jcy1bPOqrY.

Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E Gonzalez, and Ion Stoica. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling, February 2023. URL http://arxiv.org/abs/2210.02747.

Chang Liu, Feng Li, Lai-Peng Ma, and Hui-Ming Cheng. Advanced Materials for Energy Storage. *Advanced Materials*, 22(8):E28–E62, 2010. ISSN 1521-4095. doi: 10.1002/adma.200903328. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.200903328.

Wilhelm F. Maier. Early Years of High-Throughput Experimentation and Combinatorial Approaches in Catalysis and Materials Science. *ACS Comb. Sci.*, 21(6):437–444, June 2019. ISSN 2156-8952. doi: 10.1021/acscombsci.8b00189. URL https://doi.org/10.1021/acscombsci.8b00189.

MatterGen. Mattergen. Microsoft, January 2025. URL https://github.com/microsoft/mattergen.

Michael J. Mehl, David Hicks, Cormac Toher, Ohad Levy, Robert M. Hanson, Gus Hart, and Stefano Curtarolo. The AFLOW Library of Crystallographic Prototypes: Part 1. *Comput. Mater. Sci.*, 136: S1–S828, August 2017. ISSN 0927-0256. doi: 10.1016/j.commatsci.2017.01.017. URL https://www.sciencedirect.com/science/article/pii/S0927025617300241.

Amil Merchant, Simon Batzner, Samuel S. Schoenholz, Muratahan Aykol, Gowoon Cheon, and Ekin Dogus Cubuk. Scaling deep learning for materials discovery. *Nature*, 624:80–85, November 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06735-9. URL https://www.nature.com/articles/s41586-023-06735-9.

Benjamin Kurt Miller, Ricky T. Q. Chen, Anuroop Sriram, and Brandon M. Wood. FlowMM: Generating Materials with Riemannian Flow Matching, June 2024. URL http://arxiv.org/abs/2406.04713.

Yuki Nakaya and Shinya Furukawa. Catalysis of Alloys: Classification, Principles, and Design for a Variety of Materials and Reactions. *Chem. Rev.*, 123(9):5859–5947, May 2023. ISSN 0009-2665. doi: 10.1021/acs.chemrev.2c00356. URL https://doi.org/10.1021/acs.chemrev.2c00356.

Artem R. Oganov, Chris J. Pickard, Qiang Zhu, and Richard J. Needs. Structure prediction drives materials discovery. *Nat Rev Mater*, 4(5):331–348, May 2019. ISSN 2058-8437. doi: 10.1038/s41578-019-0101-8. URL https://www.nature.com/articles/s41578-019-0101-8.

Shyue Ping Ong, William Davidson Richards, Anubhav Jain, Geoffroy Hautier, Michael Kocher, Shreyas Cholia, Dan Gunter, Vincent L. Chevrier, Kristin A. Persson, and Gerbrand Ceder. Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis. *Computational Materials Science*, 68:314–319, February 2013. ISSN 0927-0256. doi: 10.1016/j.commatsci.2012.10.028. URL https://www.sciencedirect.com/science/article/pii/S0927025612006295.

Chris J. Pickard. Airss data for carbon at 10gpa and the c+n+h+o system at 1gpa, 2020. URL https://archive.materialscloud.org/record/2020.0026/v1.

Chris J Pickard and R J Needs. *Ab Initio* random structure searching. *J. Phys.: Condens. Matter*, 23(5): 053201, February 2011. ISSN 0953-8984, 1361-648X. doi: 10.1088/0953-8984/23/5/053201. URL https://iopscience.iop.org/article/10.1088/0953-8984/23/5/053201.

Radislav Potyrailo, Krishna Rajan, Klaus Stoewe, Ichiro Takeuchi, Bret Chisholm, and Hubert Lam. Combinatorial and High-Throughput Screening of Materials Libraries: Review of State of the Art. *ACS Comb. Sci.*, 13(6):579–633, November 2011. ISSN 2156-8952. doi: 10.1021/co200007w. URL https://doi.org/10.1021/co200007w.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding, May 2022.

Víctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9323–9332. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/satorras21a.html.

Jonathan Schmidt, Noah Hoffmann, Hai-Chen Wang, Pedro Borlido, Pedro J. M. A. Carriço, Tiago F. T. Cerqueira, Silvana Botti, and Miguel A. L. Marques. Large-scale machine-learning-assisted exploration of the whole materials space, October 2022a. URL http://arxiv.org/abs/2210.00579.

Jonathan Schmidt, Hai-Chen Wang, Tiago F. T. Cerqueira, Silvana Botti, and Miguel A. L. Marques. A dataset of 175k stable and metastable materials calculated with the PBEsol and SCAN functionals. *Scientific Data*, 9(1):64, March 2022b. ISSN 2052-4463. doi: 10.1038/s41597-022-01177-w.

G. Jeffrey Snyder and Eric S. Toberer. Complex thermoelectric materials. *Nature Mater*, 7(2): 105–114, February 2008. ISSN 1476-4660. doi: 10.1038/nmat2090. URL https://www.nature.com/articles/nmat2090.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep Unsupervised Learning using Nonequilibrium Thermodynamics, November 2015. URL http://arxiv.org/abs/1503.03585.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations, February 2021. URL http://arxiv.org/abs/2011.13456.

Anuroop Sriram, Benjamin Miller, Ricky TQ Chen, and Brandon Wood. Flowllm: Flow matching for material generation with large language models as base distributions. *Advances in Neural Information Processing Systems*, 37:46025–46046, 2024.

Dusan Strmcnik, Pietro Papa Lopes, Bostjan Genorio, Vojislav R. Stamenkovic, and Nenad M. Markovic. Design principles for hydrogen evolution reaction catalyst materials. *Nano Energy*, 29: 29–36, November 2016. ISSN 2211-2855. doi: 10.1016/j.nanoen.2016.04.017. URL https://www.sciencedirect.com/science/article/pii/S2211285516300738.

William W Tipton and Richard G Hennig. A grand canonical genetic algorithm for the prediction of multi-component phase diagrams and testing of empirical potentials. *Journal of Physics: Condensed Matter*, 25(49):495401, November 2013. ISSN 0953-8984. doi: 10.1088/0953-8984/25/49/495401. URL https://dx.doi.org/10.1088/0953-8984/25/49/495401.

Atsushi Togo, Kohei Shinohara, and Isao Tanaka. Spglib: A software library for crystal symmetry search, March 2024. URL http://arxiv.org/abs/1808.01590.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport, March 2024. URL http://arxiv.org/abs/2302.00482.

Joshua A. Vita, Eric G. Fuemmeler, Amit Gupta, Gregory P. Wolfe, Alexander Quanming Tao, Ryan S. Elliott, Stefano Martiniani, and Ellad B. Tadmor. ColabFit exchange: Open-access datasets for data-driven interatomic potentials. *J. Chem. Phys.*, 159(15):154802, October 2023. ISSN 0021-9606. doi: 10.1063/5.0163882. URL https://doi.org/10.1063/5.0163882.

Tian Xie, Xiang Fu, Octavian-Eugen Ganea, Regina Barzilay, and Tommi Jaakkola. Crystal Diffusion Variational Autoencoder for Periodic Material Generation, March 2022.

Han Yang, Chenxi Hu, Yichi Zhou, Xixian Liu, Yu Shi, Jielan Li, Guanzhi Li, Zekun Chen, Shuizhou Chen, Claudio Zeni, Matthew Horton, Robert Pinsler, Andrew Fowler, Daniel Zügner, Tian Xie, Jake Smith, Lixin Sun, Qian Wang, Lingyu Kong, Chang Liu, Hongxia Hao, and Ziheng Lu. MatterSim: A Deep Learning Atomistic Model Across Elements, Temperatures and Pressures, May 2024. URL http://arxiv.org/abs/2405.04967.

Jason Yim, Andrew Campbell, Andrew Y. K. Foong, Michael Gastegger, José Jiménez-Luna, Sarah Lewis, Victor Garcia Satorras, Bastiaan S. Veeling, Regina Barzilay, Tommi Jaakkola, and Frank Noé. Fast protein backbone generation with SE(3) flow matching, October 2023. URL http://arxiv.org/abs/2310.05297.

Nader Zaki, Hyowon Park, Richard M. Osgood, Andrew J. Millis, and Chris A. Marianetti. Failure of dft-based computations for a stepped-substrate-supported correlated co wire. *Phys. Rev. B*, 89: 205427, May 2014. doi: 10.1103/PhysRevB.89.205427. URL https://link.aps.org/doi/10.1103/PhysRevB.89.205427.

Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Zilong Wang, Aliaksandra Shysheya, Jonathan Crabbé, Shoko Ueda, Roberto Sordillo, Lixin Sun, Jake Smith, Bichlien Nguyen, Hannes Schulz, Sarah Lewis, Chin-Wei Huang, Ziheng Lu, Yichi Zhou, Han Yang, Hongxia Hao, Jielan Li, Chunlei Yang, Wenjie Li, Ryota Tomioka, and Tian Xie. A generative model for inorganic materials design. *Nature*, January 2025. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-025-08628-5.

Ruiming Zhu, Wei Nong, Shuya Yamazaki, and Kedar Hippalgaonkar. WyCryst: Wyckoff Inorganic Crystal Generator Framework, March 2024. URL http://arxiv.org/abs/2311.17916.

A IMPLEMENTATION DETAILS OF STOCHASTIC INTERPOLANTS

A.1 OMATG FRAMEWORK

Figures 4 and 5 summarize the training and the integration pipeline of the OMatG framework, respectively. Depending on the specific task, there are several stochastic interpolants at once. For CSP, one stochastic interpolant considers lattice vectors $\boldsymbol{L}$, and another one considers fractional coordinates $\boldsymbol{X}$. The model output of CSPNet (see Appendix B.1) depends on the full structural representation $\{\boldsymbol{A}, \boldsymbol{X}, \boldsymbol{L}\}$ and time t, where $\boldsymbol{A}$ are the atomic species. For the DNG task, we additionally use discrete flow matching for the atomic species $\boldsymbol{A}$ (Campbell et al., 2024).

During the numerical integration in the CSP task, $\boldsymbol{X}$ and $\boldsymbol{L}$ are integrated jointly while $\boldsymbol{A}$ is fixed. For DNG, $\boldsymbol{A}$ is evolved according to discrete flow matching (Campbell et al., 2024) (see Appendix A.5). For the SDE sampling scheme in Fig. 5, one chooses a time-dependent noise $\varepsilon(t)$ that only appears during integration and not during training. Also, $\gamma(t)$ has to be unequal zero in order to prevent the divergence in $1/\gamma(t)$. However, since $\gamma(t)$ necessarily vanishes at times $t = 0$ and $t = 1$ (see Appendix A.2), one should choose a time-varying $\varepsilon(t)$ that vanishes near these endpoints (Albergo et al., 2023).

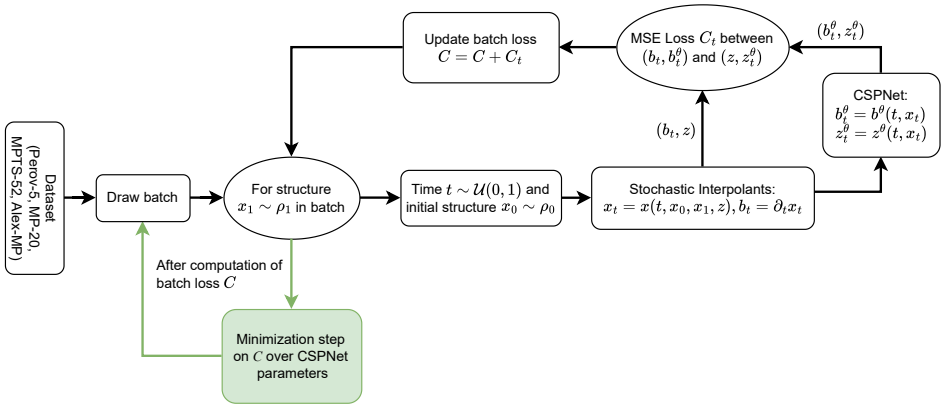

Figure 4: Training pipeline of the OMatG framework: A batch of structures is drawn from a dataset with target distribution ρ_1. Every structure $x_1 \sim \rho_1$ is connected with a structure x_0 from the base distribution ρ_0 with stochastic interpolants that yield the interpolated structure $x_t = x(t, x_0, x_1, z)$ and the drift $b_t = \partial_t x_t$ at time $t \sim \mathcal{U}(0, 1)$, possibly using a random variable $z \sim \mathcal{N}(0, \boldsymbol{I})$. The model CSPNet predicts $b_t^\theta = b^\theta(t, x_t)$ and $z_t^\theta = z(t, x_t)$ and its parameters are minimized based on the MSE losses in Eqs 2 and 3.

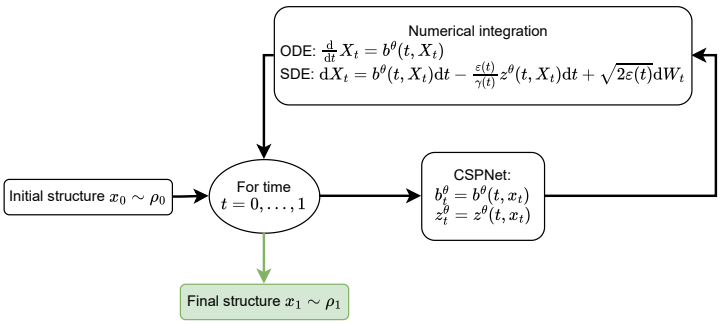

Figure 5: Numerical integration pipeline of the OMatG framework: An initial structure x_0 from the base distribution ρ_0 is numerically integrated following either an ODE or an SDE based on the model predictions b_t^θ and z_t^θ. For an SDE, one can choose a noise $\varepsilon(t)$ during integration.

Table 4: SI parameters from Albergo et al. (2023).

	Stochastic Interpolant	$\alpha(t)$	$\beta(t)$	$\gamma(t)$
Arbitrary ρ_0	linear	$1-t$	t	$\sqrt{at(1-t)}$
	trig	$\cos\left(\frac{\pi}{2}t\right)$	$\sin\left(\frac{\pi}{2}t\right)$	$\sqrt{at(1-t)}$
	enc-dec	$\cos^2(\pi t)1_{[0,\frac{1}{2})}(t)$	$\cos^2(\pi t)1_{(\frac{1}{2},1]}(t)$	$\sin^2(\pi t)$
Gaussian ρ_0	SBD	$\sqrt{1-t^2}$	t	0

A.2 Interpolant choice

In this work, we are concerned with interpolants with spatially linear interpolants of the form specified in Eq. (1). Here, the following conditions must be met (Albergo et al., 2023):

$$\alpha(0) = \beta(1) = 1, \quad \alpha(1) = \beta(0) = \gamma(0) = \gamma(1) = 0, \quad \gamma(t) > 0 \,\forall t \in (0,1). \tag{5}$$

Under these constraints, the form of the SI is relatively flexible, and many different interpolants can be defined. Also, the base distribution can be arbitrary as in CFM (Tong et al., 2024). In this work, we mostly rely on interpolants originally defined in Albergo et al. (2023) and listed in Tab. 4. For the SBD interpolant, we allow for different variances σ of the Gaussian base distribution ρ_0. The encoder-decoder (enc-dec) interpolant as defined in Tab. 4 evolves samples from the base distribution ρ_0 to follow an intermediate Gaussian distribution with variance 1 at the switch time $T_{\text{switch}} = 0.5$, before mapping them to a sample from ρ_1. This can be generalized to arbitrary variances $a > 0$ and switch times $T_{\text{switch}} \in (0,1)$:

$$\begin{aligned}
\alpha(t) &= \cos^2\left(\frac{\pi(t - T_{\text{switch}}t)^p}{(T_{\text{switch}} - T_{\text{switch}}t)^p + (t - T_{\text{switch}}t)^p}\right) 1_{[0,T_{\text{switch}})}(t), \\
\beta(t) &= \cos^2\left(\frac{\pi(t - T_{\text{switch}}t)^p}{(T_{\text{switch}} - T_{\text{switch}}t)^p + (t - T_{\text{switch}}t)^p}\right) 1_{(T_{\text{switch}},1]}(t), \\
\gamma(t) &= \sqrt{a}\sin^2\left(\frac{\pi(t - T_{\text{switch}}t)^p}{(T_{\text{switch}} - T_{\text{switch}}t)^p + (t - T_{\text{switch}}t)^p}\right),
\end{aligned} \tag{6}$$

where $p \geq 1/2$. We consider the cases $p \in \{1/2, 1\}$ and note that the general interpolant in Eq. (6) reduces to the interpolant in Tab. 4 for $a = 1$, $p = 1$, and $T_{\text{switch}} = 0.5$.

A.3 Antithetic sampling

As shown by Albergo et al. (2023), the loss function can become unstable around $t = 0$ and $t = 1$ for certain choices of $\gamma(t)$. To account for this, we implement antithetic sampling. This requires to compute the loss at both x^+ and x^- where:

$$x^+(t, x_0, x_1, z) = \alpha(t)x_0 + \beta(t)x_1 + \gamma(t)z, \tag{7}$$

$$x^-(t, x_0, x_1, z) = \alpha(t)x_0 + \beta(t)x_1 - \gamma(t)z. \tag{8}$$

A.4 Interpolation with periodic boundary conditions

We show in Fig. 6 how we implement periodic versions of interpolants in order to represent fractional coordinates in a unit cell with periodic boundary conditions. We emphasize that this procedure is important not only for the choice of interpolant, but also for the addition of the latent variable $\gamma(t)z$ which also moves the interpolation trajectory away from the geodesic.

A.5 DFM Details

DFM allows for generative modeling of discrete sequences of tokens while respecting the discrete nature of the design space. As discussed, a parameterized neural network $p^\theta_{1|t}(x_1|x_t)$ is learned, which

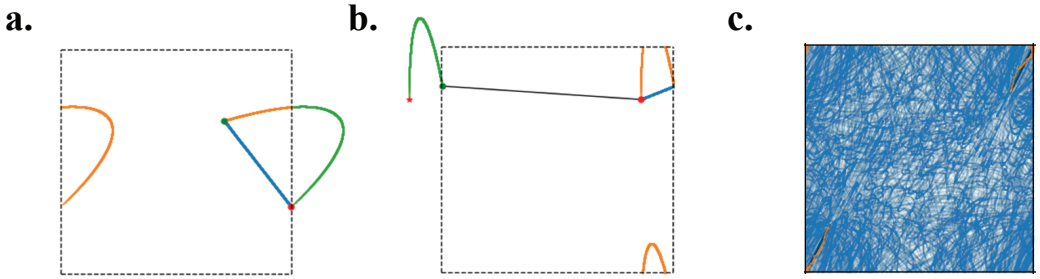

Figure 6: Extending interpolants to incorporate periodic boundary conditions. **(a–b)** The path for a score based diffusion interpolant is calculated by first computing the shortest-path geodesic (blue) between the initial (green dot) and final positions (red dot). Next, the path of the interpolant moving the final position outside the bounding box is computed (green), and finally the path is wrapped back into the bounding box to produce the interpolant trajectory (orange). **(c)** The effect of adding a latent variable to any interpolant must be handled similarly to calculating the path of a non-linear interpolant. For a linear interpolant with a nonzero γ, we show samples of possible paths (blue) and their averaged path (orange) which collapses onto the path of the linear interpolant.

attempts to predict the final sequence from the current sequence. Borrowing from Campbell et al. (2024), we choose a conditional rate matrix $R_t(x_t, i|x_1)$ generating the conditional flow $p_{t|1}(x_t|x_1)$ of the form:

$$R_t(x_t, i|x_1) = \frac{\text{ReLU}\left(\partial_t p_{t|1}(i|x_1) - \partial_t p_{t|1}(x_t|x_1)\right)}{S \cdot p_{t|1}(x_t|x_1)}, \tag{9}$$

where S is the number of possible tokens a sequence element can take on. This conditional rate matrix can be modified by including a term that introduces stochasticity in the form of a detailed balance rate matrix R_t^{DB} by writing $R_t^{\eta} = R_t + \eta R_t^{DB}$. Here, (Campbell et al., 2024):

$$R_t^{DB}(i, j|x_1) = \eta\delta\{i, x_1\}\delta\{j, M\} + \frac{\eta t}{1-t}\delta\{i, M\}\delta\{j, x_1\}, \tag{10}$$

where M is the masking token. The parameter $\eta \in \mathbb{R}^+$ represents the level of stochasticity that only appears during generation.

During generation, our objective is to compute $R_t^{\theta}(x_t, i)$ based on the learned distribution $p_{1|t}^{\theta}(x_1|x_t)$. Formally, we have

$$R_t^{\theta}(x_t, i) = \mathbb{E}_{p_{1|t}^{\theta}(x_1|x_t)}\left[R_t^{\eta}(x_t, i|x_1)\right] \tag{11}$$

In practice, Campbell et al. (2024) show that we need not compute a full expectation, but rather, simply draw $x_1 \sim p_{1|t}^{\theta}(x_1|x_t)$, evaluate the conditional rate matrix, and perform an update of x_t to $x_{t+\Delta t}$ with discrete time step Δt directly from this by sampling $x_{t+\Delta t}$ according to

$$p_{t+\Delta t|t}(x_{t+\Delta t}|x_1, x_t) = \delta\{x_t, x_{t+\Delta t}\} + R_t^{\eta}(x_t, i|x_1)\Delta t. \tag{12}$$

A.6 Velocity annealing

Velocity annealing—rescaling the learned velocity field during generation to increase velocity over time as $b^{\theta}(t, x) \to (1 + st)\, b^{\theta}(t, x)$ with s as an hyperparameter during integration—has been empirically shown to improve performance in a number of studies that apply CFM to physical systems (Yim et al., 2023; Bose et al., 2024; Miller et al., 2024). For instance, Miller et al. (2024) demonstrated that applying velocity annealing significantly improves performance in CSP and DNG benchmarks for materials. Motivated by these findings, we include velocity annealing in OMatG as a tunable hyperparameter, while emphasizing that this technique lacks a formal theoretical justification within the mathematical frameworks underlying flow models and stochastic interpolants.

A.7 Data-dependent coupling

SIs have been used with data-dependent couplings (Albergo et al., 2024), where a coupling function $\nu(x_0, x_1)$ enables biasing of x_0 based on the sampled x_1. In OMatG, we incorporate an optional

data-dependent coupling that enforces an ordering (*i.e.*, a permutation on the order of atomic elements within a structure) that produces the minimum fractional-coordinate distance between each particle pair $(\boldsymbol{x}_0^i, \boldsymbol{x}_1^i)$ from structures $x_0 \in \rho_0$ and $x_1 \in \rho_1$. We find that the inclusion of this data-dependent coupling is optimal during hyperparameter tuning depending on the type of model: CSP models typically performed better without this coupling, but DNG models (see Tab. 10) can benefit in certain cases from minimizing traveled distance *via* permutation of elements.

Formally, our coupling is conditional on the sampled (x_0, x_1) and is defined as

$$\arg\min_p \sum_i d(p(x_0^i), x_1^i). \tag{13}$$

Here, $d(\cdot, \cdot)$ is a distance metric which we define on a periodic manifold in fractional-coordinate space (*i.e.*, a four-dimensional torus) and p is some permutation function that permutes the discrete indices i. Under this coupling, we still sample (x_0, x_1) independently but then bias the sampled x_0 to travel the minimum permutational distance necessary to reach the target structure.

B MODEL ARCHITECTURE

B.1 GRAPH NEURAL NETWORK

We implement a message-passing GNN with CSPNet as introduced in Jiao et al. (2023):

$$\boldsymbol{h}_{(0)}^{i} = \phi_{\boldsymbol{h}_{(0)}}(\boldsymbol{a}^{i}) \tag{14}$$

$$\boldsymbol{m}_{(s)}^{ij} = \varphi_{m}\left(\boldsymbol{h}_{s-1}^{i}, \boldsymbol{h}_{s-1}^{j}, \boldsymbol{l}, \text{SinusoidalEmbedding}(\boldsymbol{x}^{j} - \boldsymbol{x}^{i})\right) \tag{15}$$

$$\boldsymbol{m}_{(s)}^{i} = \sum_{j=1}^{N} \boldsymbol{m}_{(s)}^{ij} \tag{16}$$

$$\boldsymbol{h}_{(s)}^{i} = \boldsymbol{h}_{(s-1)}^{i} + \varphi_{h}(\boldsymbol{h}_{(s-1)}, \boldsymbol{m}_{(s)}^{i}) \tag{17}$$

$$b_{\boldsymbol{x}} = \varphi_{\boldsymbol{x}}\left(\boldsymbol{h}_{(\max s)}^{i}\right) \tag{18}$$

$$b_{\boldsymbol{l}} = \varphi_{\boldsymbol{l}}\left(\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{h}_{(\max s)}^{i}\right) \tag{19}$$

Here, node embeddings, $\boldsymbol{h}$, are initialized as a function of the atom types, $\boldsymbol{a}$. Embeddings are then updated by a message passing scheme through a series of graph convolution layers. Messages are computed with a parameterized neural network, φ_m, from neighboring node embeddings as well as information about the lattice, $\boldsymbol{l}$, and distance between the fractional coordinates $\boldsymbol{x}$. All necessary drift and denoiser terms are computed from single layer MLPs applied to the final node embeddings.

For the CFP model that should only predict compositions, we simply remove the input of the lattice $\boldsymbol{l}$ and the fractional coordinates $\boldsymbol{x}$ from the computation of the message in Eq. (15). This ensures that the output $p_{1|t}^{\theta}(\boldsymbol{a}_1|x_t)$ of CSPNet for the composition does not depend on lattice vectors or fractional coordinates, while preserving permutational equivariance.

B.2 LOSS FUNCTION

With Eqs (2), (3), and (4), we can construct a loss function for the modeling of our joint distribution of interest,

$$\begin{aligned}\mathcal{L}(\theta) =& \mathbb{E}_{t,z,x_0,x_1}\big[\\ & \lambda_{\boldsymbol{x},b}\left[|b_{\boldsymbol{x}}^{\theta}(t,x_t)|^2 - 2\partial_t x(t,x_0,x_1,z)\cdot b_{\boldsymbol{x}}^{\theta}(t,x_t)\right] + \lambda_{\boldsymbol{x},z}\left[|z_{\boldsymbol{x}}^{\theta}(t,x_t)|^2 - 2 z_{\boldsymbol{x}}^{\theta}(t,x_t)\cdot z\right] \\ & + \lambda_{\boldsymbol{l},b}\left[|b_{\boldsymbol{l}}^{\theta}(t,x_t)|^2 - 2\partial_t x(t,x_0,x_1,z)\cdot b_{\boldsymbol{l}}^{\theta}(t,x_t)\right] + \lambda_{\boldsymbol{l},z}\left[|z_{\boldsymbol{l}}^{\theta}(t,x_t)|^2 - 2 z_{\boldsymbol{l}}^{\theta}(t,x_t)\cdot z\right] \\ & + \lambda_{\boldsymbol{a}}\left[\log p_{1|t}^{\theta}(\boldsymbol{a}_1|x_t)\right]\big].\end{aligned} \tag{20}$$

where the λ terms correspond to the relative weights of each term in the loss function. These weighting factors are hyperparameters that are included in our hyperparameter sweep. The respective terms for the fractional coordinates and lattice vectors corresponding to Eqs (2) and (3) are equivalent to a mean-squared error (MSE) loss function. They only differ by a constant term that does not influence gradients. We do not include that constant term because the possible divergence of $\partial_t \gamma(t)$ near $t = 0$ and $t = 1$ can artifically inflate the absolute value of the loss, even when antithetic sampling is applied (see Section A.3).

B.3 HYPERPARAMETER OPTIMIZATION

For every choice of the positional interpolant, an independent hyperparameter optimization was performed using the `Ray Tune` package (Liaw et al., 2018) in conjunction with the HyperOpt Python library (Bergstra et al., 2013) for Bayesian optimization. The tuned hyperparameters include both those relevant during training—the relative losses λ, the choice of stochastic interpolant for the lattice vectors, the parameters for chosen $\gamma(t)$ (if necessary), the sampling scheme, the usage of data-dependent coupling, the batch size, and the learning rate—and during inference—the number of

integration steps, the choice of the noises $\varepsilon(t)$ and η, and the magnitude of the velocity annealing parameter s for both lattice vectors and atomic coordinates. Additionally, OMatG provides the option to use the Niggli-reduced primitive cell (Grosse-Kunstleve et al., 2004; Hjorth Larsen et al., 2017) for training.

We provide hyperparameter-tuned models with the relevant performance metrics and hyperparameters for *perov-5* CSP in Tabs 5 and 6, *MP-20* CSP in Tabs 7 and 8, and *MP-20* DNG in Tabs 9 and 10.

B.4 Discussion on performance and optimal hyperparameters

By adjusting the relative cost between the position, cell, and species terms in the loss, as well as optionally including data-dependent coupling (minimum distance permutation), Niggli-reduction, and velocity annealing, we observe that the main learning challenge lies in the accurate prediction of the atomic coordinates, which tended to have a higher relative weight in calculating the full loss function in Eq. (20) (see Tabs 6 and 8). However, the two best-performing models for the *perov-5* dataset instead exhibited the opposite—lending the most weight to learning of the cell vectors.

For the CSP task on the *perov-5* dataset, we highlighted the particularly strong performance of the SBD and trigonometric interpolants in achieving a high match rate, as shown in Tab. 1. Unlike other datasets, *perov-5* has a fixed number of atoms $N = 5$ per unit cell and a fixed (cubic) cell with varying side lengths and similar fractional positions—a combination which should not expose the model to a large variety of unit cell choices during interpolation or generation. By contrast, in other datasets, *no unique representation of the periodic repeat unit is imposed on flows*, meaning the model cannot learn the invariance (or even equivariance) to the choice of periodic repeat unit.[6] This likely contributes to the difficulty of unconstrained flow-based models in generating highly symmetric structures. Thus, the *perov-5* dataset presents a unique case where the invariance to unit cell choice does not need to be learned, making this dataset a useful benchmark for evaluating positional interpolant performance. It is possible that the superior performance of the SBD or trigonometric interpolants arise from their more circuitous interpolation, and therefore generation trajectories, by comparison to the strictly geodesic paths imposed by the linear interpolant—akin to the reasoning behind using latent variables to enhance learning in SIs (Albergo et al., 2023).

For the CSP task on the *MP-20* dataset, we noted several models in Tab. 7 which surpassed the previous state-of-the-art for match rate: the linear positional interpolant, with both an ODE and SDE sampling scheme, as well as the trigonometric positional interpolant with an SDE sampling scheme (which also requires a nonzero latent variable $\gamma(t)z$).

Finally, we note trends among the best-performing DNG models identified through hyperparameter optimization. We show in Tabs 9 and 10 the performance metrics and hyperparameters for each model by choice of positional interpolant, sampling scheme, and $\gamma(t)$ in the latent variable. In particular, we observe several of our best performing models (with respect to S.U.N. and RMSD) possess lower levels of 'species noise' η which sets the probability that an atom will change its identity if already in an unmasked state (see Appendix A.5). Additionally, we find a correlation between enabling element order permutation (to minimimize the distance between x_0 and x_1) and the use of linear or trigonometric interpolation paths, with both favoring the use of minimum distance permutation in contrast to the encoder-decoder and SBD postional interpolants. With respect to species noise, we noted that all models with the trigonometric positional interpolant had a high weighting relative to other positional interpolants. Finally, we note that SBD models require a similar magnitude of annealing for positions and lattices. This is in stark contrast to all other models, where a significantly larger annealing parameter s is required for generating positions.

[6] Using Niggli reduction during learning to enforce a unique choice of unit cell on structures from our datasets is not sufficient for enforcing this invariance during generation of structures.

Table 5: Study for the *perov-5* dataset comparing CSP performance metrics for choice of positional interpolant, sample scheme, and $\gamma(t)$ in the latent variable (or Gaussian width σ for SBD interpolants).

Positional interpolant	Positional sampling scheme	Positional $\gamma(t)$	Match rate (%, Full / Valid)	RMSE (Full / Valid)
Linear	ODE	0	51.86% / 50.62%	0.0757 / 0.0760
Linear	ODE	$\sqrt{0.034t(1-t)}$	72.21% / 62.54%	0.3510 / 0.3444
Linear	SDE	$\sqrt{0.028t(1-t)}$	74.16% / 72.87%	0.3307 / 0.3315
Trigonometric	ODE	0	81.51% / 52.36%	0.3674 / 0.3628
Trigonometric	ODE	$\sqrt{0.011t(1-t)}$	80.85% / 79.55%	0.3864 / 0.3873
Trigonometric	SDE	$\sqrt{0.063t(1-t)}$	73.37% / 71.60 %	0.3610 / 0.3614
Encoder-Decoder	ODE	$\sqrt{0.66}\sin^2\left(\frac{\pi(t-0.80t)}{(0.80-0.80t)+(t-0.80t)}\right)$	68.08% / 64.60%	0.4005 / 0.4003
Encoder-Decoder	SDE	$\sqrt{8.45}\sin^2\left(\frac{\pi(t-0.61t)}{(0.61-0.61t)+(t-0.61t)}\right)$	78.28% / 76.80%	0.3616 / 0.3620
Score-Based Diffusion	ODE	- ($\sigma = 0.28$)	83.06% / 81.27%	0.3753 / 0.3755
Score-Based Diffusion	SDE	- ($\sigma = 0.13$)	76.54% / 64.46%	0.3529 / 0.3402

Table 6: Study for the *perov-5* dataset CSP comparing hyperparameters for each choice of positional interpolant, sample scheme, and $\gamma(t)$ (as reported in Tab. 5).

Pos. interpolant, Sampling scheme, γ	Cell interpolant Sampling scheme, $\gamma(t)$	Annealing param. s (Pos. / Cell)	Integration steps	Min. dist. permutation	Niggli	$\lambda_{\boldsymbol{x},b}/\lambda_{\boldsymbol{l},b}/\lambda_{\boldsymbol{x},z}/\lambda_{\boldsymbol{l},z}$
Linear, ODE, None	Linear, ODE, $\gamma = 0$	14.11 / 2.90	820	False	False	0.9729 / 0.0271 / - / -
Linear, ODE, LatentSqrt	Linear, ODE, $\gamma = 0$	0.008 / 12.19	820	True	True	0.9724 / 0.0276 / - / -
Linear, SDE, LatentSqrt	Linear, ODE, $\gamma = \sqrt{0.013t(1-t)}$	8.20 / 1.46	910	True	True	0.0024 / 0.0051 / 0.9925 / -
Trig, ODE, None	Linear, ODE, $\gamma = \sqrt{0.021t(1-t)}$	14.99 / 14.97	880	True	False	0.9983 / 0.0017 / - / -
Trig, ODE, LatentSqrt	Linear, ODE, $\gamma = 0$	9.68 / 2.42	110	False	False	0.1130 / 0.8870 / - / -
Trig, SDE, LatentSqrt	Linear, ODE, $\gamma = \sqrt{0.051t(1-t)}$	3.43 / 0.03	900	True	True	0.6868 / 0.0643 / 0.2489 / -
Enc-Dec, ODE, Enc-Dec	Linear, ODE, $\gamma = 0$	14.94 / 0.318	460	True	True	0.8563 / 0.1437 / - / -
Enc-Dec, SDE, Enc-Dec	Linear, ODE, $\gamma = \sqrt{0.154t(1-t)}$	14.55 / 0.075	930	True	False	0.2828 / 0.0004 / 0.7168 / -
SBD, ODE	SBD, SDE, $\sigma = 0.61$	12.79 / 2.69	130	True	True	0.0035 / 0.0121 / - / 0.9844
SBD, SDE	Trig, SDE, $\gamma = \sqrt{0.029t(1-t)}$	11.54 / 11.53	350	True	False	0.2898 / 0.1960 / 0.3259 / 0.1883

Table 7: Study for the *MP-20* dataset comparing CSP performance metrics for choice of positional interpolant, sample scheme, and $\gamma(t)$ in the latent variable (or Gaussian width σ for SBD interpolants).

Positional interpolant	Positional sampling scheme	Positional $\gamma(t)$	Match rate (%, Full / Valid)	RMSE (Full / Valid)
Linear	ODE	0	69.83% / 63.75%	0.0741 / 0.0720
Linear	ODE	$\sqrt{0.257t(1-t)}$	55.60% / 50.04%	0.1531 / 0.1494
Linear	SDE	$\sqrt{0.063t(1-t)}$	68.20% / 61.88%	0.1632 / 0.1611
Trigonometric	ODE	0	65.30% / 58.94%	0.1184 / 0.1149
Trigonometric	ODE	$\sqrt{0.033t(1-t)}$	65.62% / 59.15%	0.1038 / 0.0998
Trigonometric	SDE	$\sqrt{0.049t(1-t)}$	67.47% / 61.39%	0.1340 / 0.1321
Encoder-Decoder	ODE	$\sqrt{1.99}\sin^2\left(\frac{\pi(t-0.65t)}{(0.65-0.65t)+(t-0.65t)}\right)$	55.15% / 49.45%	0.1306 / 0.1260
Encoder-Decoder	SDE	$\sqrt{0.04}\sin^2\left(\frac{\pi(t-0.42t)^{0.5}}{(0.42-0.42t)^{0.5}+(t-0.42t)^{0.5}}\right)$	57.69% / 52.44%	0.1160 / 0.1125
Score-Based Diffusion	ODE	- ($\sigma = 0.22$)	43.21% / 37.39%	0.2005 / 0.1890
Score-Based Diffusion	SDE	- ($\sigma = 2.29$)	42.29% / 38.08%	0.2124 / 0.2088

Table 8: Study for the *MP-20* dataset comparing CSP performance metrics for choice of positional interpolant, sample scheme, and $\gamma(t)$ (as reported in Tab. 7).

Pos. interpolant, Sampling scheme, γ	Cell interpolant Sampling scheme, $\gamma(t)$	Annealing param. s (Pos. / Cell)	Integration steps	Min. dist. permutation	Niggli	$\lambda_{\boldsymbol{x},b}/\lambda_{\boldsymbol{l},b}/\lambda_{\boldsymbol{x},z}/\lambda_{\boldsymbol{l},z}$
Linear, ODE, None	Linear, ODE, $\gamma = 0$	10.18 / 1.82	210	False	False	0.9994 / 0.0006 / - / -
Linear, ODE, LatentSqrt	Trig, ODE, $\gamma = \sqrt{2.976t(1-t)}$	7.76 / 4.12	690	False	True	0.9976 / 0.0024/ - / -
Linear, SDE, LatentSqrt	Linear, SDE, $\gamma = \sqrt{0.132t(1-t)}$	11.58 / 5.08	310	False	False	0.0073 / 0.0642 / 0.9154 / 0.0131
Trig, ODE, None	Enc-Dec, SDE, $\gamma = \sqrt{5.27}\sin^2\left(\frac{\pi(t-0.41t)^{0.5}}{(0.41-0.41t)^{0.5}+(t-0.41t)^{0.5}}\right)$	12.34 / 3.61	170	False	False	0.9967 / 0.0023 / - / 0.0010
Trig, ODE, LatentSqrt	Linear, SDE, $\gamma = \sqrt{0.017t(1-t)}$	13.54 / 2.38	780	False	True	0.9830 / 0.0167 / - / 0.0003
Trig, SDE, LatentSqrt	Trig, ODE, $\gamma = 0$	11.48 / 0.43	740	True	True	0.2468 / 0.0301 / 0.7231 / -
Enc-Dec, ODE, Enc-Dec	Trig, SDE, $\gamma = \sqrt{0.219t(1-t)}$	12.29 / 4.30	820	False	True	0.6892 / 0.1235 / - / 0.1873
Enc-Dec, SDE, Enc-Dec	Linear, ODE, $\gamma = \sqrt{4.961t(1-t)}$	3.78 / 1.14	710	False	True	0.6143 / 0.0063 / 0.3794 / -
SBD, ODE	Linear, ODE, $\gamma = 0$	6.61 / 2.45	890	True	True	0.9598 / 0.0402 / - / -
SBD, SDE	Linear, ODE, $\gamma = \sqrt{3.684t(1-t)}$	6.46 / 0.67	600	True	True	0.6060 / 0.0112 / 0.3828 / -

Table 9: Study for the *MP-20* dataset comparing DNG performance metrics for choice of positional interpolant, sample scheme, and $\gamma(t)$ in the latent variable (or Gaussian width σ for SBD interpolants).

Positional interpolant	Positional sampling scheme	Positional $\gamma(t)$	S.U.N. Rate	RMSD
Linear	ODE	0	18.59%	0.2939
Linear	ODE	$\sqrt{1.450t(1-t)}$	9.95%	1.6660
Linear	SDE	$\sqrt{0.018t(1-t)}$	22.07%	0.6148
Trigonometric	ODE	0	19.63%	0.8289
Trigonometric	ODE	$\sqrt{0.027t(1-t)}$	19.96%	0.6570
Trigonometric	SDE	$\sqrt{0.023t(1-t)}$	17.60%	0.7763
Encoder-Decoder	ODE	$\sin^2(\pi t)$	17.59%	0.3899
Encoder-Decoder	SDE	$\sqrt{0.10}\sin^2\left(\frac{\pi(t-0.73t)^{0.5}}{(0.73-0.73t)^{0.5}+(t-0.73t)^{0.5}}\right)$	16.27%	1.1795
Score-Based Diffusion	ODE	- ($\sigma = 0.23$)	17.30%	1.1376
Score-Based Diffusion	SDE	- ($\sigma = 7.14$)	22.10%	0.7631

Table 10: Study for the *MP-20* dataset comparing DNG performance metrics for choice of positional interpolant, sample scheme, and $\gamma(t)$ (as reported in Tab. 9).

Pos. interpolant, Sampling scheme, γ	Cell interpolant Sampling scheme, $\gamma(t)$	Annealing param. s (Pos. / Cell)	Integration steps	Min. dist. permutation	Niggli	Species noise (η)	$\lambda_{\boldsymbol{x},b}/\lambda_{\boldsymbol{l},b}/\lambda_{\boldsymbol{x},z}/\lambda_{\boldsymbol{l},z}/\lambda_{\boldsymbol{a}}$
Linear, ODE, None	Linear, ODE, $\gamma = 0$	13.62 / 1.07	150	True	False	7.08	0.9775/0.0006/-/-/0.0218
Linear, ODE, LatentSqrt	Enc-Dec, SDE, $\gamma = \sqrt{7.88}\sin^2\left(\frac{\pi(t-0.14t)}{(0.14-0.14t)+(t-0.14t)}\right)$	14.83 / 5.91	130	True	False	23.87	0.7683/0.0089/-/0.0012/0.2216
Linear, SDE, LatentSqrt	Linear, ODE, $\gamma = 0$	6.33 / 1.07	710	True	False	0.19	0.1309/0.0065/0.2708/-/0.5918
Trig, ODE, None	Trig, ODE, $\gamma = \sqrt{1.183t(1-t)}$	8.59 / 0.29	860	True	False	32.69	0.3302/0.0023/-/-/0.6675
Trig, ODE, LatentSqrt	Linear, SDE, $\gamma = \sqrt{0.848t(1-t)}$	7.79 / 0.30	680	True	True	27.25	0.2322/0.0035/-/0.3338/0.4306
Trig, SDE, LatentSqrt	Trig, ODE, $\gamma = \sqrt{0.316t(1-t)}$	12.80 / 4.36	760	True	False	13.15	0.6304/0.1582/0.0753/-/0.1360
Enc-Dec, ODE, Enc-Dec	Linear, ODE, $\gamma = 0$	10.27 / 0.08	840	False	False	0.85	0.7268/0.0084/-/-/0.2648
Enc-Dec, SDE, Enc-Dec	Linear, ODE, $\gamma = \sqrt{1.651t(1-t)}$	7.87 / 3.92	610	False	False	19.78	0.2143/0.1547/0.1968/-/0.4341
SBD, ODE	Trig, ODE, $\gamma = \sqrt{7.797t(1-t)}$	2.30 / 2.74	710	False	False	20.27	0.4053/0.0447/-/-/0.5500
SBD, SDE	Trig, SDE, $\gamma = \sqrt{3.100t(1-t)}$	9.06 / 11.77	870	False	False	8.52	0.5184/0.0044/0.0008/0.1180/0.3584

C METRICS FOR EVALUATING LEARNED MODELS

We consider a variety of different benchmarks to evaluate the performance of OMatG and its competitors. For the CSP task (see Tab. 1), we generate a structure for every composition in the test dataset. We then attempt to match every generated structure with the corresponding test structure using Pymatgen's `StructureMatcher` module (Ong et al., 2013) with tolerances (`stol` $= 0.5$, `ltol` $= 0.3$, `angletol` $= 10$). We finally report the match rate and the average root-mean square displacement (RMSE) between the test structures and *matched* generated structures. Here, the RMSEs computed by Pymatgen are normalized by $(V/N)^{1/3}$, where V is the (matched) volume and N is the number of atoms.

For the DNG task (see Tab. 2), metrics include validity (structural and compositional), coverage (recall and precision), and Wasserstein distances between distributions of properties including density ρ, number of unique elements N (*i.e.*, an Nary material), and average coordination number by structure $\langle CN \rangle$. We introduce the average coordination number benchmark in part due to the difficulty of generating symmetric structures; a structure's average coordination number is a useful fingerprint, and higher-coordinated structures tend to be more symmetric. In addition, the stability rate, the S.U.N. (stable, unique, and novel) rate, the root mean squared displacements (RMSD) between the generated and relaxed structures (unnormalized, in units of Å), and the average energy above the convex hull are computed.

C.1 MATCH RATE AND RMSE

Previously reported match rates filtered the matched generated structures by their structural and compositional validity. We note, however, that the datasets themselves contain invalid structures—for example, the *MP-20* test dataset has $\sim 10\%$ compositionally invalid structures. Thus, we argue that the removal of these invalid structures for computation of match rate and RMSE is not reasonable for assessing learning performance; we do, however, provide both match rates (with and without validation filtering).

We note that the match rate (computed with each generated-reference structure pair) does *not* account for cases where multiple stable polymorphs of the same composition exist in the reference dataset.

We include here a discussion on the tradeoff between match rate and RMSE, as we noticed this most strongly influences the *perov-5* dataset. We show in Fig. 7 how different positional interpolants for the atomic coordinates (trigonometric vs. linear with ODE sampling schemes) learn to generate matched structures differently. For the linear case the change in matching tolerance (`ltol`) makes little difference; for the trigonometric interpolant it makes a far more significant difference and leads to a much higher match rate, suggesting that the trigonometric interpolant learns structures more reliably but less accurately. However, match rate remains the primary metric of interest, as the generated structure can always be relaxed after generation if seeking minimum energy structures (i.e., at zero temperature).

C.2 VALIDITY METRICS

The structural validity of generated structures is defined according to the bond lengths present in the structure—all lengths must be $>$0.5 Å to be considered valid. The compositional validity is defined according to the SMACT software package (Davies et al., 2019). We note that the default oxidation states have been updated with the release of SMACT version 3.0 which changed the DNG compositional validity rates by several percent. This also impacts the CSP match rate when filtered by valid structures. As such, all values for all models were recomputed with the most up-to-date version (3.0) of the SMACT software.

C.3 CALCULATION OF S.U.N. RATES

Evaluation of DNG structures was performed using scripts provided by the developers of MatterGen (MatterGen, 2025). A total of 10,000 structures were generated from each of OMatG, DiffCSP, and FlowMM. These structures were then filtered to remove any that contained elements not supported by the MatterSim potential (version `MatterSim-v1.0.0-1M`) or the reference convex hull. These included heavy elements with atomic numbers $>$89, radioactive elements, and the noble gases

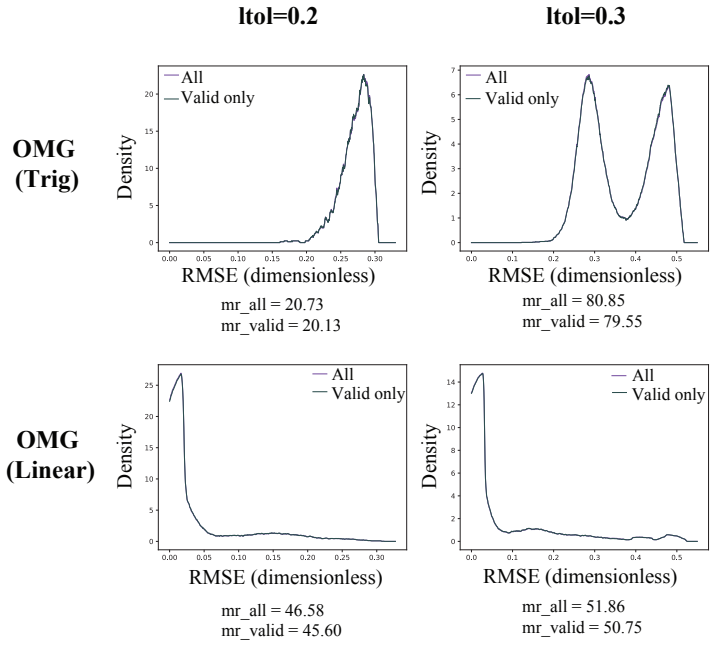

Figure 7: We show here the effect of making matching more difficult by decreasing the length tolerance used by pymatgen's `StructureMatcher`. We plot the density of the normalized RMSE distributions from CSP models trained on the *perov-5* dataset. We note that the curves for all generated structures and only valid generated structures overlap significantly.

(specifically: 'Ac', 'U', 'Th', 'Ne', 'Tc', 'Kr', 'Pu', 'Np', 'Xe', 'Pm', 'He', 'Pa'). Stability and novelty were computed with respect to the default dataset provided by MatterGen which contains 845 997 structures from the *MP-20* (Jain et al., 2013; Xie et al., 2022) and *Alexandria* (Schmidt et al., 2022b;a) datasets. This provides a more challenging reference for computing novelty as each model was trained only on the $\sim$ 27 000 structures from the *MP-20* training set.

MatterSim serves as a replacement for a full DFT relaxation, requiring far less compute resources. In order to confirm that reasonable conclusions can be made based on MatterSim relaxations, we compare the S.U.N. rates for the same structures based on MatterSim and DFT relaxations, respectively, in Tab. 11. Overall, there is a strong qualitative agreement between the two, with DFT based S.U.N. rates slightly lower than the MatterSim ones. Figure 8 compares the energies above hull of generated structures from MatterSim and DFT. There is again a very strong correlation between the two results, suggesting that the use of MatterSim can lead to reasonable conclusions. All DFT relaxations utilized `MPGGADoubleRelaxStatic` flows from the Atomate2 Ganose et al. (2025) package to produce MP20-compatible data.

C.4 Stability and structural analysis of generated structures

In Fig. 9, we show the distribution of computed energies above the convex hull across various OMatG models, showing best stability of generated structures for linear, encoder-decoder, trigonometric, and SBD positional interpolants.

By evaluating the distribution of N-ary structures (Fig. 10), the distribution of average coordination numbers (both by structure in Fig. 11 and by species in Fig. 12), as well as distribution of crystal systems (Fig. 13) which are related to a structure's Bravais lattice, we provide qualitative analysis for model performance. Space groups (and thus crystal systems) were determined using the `spglib` software (Togo et al., 2024). Generally, OMatG and DiffCSP showed the best performance in matching the distribution of average coordination number for each structure, particularly for high-coordinated structures. DiffCSP performed most poorly among the four models on the N-ary structural similarity. The average coordination number for species were best-matched by the two OMatG models, and of these two the linear interpolant generated the distribution of crystal systems best.

MatterSim S.U.N.	DFT S.U.N.
0.16	0.09
0.13	0.12
0.06	0.06
0.12	0.10
0.14	0.12
0.13	0.12
0.16	0.15
0.10	0.09
0.13	0.13
0.11	0.08

Table 11: Comparison of MatterSim based and DFT based S.U.N. rates (reported as a fraction) for 10 different sets of 100 randomly selected generated structures.

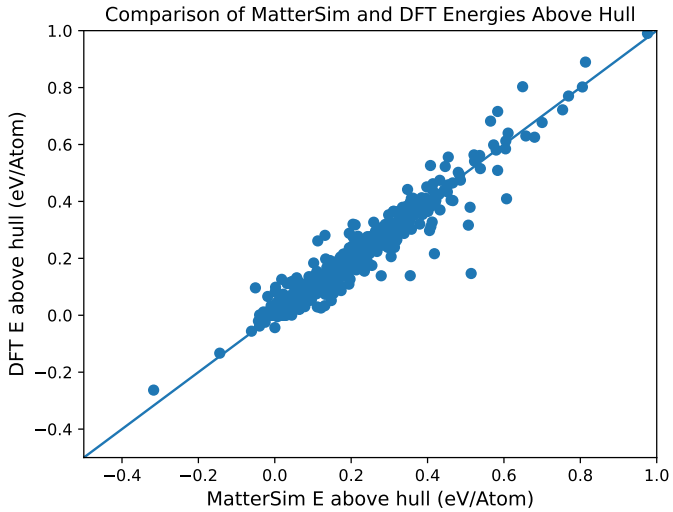

Figure 8: Comparison of energies above hull for generated structures computed by MatterSim and DFT. There is strong correlation between the two methods suggesting that MatterSim is a cost effective method for evaluating the stability of generated structures.

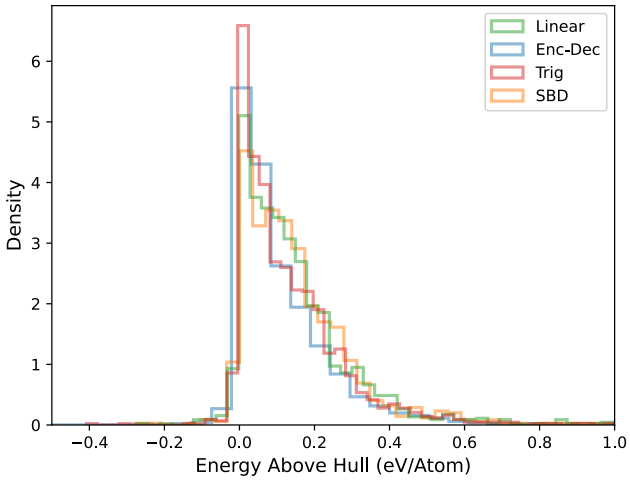

Figure 9: Histogram of the computed energies above the convex hull for structures generated by four OMatG DNG models. We show that all positional interpolants are effective at generating structures close to the convex hull, with the SBD interpolant performing slightly worse than other interpolants.

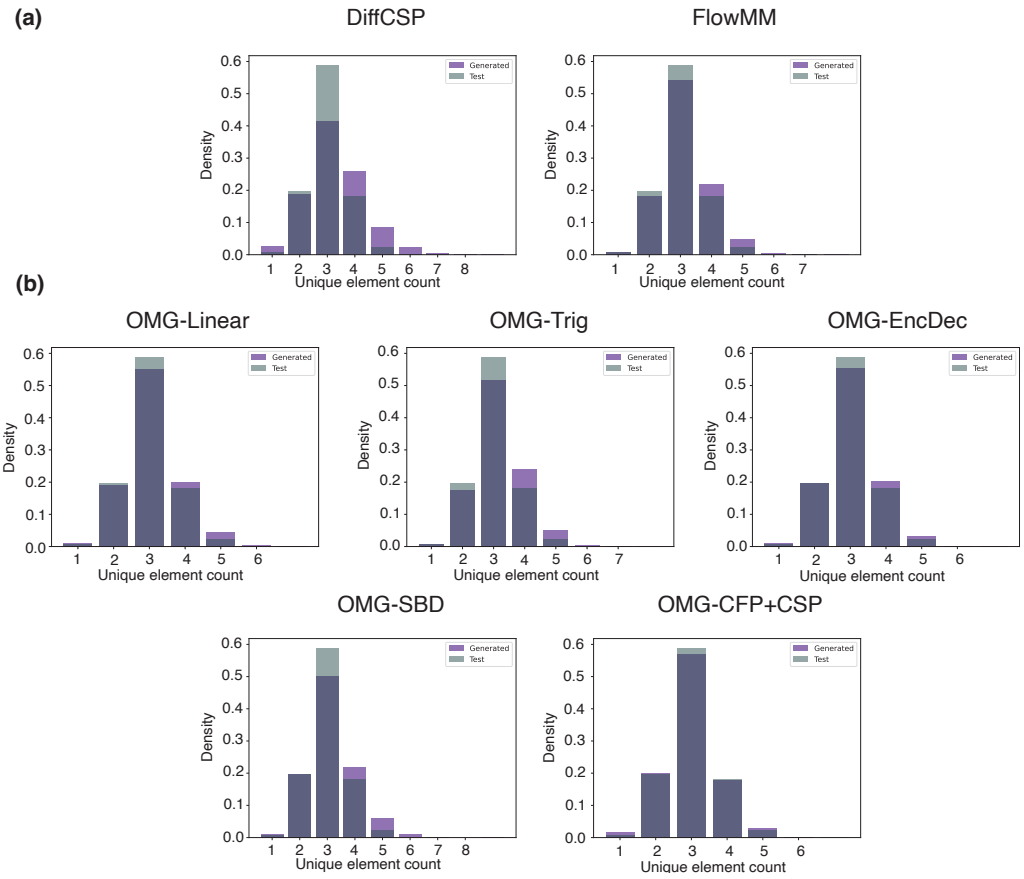

Figure 10: Qualitative performance of the distribution of N-ary crystals for **(a)** Non-OMatG models and **(b)** OMatG models across structural benchmarks computed on generated structures and test set structures from the *MP-20* dataset.

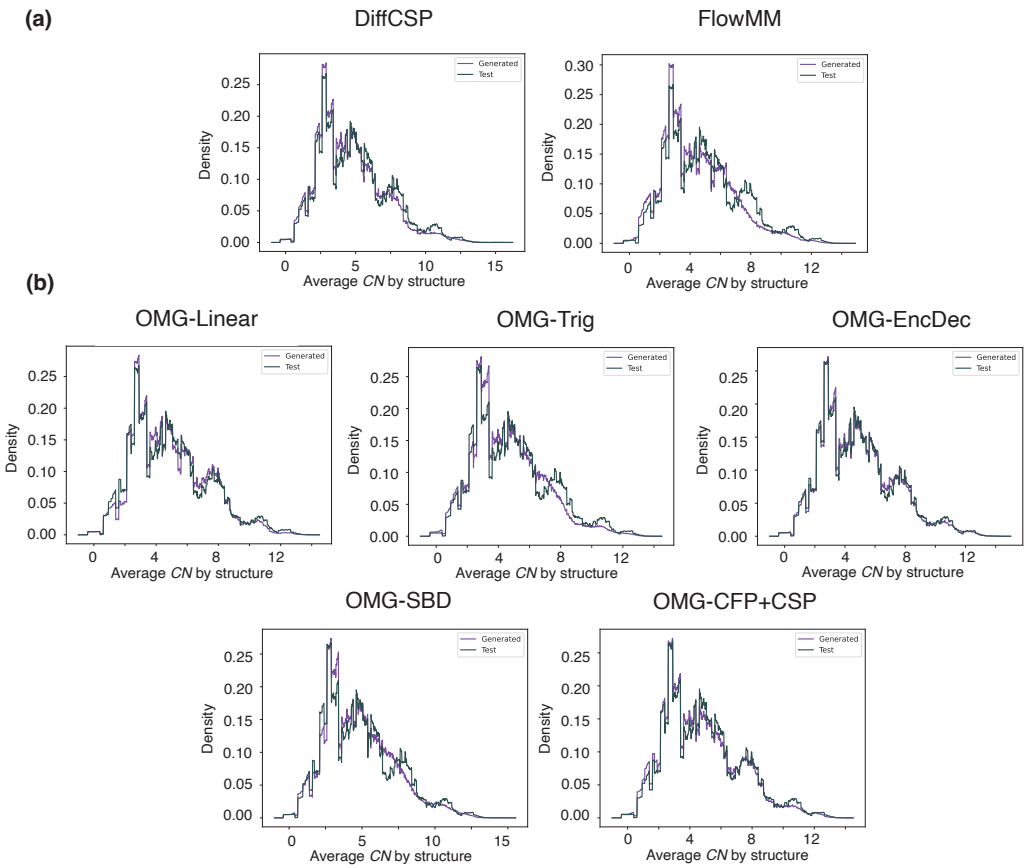

Figure 11: Qualitative performance of the distribution of average coordination number by structure for **(a)** Non-OMatG models and **(b)** OMatG models across structural benchmarks computed on generated structures and test set structures from the *MP-20* dataset.

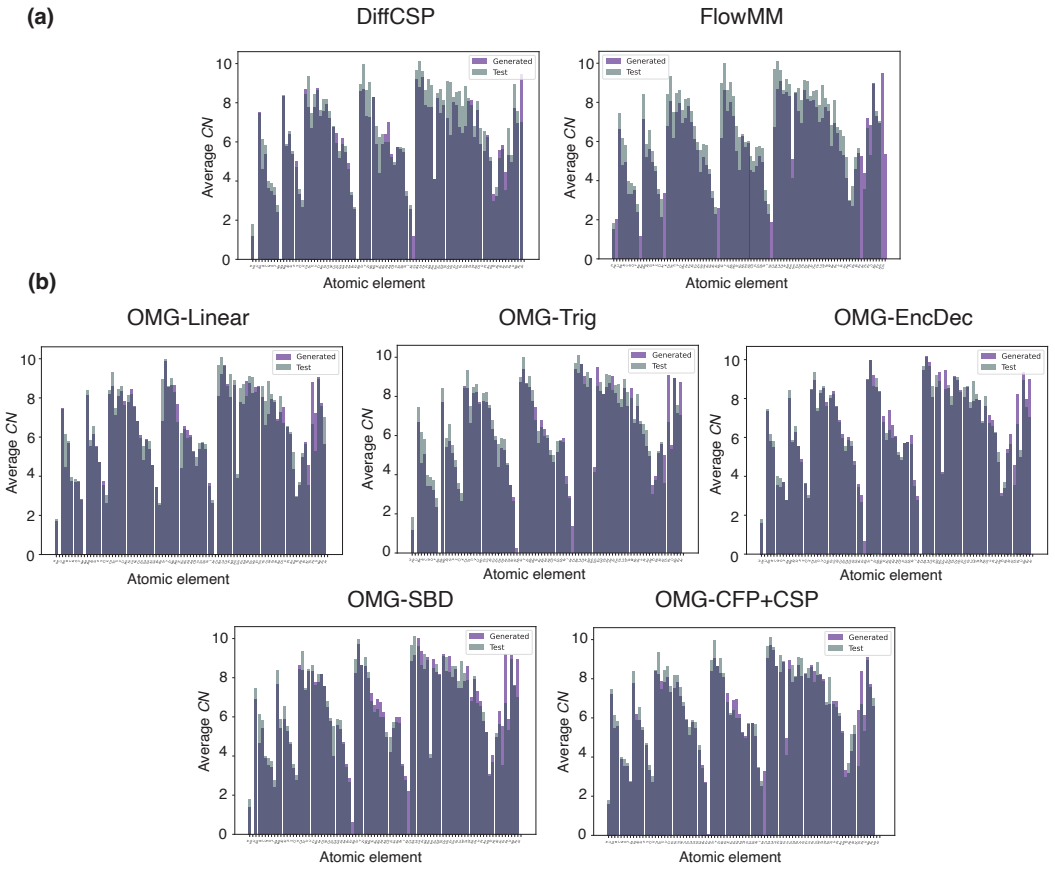

Figure 12: Qualitative performance of the distribution of average coordination number by species (listed left to right in order of atomic number) for **(a)** Non-OMatG models and **(b)** OMatG models across structural benchmarks computed on generated structures and test set structures from the *MP-20* dataset.

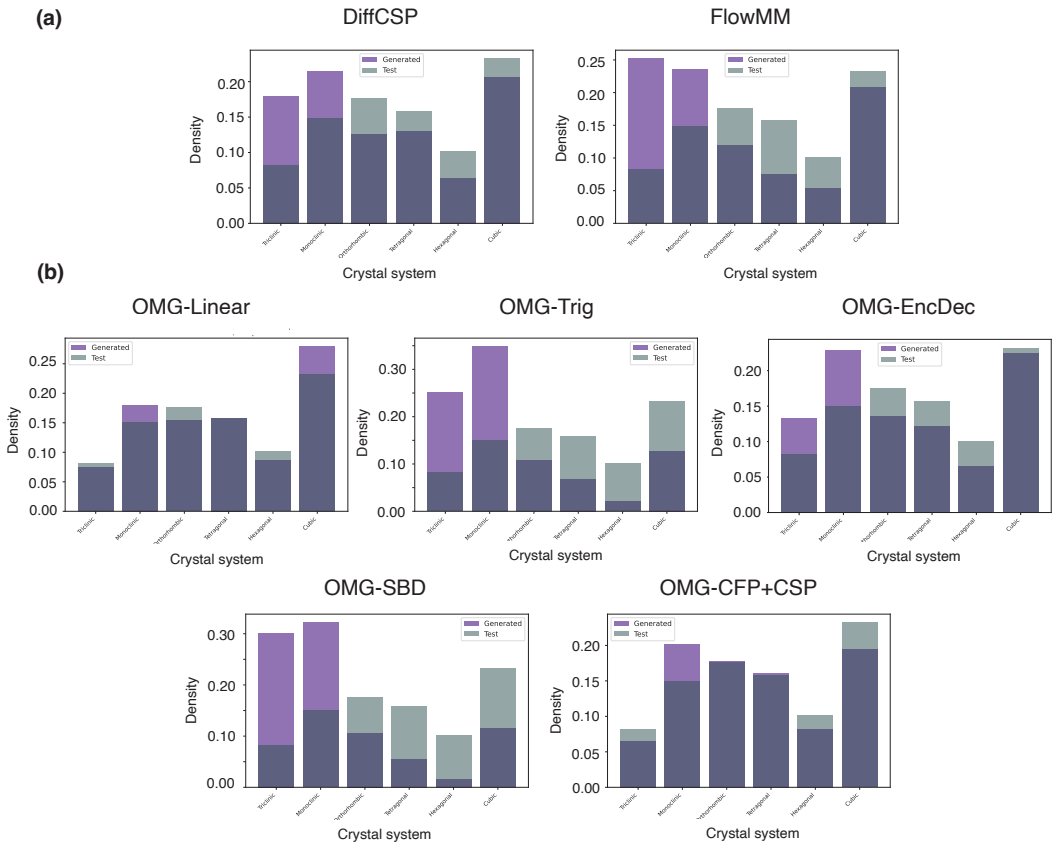

Figure 13: Qualitative performance of the distribution of crystal system by structure for **(a)** Non-OMatG models and **(b)** OMatG models across structural benchmarks computed on generated structures and test set structures from the *MP-20* dataset.

Table 12: FlowLLM's and OMatG-LLM's performance (with linear and the trigonometric interpolants) when using the same fine-tuned LLM (Gruver et al., 2024) as the base distribution. The best performance for each metric is in bold.

Method	**Validity** (% ↑)			**Coverage** (% ↑)		**Property** (↓)		
	Structural	Composition	Combined	Recall	Precision	wdist (ρ)	wdist (Nary)	wdist ($\langle CN \rangle$)
FlowLLM	96.27	86.40	83.55	97.98	96.55	0.9922	0.5427	0.5936
OMatG-LLM								
Linear	**97.86**	86.40	**84.85**	**99.16**	**98.40**	**0.9100**	0.5427	0.8600
Trigonometric	**97.78**	86.40	**84.72**	97.41	**99.12**	3.6214	0.5427	**0.4448**

D Large Language Models as Base Distributions

FlowLLM (Sriram et al., 2024) combines large language models (LLMs) with the conditional flow-matching framework FlowMM (Miller et al., 2024) to design novel crystalline materials. A fine-tuned LLM serves as the base distribution and samples initial structures; FlowMM then refines the fractional coordinates and lattice parameters as in the CSP task. This idea can be similarly applied to OMatG, which then allows to combine LLMs with the general SI framework for materials generation.

We extend OMatG to OMatG-LLM by allowing for LLM-generated structures as the initial structures. We evaluate both FlowLLM and OMatG-LLM on the LLM dataset released by FlowLLM. Specifically, we use the training (containing $40\,000$ structures) and validation sets (6000 structures) from `https://github.com/facebookresearch/flowmm` and the LLM-generated initial structures ($10\,000$ structures) from `https://github.com/facebookresearch/crystal-text-llm` as the test set. These initial structures are generated by a fine-tuned Llama-70B model (Gruver et al., 2024). As shown in Tab. 12, OMatG-LLM's linear and trigonometric interpolants outperform FlowLLM in almost all DNG metrics. Since the Wasserstein distance with respect to the N-arity distributions and the compositional validity only depend on the atomic species generated by the LLM, these two metrics are necessarily equal for FlowLLM and OMatG-LLM. Note that the original FlowLLM Sriram et al. (2024) is trained on 3M LLM-generated structures while our experiments are conducted on the 40K structures open-sourced by the authors. The performance of FlowLLM in our experiments thus differs from the scores reported in Sriram et al. (2024).