FlowMM: Generating Materials with Riemannian Flow Matching

Benjamin Kurt Miller [1 2] **Ricky T. Q. Chen** [2] **Anuroop Sriram** [2] **Brandon M. Wood** [2]

Abstract

Crystalline materials are a fundamental component in next-generation technologies, yet modeling their distribution presents unique computational challenges. Of the plausible arrangements of atoms in a periodic lattice only a vanishingly small percentage are thermodynamically stable, which is a key indicator of the materials that can be experimentally realized. Two fundamental tasks in this area are to (a) predict the stable crystal structure of a known composition of elements and (b) propose novel compositions along with their stable structures. We present FlowMM, a pair of generative models that achieve state-of-the-art performance on both tasks while being more efficient and more flexible than competing methods. We generalize Riemannian Flow Matching to suit the symmetries inherent to crystals: translation, rotation, permutation, and periodic boundary conditions. Our framework enables the freedom to choose the flow base distributions, drastically simplifying the problem of learning crystal structures compared with diffusion models. In addition to standard benchmarks, we validate FlowMM's generated structures with quantum chemistry calculations, demonstrating that it is ∼3x more efficient, in terms of integration steps, at finding stable materials compared to previous open methods.

1. Introduction

Materials discovery has played a critical role in key technological developments across history (Appl, 1982). The huge number of plausible materials offers the potential to advance key areas such as energy storage (Ling, 2022), carbon capture (Hu et al., 2023), and microprocessing (Choubisa et al., 2023). With the promise comes a challenge: The search space of crystal structures is combinatorial in the number of atoms and elements under consideration, resulting in an intractable number of potential structures. Of these, only a vanishing small percentage will be experimentally realizable. These factors have motivated computational exploration of the materials design space, which has accelerated discovery campaigns, but typically relies on random structure search (Potyrailo et al., 2011) and expensive quantum mechanical computations. Generative modeling has shown promise for navigating large design spaces for scientific research, but the application to materials is still relatively novel.

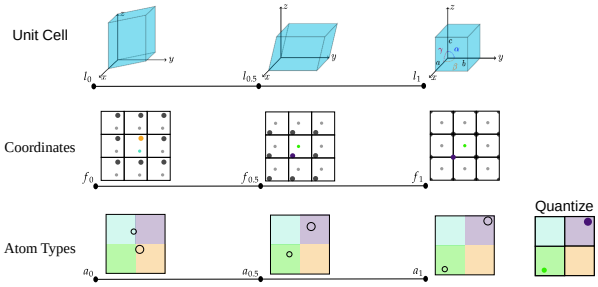


Figure 1. A conceptual representation of the evolution from base distribution to target distribution, according the vector field learned by our model. We model a joint distribution over lattice parameters, periodic fractional coordinates, and a binary representation of atom type. The highlights include symmetry-aware geodesic paths and a base distribution that directly produces plausible lattices. Note that coordinates and atom types are merely depicted in 2d for clarity.

The specific task we will focus on in this paper is the generation of periodic crystals. We're interested in two cases: *Crystal Structure Prediction* (CSP), which we define as the setting where the user provides the number of constituent atoms and their elemental types, i.e. their *composition*, and *De Novo Generation* (DNG) when the generative model is tasked to produce the composition alongside its crystal structure. An additional complexity is that we want to generate crystals that are *stable*. Naively, stability is determined by a thermodynamic competition between a structure and competing alternatives. The known stable structures define a *convex hull* of stable compositions over the energy landscape. We use the heuristic that the stability of a material is determined by its energetic distance to the convex hull, denoted E_{hull}. Structures with $E_{hull} \leq 0.0$ eV/atom are considered stable. Otherwise, the structure is unfavorable and will decompose into a stable neighboring composition.

Research done while BKM was an intern at FAIR, Meta AI. [1]University of Amsterdam [2]FAIR, Meta AI. Correspondence to: BKM <b.k.miller@uva.nl>, BW <bmwood@meta.com>.

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

Modeling crystals is challenging because it requires fitting distributions jointly over several variables, each with different and interdependent structure. The atom types are discrete, while the *unit cell*, i.e. the repeating fundamental volume element, and the atomic positions are continuous. Side lengths of the unit cell are strictly positive, the angles are bounded, and the atomic coordinates are periodic. The number of atoms varies, but the unit cell dimensionality is fixed.

While diffusion models found some success in generating stable materials they have fundamental limitations, making them suboptimal for the problem. In graph neural network implementations, each variable has required a different diffusion framework to fit its structure (Jiao et al., 2023; Zeni et al., 2023). Atomic coordinates are modeled using score matching (Song et al., 2020), enabling a wrapped normal transition distribution and a uniform asymptotic distribution, atomic types utilize discrete diffusion (Austin et al., 2021), and the unit cell is modeled with denoising diffusion (Sohl-Dickstein et al., 2015; Ho et al., 2020). Both of these approaches must choose a complex representation for the lattice in order to fit within the diffusion framework since their limiting distribution must be normal, yet still represent a "realistic material" while doing diffusion. Furthermore Jiao et al. (2023), must perform an ad hoc Monte Carlo estimate of a time evolution scaling factor (top of page 6 in their paper), and they do not simulate the forward trajectory in training. This simulation is a slow, but necessary step, in rigorous diffusion models on manifolds (Huang et al., 2022).

Our contribution We introduce *FlowMM*, a pair of generative models for CSP and DNG that each jointly estimate symmetric distributions over fractional atomic coordinates and the unit cell (along with atomic types for DNG) in a single framework based on Riemannian Flow Matching (Lipman et al., 2022; Chen & Lipman, 2024). We train a Continuous Normalizing Flow (Chen et al., 2018) with a finite time evolution and produce high-quality samples, as measured by standard metrics and thermodynamic stability, with significantly fewer integration steps than diffusion models.

First, we generalize the Riemannian flow matching framework to estimate a point cloud density that is invariant to translation with periodic boundary conditions, a novel achievement for continuous normalizing flows, by proposing a new objective in Section 3. With this step, it becomes possible to enforce isometric invariances inherent to the geometry of crystals as an inductive bias in the generative model. Second, after selecting a rotation invariant representation, we choose a natural base distribution that samples plausible unit cells by design. We find that this drastically simplifies fitting the lattice compared with diffusion models, which are forced to take an unnatural base distribution due to inherent limitations in their framework. Third, for DNG, we choose a binary representation (Chen et al., 2022) for the atom types that drastically reduces the dimensionality compared with the simplex (one-hot). Our representation is $\lceil \log_2(100) \rceil = 7$ dimensions per atom, while the simplex requires 100 dimensions per atom. (Note that $\lceil \cdot \rceil$ denotes the ceiling function.) We attribute the accuracy of FlowMM in predicting the number of unique elements per unit cell to this design choice, something other models struggle with. Finally, we compare our method to diffusion model baselines with extensive experiments on two realistic datasets and two simplified unit tests. In addition to competitive or state-of-the-art performance on standard metrics, we take the additional step to estimate thermodynamic stability of generated structures by performing expensive quantum chemistry calculations. We find that FlowMM is able to generate materials with comparable stability to these other methods, while being *significantly* faster at inference time.

We made our code publicly available on `GitHub`[1].

Related work The earliest approaches for both CSP and DNG new materials rely on proposing a large number of possible candidate materials and screening them with high-throughput quantum mechanical (Kohn & Sham, 1965) calculations to estimate the energy and find stable materials. Those candidate materials are proposed using simple replacement rules (Wang et al., 2021) or accelerated by genetic algorithms (Glass et al., 2006; Pickard & Needs, 2011). This search can be accelerated using machine learned models to predict energy (Schmidt et al., 2022; Merchant et al., 2023).

Various generative models have been designed to accelerate material discovery by directly generating materials, avoiding expensive brute force search (Court et al., 2020; Yang et al., 2021; Nouira et al., 2018). Diffusion models have been widely used for this task. Initially without diffusing the lattice but predicting it with a variational autoencoder (Xie et al., 2021); later by jointly diffusing the positions, lattice structure and atom types (Jiao et al., 2023; Yang et al., 2023; Zeni et al., 2023). We only compare to open models with verifiable results at the time of submission of this paper to ICML 2024 in January. More recently, other models have used space groups as additional inductive bias (AI4Science et al., 2023; Jiao et al., 2024; Cao et al., 2024). Other approaches include using large language models (Flam-Shepherd & Aspuru-Guzik, 2023; Gruver et al., 2024), and with normalizing flows Wirnsberger et al. (2022).

2. Preliminaries

We are concerned with fitting probability distributions over crystal lattices, which are collections of atoms periodically arranged to fill space in a repeated pattern. One way to construct a three-dimensional crystal is by tiling a paral-

[1] https://github.com/facebookresearch/flowmm

lelepiped, or *unit cell*, such that the entirety of space is covered. The unit cell contains an arrangement of atoms, i.e., a labeled point cloud, which produces the crystal lattice under the tiling. The following is the exposition towards our generative model. As background, we recommend a primer on the symmetries of crystals (Adams & Orbanz, 2023).

2.1. Representing crystals and their symmetries

Representation of a crystal We label the particles in a crystal $\tilde{\boldsymbol{c}} \in \tilde{\mathcal{C}}$ with $n \in \mathbb{N}$ sites (atoms) by column in matrix $\boldsymbol{a} \coloneqq \left[a^1, \ldots, a^n\right] \in \mathcal{A}$ where each atomic number maps to a unique h-dimensional categorical vector. The unit cell is a matrix of Cartesian column vectors $\tilde{\boldsymbol{l}} \coloneqq \left[\tilde{l}^1, \tilde{l}^2, \tilde{l}^3\right] \in \tilde{\mathcal{L}} = \mathbb{R}^{3\times 3}$. The position of each particle can be represented using a matrix $\boldsymbol{x} \coloneqq \left[x^1, \ldots, x^n\right] \in \mathcal{X} = \mathbb{R}^{3\times n}$ with Cartesian coordinates in the rows and particles in the columns, but we will choose an alternative fractional representation that leverages the periodic boundary of $\tilde{\boldsymbol{l}}$. Positions are equivalently $\boldsymbol{x} = \tilde{\boldsymbol{l}}\boldsymbol{f}$ where $\boldsymbol{f} \coloneqq \tilde{\boldsymbol{l}}^{-1}\boldsymbol{x} = \left[f^1, \ldots, f^n\right] \in \mathcal{F} = [0,1)^{3\times n}$. The volume of a unit cell $\mathrm{Vol}(\tilde{\boldsymbol{l}}) \coloneqq |\det \tilde{\boldsymbol{l}}|$ must be nonzero, implying that $\tilde{\boldsymbol{l}}$ is invertible. Then, the crystal is

$$\left\{ \left(\boldsymbol{a}', \boldsymbol{f}'\right) \,\middle|\, \boldsymbol{a}' = \boldsymbol{a}, \boldsymbol{f}' = \boldsymbol{f} + \tilde{\boldsymbol{l}}k\mathbf{1}, \forall k \in \mathbb{Z}^{3\times 1} \right\}$$

where elements of k denote integer translations of the lattice and $\mathbf{1}$ is a $1 \times n$ matrix of ones to emulate "broadcasting."

This representation is not unique since there is freedom to choose an alternative unit cell and corresponding fractional coordinates producing the same crystal lattice, e.g. by doubling the volume or skewing the unit cell. Among minimum volume unit cells, we choose $\tilde{\boldsymbol{l}}$ to be the unique one determined by Niggli reduction (Grosse-Kunstleve et al., 2004).

Equivariance A function $f\colon \mathcal{S} \to \mathcal{S}'$ is *G-equivariant* when for any $s \in \mathcal{S}$ and any $g \in G$ we have $f(g \cdot s) = g \cdot f(s)$ where $\cdot$ indicates group action in the relevant space. *G-invariant* functions have instead $f(g \cdot s) = f(s)$. We will apply graph neural networks (Satorras et al., 2021) with these properties (Thomas et al., 2018; Kondor & Trivedi, 2018; Miller et al., 2020; Weiler et al., 2021; Geiger & Smidt, 2022; Liao et al., 2023; Passaro & Zitnick, 2023).

Invariant density A density $p\colon \mathcal{S} \to \mathbb{R}^+$ is *G-invariant* when p is G-invariant. When a G-invariant density p is transformed by an invertible, G-equivariant function f the resulting density is G-invariant (Köhler et al., 2020).

Symmetries of crystals We will now introduce relevant symmetry groups and their action on our crystal representation $\tilde{\boldsymbol{c}} \coloneqq (\boldsymbol{a}, \boldsymbol{f}, \tilde{\boldsymbol{l}}) \in \mathcal{A} \times \mathcal{F} \times \tilde{\mathcal{L}} =: \tilde{\mathcal{C}}$. The symmetric group S_n on n atoms has $n!$ elements corresponding to all permutations of atomic index. The element $\sigma \in S_n$ acts on a crystal by $\sigma \cdot \tilde{\boldsymbol{c}} = \left(\left[a^{\sigma(1)}, \ldots, a^{\sigma(n)}\right], \left[f^{\sigma(1)}, \ldots, f^{\sigma(n)}\right], \tilde{\boldsymbol{l}}\right)$. The special Euclidean group $\mathrm{SE}(3)$ consists of orientation preserving rigid rotations and translations in Euclidean space. The elements $(Q, \tau) \in \mathrm{SE}(3)$ can be decomposed into rotation $Q \in \mathrm{SO}(3)$, represented by a 3×3 rotation matrix, and translation $\tau \in [-\frac{1}{2}, \frac{1}{2}]^{3\times 1}$. Considering the periodic unit cell, the action on our crystal representation is $\tau \cdot \tilde{\boldsymbol{c}} = (\boldsymbol{a}, \boldsymbol{f} + \tau\mathbf{1} - \lfloor \boldsymbol{f} + \tau\mathbf{1} \rfloor, \tilde{\boldsymbol{l}})$, where $\lfloor \cdot \rfloor$ denotes the element wise floor function, i.e., $\boldsymbol{f}$ coordinates "wrap around." The rotation action is $Q \cdot \boldsymbol{c} \coloneqq (\boldsymbol{a}, \boldsymbol{f}, Q\tilde{\boldsymbol{l}})$.

The true distribution $q(\tilde{\boldsymbol{c}})$ has invariances to group operations that we would like our estimated density to inherit:

$$q(\sigma \cdot \tilde{\boldsymbol{c}}) = q(\sigma \cdot \boldsymbol{a}, \sigma \cdot \boldsymbol{f}, \tilde{\boldsymbol{l}}) = q(\tilde{\boldsymbol{c}}), \qquad \forall \sigma \in S_n \quad (1)$$

$$\begin{aligned} q(\tau \cdot \tilde{\boldsymbol{c}}) &= q(\boldsymbol{a}, \boldsymbol{f} + \tau\mathbf{1} - \lfloor \boldsymbol{f} + \tau\mathbf{1} \rfloor, \tilde{\boldsymbol{l}}) \\ &= q(\tilde{\boldsymbol{c}}), \qquad \forall \tau \in [-\tfrac{1}{2}, \tfrac{1}{2}]^3 \quad (2) \end{aligned}$$

$$q(Q \cdot \tilde{\boldsymbol{c}}) = q(\boldsymbol{a}, \boldsymbol{f}, Q\tilde{\boldsymbol{l}}) = q(\tilde{\boldsymbol{c}}), \qquad Q \in \mathrm{SO}(3) \quad (3)$$

permutation, translation, and rotation invariance, respectively. We address (1) and (2) in Section 3, but (3) here.

Probability distributions over a G-invariant representation are necessarily G-invariant. Lattice parameters $\boldsymbol{l} \in \mathcal{L}$ are a rotation invariant representation of the unit cell as a 6-tuple of three side lengths $a, b, c \in \mathbb{R}^+$ with units of Å and three internal angles $\alpha, \beta, \gamma \in [60°, 120°]$ in degrees. (This range for the lattice angles is due to the Niggli reduction.) We therefore propose a rotation invariant alternative crystal representation $\boldsymbol{c} \in \mathcal{C} \coloneqq \mathcal{A} \times \mathcal{F} \times \mathcal{L}$ and thereby any distribution $p(\boldsymbol{c})$ is rotation invariant:

$$Q \cdot \boldsymbol{c} = (\boldsymbol{a}, \boldsymbol{f}, Q \cdot \boldsymbol{l}) = (\boldsymbol{a}, \boldsymbol{f}, \boldsymbol{l}) = \boldsymbol{c}, \quad p(Q \cdot \boldsymbol{c}) = q(\boldsymbol{c}).$$

$\boldsymbol{l}$ carries all non-orientation information about the unit cell. By composing functions $\mathfrak{U}\colon \tilde{\boldsymbol{l}} \to \boldsymbol{l}$ and $\mathfrak{U}^\dagger\colon \boldsymbol{l} \to \tilde{\boldsymbol{l}}$, implemented by Ong et al. (2013), we can reconstruct $\tilde{\boldsymbol{l}}$ from $\boldsymbol{l}$ up to rotation, i.e. $\mathfrak{U}^\dagger(\mathfrak{U}(\tilde{\boldsymbol{l}})) = Q\tilde{\boldsymbol{l}}$ for some $Q \in \mathrm{SO}(3)$. Further symmetry information is in Appendix A.

Representing atomic types The representation of $\boldsymbol{a} \in \mathcal{A}$ depends on whether the generative model is doing CSP or DNG. For CSP, the atomic types are only conditional information and may be considered a tuple of n, h-dimensional one-hot vectors. For DNG, the generative model treats $\boldsymbol{a}$ as a random variable and learns a distribution over its representation. We choose to apply the binarization method proposed by Chen et al. (2022) where categorical vector $\boldsymbol{a}$ is mapped to a $\{-1, 1\}$-bit representation of length $\lceil \log_2 h \rceil$. The flow then learns to transform a $\lceil \log_2 h \rceil$-dimensional normal distribution to the corresponding bit representation and, at inference time, is discretized by the $\mathrm{sign}\colon \mathbb{R} \to \{-1, 1\}$ function. When $\lceil \log_2 h \rceil \neq \log_2 h$, we end up with "unused bits", i.e. we can represent more than h classes. We find that the model is able to learn to ignore these extra atom

types in practice. Note that Chen et al. (2022) suggest using a self-conditioning scheme, but we did not find it necessary.

2.2. Learning distributions with Flow Matching

Riemannian Manifolds (Abridged) In order to learn probability distributions over spaces that "wrap around," we must introduce smooth, connected Riemannian manifolds $\mathcal{M}$. They relax the notion of a global coordinate system in lieu of a collection of local coordinate systems that transition seamlessly between one another. Every point $m \in \mathcal{M}$ has an associated *tangent space* $\mathcal{T}_m\mathcal{M}$ with an inner product $\langle u, v\rangle$ for $u, v \in \mathcal{T}_m\mathcal{M}$, enabling the definition of distances, volumes, angles, and minimum length curves (*geodesics*). A fundamental building block for our generative model is the space of time-dependent, smooth vector fields $\mathcal{U}$ living on the manifold. Elements $u_t \in \mathcal{U}$ are maps $u_t : [0, 1] \times \mathcal{M} \to \mathcal{T}\mathcal{M}$ where the first argument t denotes time and $\mathcal{T}\mathcal{M} := \cup_{m\in\mathcal{M}}\{m\}\times\mathcal{T}_m\mathcal{M}$ denotes the *tangent bundle*, i.e., the disjoint union of each tangent space at every point on the manifold. We learn distributions by estimating functions in $\mathcal{U}$.

The geodesics for any $\mathcal{M}$ that we consider can be written in closed form using the exponential and logarithmic maps. The geodesic connecting $m_0, m_1 \in \mathcal{M}$ at time $t \in [0, 1]$ is

$$m_t := \exp_{m_0}(t \log_{m_0}(m_1)), \tag{4}$$

where $\exp_\square$ and $\log_\square$ depend on the manifold $\mathcal{M}$.

Probability paths on flat manifolds Probability densities on $\mathcal{M}$ [2] are continuous functions $p\colon \mathcal{M} \to \mathbb{R}^+$ where $\int_\mathcal{M} p(m)\, dm = 1$ and $p \in \mathcal{P}$, the space of probability densities on $\mathcal{M}$. A *probability path* $p_t : [0, 1] \to \mathcal{P}$ is a continuous time-dependent curve in probability space linking two densities $p_0, p_1 \in \mathcal{P}$ at endpoints $t = 0$, $t = 1$. A *flow* $\psi_t\colon [0, 1] \times \mathcal{M} \to \mathcal{M}$ is a time-dependent diffeomorphism defined to be the solution to the ordinary differential equation: $\frac{d}{dt}\psi_t(m) = u_t(\psi_t(m))$, subject to initial conditions $\psi_0(m) = m_0$ with $u_t \in \mathcal{U}$. A flow ψ_t is said to generate p_t if it pushes p_0 forward to p_1 following the time-dependent vector field u_t. The path is denoted $p_t = [\psi_t]_\# p_0 := p_0(\psi_t^{-1}(m)) \det \left|\frac{\partial \psi_t^{-1}(m)}{\partial m}(m)\right|$ for our choice of flat $\mathcal{M}$ (Mathieu & Nickel, 2020; Gemici et al., 2016; Falorsi & Forré, 2020). Chen et al. (2018) proposed to model ψ_t implicitly by parameterizing u_t to produce p_t in a method known as a *Continuous Normalizing Flow* (CNF).

Flow Matching Fitting a CNF using maximum likelihood, as in the style of (Chen et al., 2018), can be expensive and unstable. A more effective alternative, fitting a vector field $v_t^\theta \in \mathcal{U}$ with parameters θ, may be accomplished by doing regression on vector fields u_t that are known *a priori* to generate p_t. The method is known as *Flow Matching* (Lipman et al., 2022) and was extended to $\mathcal{M}$ by Chen & Lipman (2024). Lipman et al. (2022) note that u_t is generally intractable and formulate an alternative objective based on tractable, conditional vector fields $u_t(m \mid m_1)$ that generate conditional probability paths $p_t(m \mid m_1)$, the push-forward of the conditional flow $\psi_t(m \mid m_1)$, starting at any p and concentrating around $m = m_1 \in \mathcal{M}$ at $t = 1$. Marginalizing over target distribution q recovers the unconditional probability path $p_t(m) = \int_\mathcal{M} p_t(m \mid m_1)q(m_1)\, dm_1$ and the unconditional vector field $u_t(m) = \int_\mathcal{M} u_t(m \mid m_1)\frac{p_t(m|m_1)q(m_1)}{p_t(m)}\, dm_1$. This construction results in an unconditional path p_t where $p_0 = p$, the chosen base distribution, and $p_1 = q$, the target distribution. Their proposed objective, simplified for flat manifolds $\mathcal{M}$, is:

$$\mathfrak{L}(\theta) = \mathbb{E}_{t,q(m_1),p_t(m|m_1)}\|v_t^\theta(m) - u_t(m \mid m_1)\|^2, \tag{5}$$

where the $\|\cdot\|$ norm is induced by inner product $\langle\cdot,\cdot\rangle$ on $\mathcal{T}_m\mathcal{M}$ and $t \sim \text{Uniform}(0, 1)$. At optimum, v_t^θ generates $p_t^\theta = p_t$ with endpoints $p_0^\theta = p$, $p_1^\theta = q$. At inference, we sample p and propagate t from 0 to 1 using our estimated v_t^θ.

2.3. Crystalline Solids

Stability and the convex hull One of the most important properties of a material is its stability, a heuristic that gives a strong indication of its synthesizability. A crystal is stable when it is energetically favorable compared with competing *phases*, structures built from the same atomic constituents, but in different proportion or spacial arrangement. The energy can be computed using a first-principles quantum mechanical method called density functional theory, which estimates the energy based on the electronic structure (Kohn & Sham, 1965). The lowest energy materials form a convex hull over composition. Stable structures lie directly on the convex hull or below it, while meta-stable structures are restricted to $E^{hull} < 0.08$ eV/atom. Note that, this definition has inherent epistemic uncertainty since many materials are unknown and not represented on the convex hull. Our specific convex hull is in reference to the Materials Project database, as recorded by Riebesell (2024) in February 2023.

Arity for materials A material with N unique atom type constituents is known as an N-*ary* material and its stability is determined by an N-dimensional convex hull. High N-ary materials occupy convex hulls that not represented in the realistic datasets under consideration; the curse of dimensionality and chemical complexity limits coverage of these hulls. We posit that an effective generative model of stable structures will produce a distribution over N-ary that is close to the data distribution. This is borne out in our experiments as seen in Figure 3 and in Appendix B. On another note, several generative models produced 1-ary structures marked

[2] We consider flat tori and Euclidean space, restricting ourselves to manifolds with flat metrics, i.e. the metric is the identity matrix.

stable by the $E^{hull} < 0$ criterion. This is not possible as the one-element phase diagrams are known and there are no energetically favorable structures to be found. This is a numerical issue; we did not count those structures as stable.

2.4. Problem statements & Datasets

Crystal Structure Prediction (CSP) We aim to predict the stable structure for a given composition $\boldsymbol{a}$, but this is not well-posed because some compositions have no stable arrangement. Furthermore, the underlying energy calculations have uncertainty and some structures are degenerate. We therefore formulate CSP as a conditional generative task on metastable structures, distributed like $q(\boldsymbol{f}, \boldsymbol{l}; \boldsymbol{a})$, rather than via regression on stable structures. Our dataset consists of metastable structures, indexed by composition and count:

$$\boldsymbol{f}_{ij}, \boldsymbol{l}_{ij} \in \left\{ \boldsymbol{f}' \in \mathcal{F}, \boldsymbol{l}' \in \mathcal{L} \mid E(\boldsymbol{a}_i, \boldsymbol{f}', \boldsymbol{l}') < E_{\mathrm{m}} \right\}. \quad (6)$$

$E_{\mathrm{m}} \coloneqq 0.08$ eV/atom is fixed by metastability, $E \colon \mathcal{C} \to \mathbb{R}$ is the single point energy prediction of density functional theory, $\boldsymbol{a}_i$ is a composition with index i, and $\boldsymbol{f}'_{ij}, \boldsymbol{l}'_{ij}$ are the corresponding metastable structures with index j. The maximum values of i and j are the number of metastable compositions and structures for that composition, respectively.

In CSP, we fit a generative model $p(\boldsymbol{f}, \boldsymbol{l} \mid \boldsymbol{a})$ to the samples $\boldsymbol{f}_{ij}, \boldsymbol{l}_{ij} \sim q(\boldsymbol{f}, \boldsymbol{l}; \boldsymbol{a}_i)$. Given the right $\boldsymbol{a}$, a good generative model should generate corresponding metastable structures.

De novo generation (DNG) A major goal of materials science is to discover stable and novel crystals. In this effort, we aim to sample directly from a distribution of metastable materials $q(\boldsymbol{c})$, generating both the structure $\boldsymbol{f}, \boldsymbol{l}$ along with the composition $\boldsymbol{a}$. Our distribution must include $\boldsymbol{a}$ because it should avoid compositions that have no (meta)stable structure and because many new and interesting materials may have novel compositions. Define our dataset:

$$\boldsymbol{a}_k, \boldsymbol{f}_k, \boldsymbol{l}_k \coloneqq \boldsymbol{c}_k \in \{\boldsymbol{c}' \in \mathcal{C} \mid E(\boldsymbol{c}') < E_{\mathrm{m}}\}, \quad (7)$$

consisting of $\max k$ metastable crystals.

In DNG, we fit a generative model $p(\boldsymbol{c})$ to samples $\boldsymbol{c}_k \sim q(\boldsymbol{c})$. A good generative model should generate both novel and known metastable materials. Determining stability and novelty requires further computation and a convex hull.

Practical Considerations & Data We consider two realistic datasets: *MP-20*, containing all materials with a maximum of 20 atoms per unit cell and within 0.08 eV/atom of the convex hull in the Materials Project database from around July 2021 (Jain et al., 2013), and *MPTS-52*, a challenging dataset containing structures with up to 52 atoms per unit cell and separated into "time slices" where the training, validation, and test sets are organized chronologically by earliest published year in literature (Baird et al., 2024). We include two additional datasets *Perov-5* (Castelli et al., 2012) and *Carbon-24* (Pickard, 2020) as unit tests. These do not feature crystals near their energy minima. Perov-5 consists of crystals with varying atomic types, but all structures have the same fractional coordinates. Carbon-24 structures take on many arrangements in fractional coordinates, but only consist of one atom type. Stability analysis is not applicable to Perov-5 and Carbon-24, but proxy metrics introduced by Xie et al. (2021) are applicable to all datasets.

Standard (proxy) metrics for CSP and DNG Although computing the stability for generated materials is ideal, it is extremely expensive and technically challenging. In light of these difficulties, a number of *proxy metrics* have been developed by Xie et al. (2021). The primary advantage of these metrics is their low cost. We benchmark FlowMM and alternatives using specialized metrics for CSP and DNG.

In CSP we compute the *match rate* and the *Root-Mean-Square Error (RMSE)*. They measure the percentage of reconstructions from $q(\boldsymbol{f}, \boldsymbol{l}; \boldsymbol{a})$ that are satisfactorily close to the ground truth structure and the RMSE between coordinates, respectively. In DNG, we compute a *Structural & Compositional Validity* percentage using heuristics about interatomic distances and charge, respectively. We also compute *Coverage Recall & Precision* on chemical fingerprints and the *Wasserstein distance* between ground truth and generated material properties, namely atomic density ρ and number of unique elements per unit cell N_{el}. Note $N_{el} = N$-ary. See Appendix A for more details.

Stability metrics for DNG The ultimate goal of materials discovery is to propose stable, unique, and novel materials efficiently w.r.t. compute. For flow matching and diffusion models, the most expensive inference-time cost is integration steps. We therefore define several metrics to address these factors on a budget of 10,000 generations.

We compute the percent of generated materials that are stable (*Stability Rate*), but that does not address novelty. Following Zeni et al. (2023), we additionally compute the percentage of *stable, unique, and novel (S.U.N.)* materials (*S.U.N. Rate*). To address cost, we compute the average number of integration steps needed to generate a stable material (*Cost*) and a S.U.N. material (*S.U.N.* Cost). We explain identification of S.U.N. materials in Appendix A.

3. Riemannian Flow Matching for Materials

Our goal is to define a parametric generative model on the Riemannian manifold $\mathcal{C}$ that carries the geometry and invarinces inherent to crystals. We plan to accommodate both CSP and DNG with simple changes to our model and base distribution.

Concretely, we have a set of samples $\boldsymbol{a}_1, \boldsymbol{f}_1, \boldsymbol{l}_1 \sim q(\boldsymbol{a}, \boldsymbol{f}, \boldsymbol{l})$ where $q \in \mathcal{P}$ is an unknown probability distribution over $\mathcal{C}$, and we want to implicitly estimate the probability path $p_t : [0,1] \times \mathcal{P} \to \mathcal{P}$ that transforms our chosen base distribution $p_0 = p \in \mathcal{P}$ to $p_1 = q$. We achieve this by learning a parametric time-dependent vector field v_t^θ with parameters θ that optimizes (5), adapted to $\mathcal{C}$, on samples $\boldsymbol{a}_1, \boldsymbol{f}_1, \boldsymbol{l}_1$. Additionally, we enforce that the unconditional probability path be invariant to symmetries $g \in (\sigma, Q, \tau)$ at all times t:

$$p_t(g \cdot \boldsymbol{c}) \coloneqq \int_{\mathcal{C}} p_t(g \cdot \boldsymbol{c} \mid \boldsymbol{c}_1)\, q(\boldsymbol{c}_1)\, d\boldsymbol{c}_1 = p_t(\boldsymbol{c}). \quad (8)$$

We explain the necessary building blocks of our method (a) the geometry of $\mathcal{C}$, (b) the base distribution p and its invariances, (c) the conditional vector fields and our objective derived from (5), and finally (d) how our construction affirms that the marginal probability path $p_t(\boldsymbol{c})$ generated by $u_t(\boldsymbol{c})$ has the invariance properties of q for all $t \in [0,1]$.

Geometry of $\mathcal{C}$ Recall $\mathcal{C} \coloneqq \mathcal{A} \times \mathcal{F} \times \mathcal{L}$ forms a product manifold, implying that the inner product $\langle \boldsymbol{c}, \boldsymbol{c}' \rangle$ decomposes with addition, see Appendix A. We now consider the geometry of each component of $\mathcal{C}$ individually.

$\mathcal{F}$ is a permutation invariant collection of $n \times 3$-dimensional flat tori. That means it carries the Euclidean inner product locally, but each side is identified with its opposite. This is relevant for the geodesic and explains why paths "wrap around" the domain's edges. The identification is implemented by the action of the translation operator defined in Section 2.

The space of lattice parameters subject to the Niggli reduction $\mathcal{L} \coloneqq \mathbb{R}^{+3} \times [60, 120]^3$ is Euclidean, but has boundaries. We can safely ignore these boundaries for the lengths in $\mathbb{R}^{+3}$ since (i) the data does not lie on the boundary ($a, b, c > 0$) and (ii) we select a positive base distribution. Meanwhile, the angles α, β, γ do often lie directly on the boundary, posing a problem as the target u_t is not a *smooth* vector field. We address the issue with a diffeomorphism $\varphi\colon [60, 120] \to \mathbb{R}$ to *unconstrained space*, applied element wise to α, β, γ:

$$\operatorname{logit}(\xi) \coloneqq \log \frac{\xi}{1-\xi}, \qquad \varphi(\eta) \coloneqq \operatorname{logit}\left(\frac{\eta - 60}{120}\right),$$
$$\mathfrak{S}(\xi') = \frac{\exp(\xi')}{1+\exp(\xi')}, \quad \varphi^{-1}(\eta') = 120\,\mathfrak{S}(\eta') + 60,$$

where $\mathfrak{S}$ is the sigmoid function. Practically, geodesics and conditional vector field u_t are both represented in unconstrained space. Base samples and (estimated) target samples are transformed into unconstrained space for learning and integration then evaluated in $[60, 120]$, transformed with φ^{-1}.

The details of $\mathcal{A}$ depend on whether our task is CSP, where we estimate $q(\boldsymbol{f}, \boldsymbol{l}; \boldsymbol{a})$ or DNG, where we estimate $q(\boldsymbol{a}, \boldsymbol{f}, \boldsymbol{l})$. In CSP, $\boldsymbol{a}$ is given and the geometry is simple: $\mathcal{A}$ is a h-dimensional one-hot vector. Components of its unconditional vector field $u_t^{\mathcal{A}}(\boldsymbol{a}) = v_t^{\mathcal{A},\theta}(\boldsymbol{a}) = 0$ everywhere. Further discussion about $\mathcal{A}$ for CSP is unnecessary in this section and thus omitted! When doing DNG, we take $\mathcal{A}$ to be a $\lceil \log_2 h \rceil$-dimensional Euclidean space with a flat metric. Here, $\mathcal{A}$ has a simple geometry but the interesting part is that it represents atomic types more efficiently than a one-hot vector, in terms of dimension, after discretization with sign.

Base distribution on $\mathcal{C}$ Our base distribution on $\mathcal{C}$ is a product of distributions on $\mathcal{A}$, $\mathcal{F}$, and $\mathcal{L}$, see Appendix A. The base distribution $p(\boldsymbol{a})$ takes the same base distribution as Chen et al. (2022), namely $p(\boldsymbol{a}) = \mathcal{N}(\boldsymbol{a}; 0, 1)$ where $\mathcal{N}$ denotes the normal distribution. Next, we choose the distribution over $\mathcal{F}$ to be $p(\boldsymbol{f}) \coloneqq \text{Uniform}(0, 1)$, which covers the torus with equal density everywhere. Finally, we decided to leverage the flexibility afforded by Flow Matching in choosing an *informative* base distribution on $\boldsymbol{l}$. Recall, $\boldsymbol{l}$ can be split into three length and three angle parameters. Since length parameters are all positive we set $p(a, b, c) \coloneqq \prod_{\eta \in \{\alpha,\beta,\gamma\}} \text{LogNormal}(\eta; \text{loc}_\eta, \text{scale}_\eta)$ where $\text{loc}_\eta, \text{scale}_\eta$ are fit to training data using maximum likelihood. In the constrained space $[60, 120]$, angle parameters α, β, γ get base distribution $p(\alpha, \beta, \gamma) \coloneqq \text{Uniform}(60, 120)$. Samples can be drawn in unconstrained space by applying φ and the density can be computed using the change of variables formula.

The base distributions $p(\boldsymbol{a})$ and $p(\boldsymbol{f})$ are factorized and have no dependency on index. Therefore, they are permutation invariant. Next, $p(\boldsymbol{f})$ is translation invariant since,

$$\begin{aligned} p(\tau \cdot \boldsymbol{f}) &= \prod_{i=1,\ldots,n} U(f^i + \tau - \lfloor f^i + \tau \rfloor; 0, 1) \\ &= \prod_{i=1,\ldots,n} U(f^i; 0, 1) = p(\boldsymbol{f}), \end{aligned} \quad (9)$$

for all translations $\tau \in [-\frac{1}{2}, \frac{1}{2}]^3$. Our base distribution is $p(\boldsymbol{a}, \boldsymbol{f}, \boldsymbol{l}) \coloneqq p(\boldsymbol{a})p(\boldsymbol{f})p(\boldsymbol{l})$, so it carries these invariances. It remains to be shown that ψ_t is equivariant to these groups.

Conditional vector fields on $\mathcal{C}$ Recall from Chen & Lipman (2024), the conditional vector field on flat $\mathcal{M}$ is

$$u_t(m \mid m_1) = \frac{d \log \kappa(t)}{dt} d(m, m_1) \frac{\nabla_m d(m, m_1)}{\|\nabla_m d(m, m_1)\|^2} \quad (10)$$

where $d : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ is the geodesic distance (4) and $\kappa(t) = 1 - t$ is a linear time scheduler. Both $\mathcal{A}$ for DNG and (transformed) $\mathcal{L}$ are Euclidean manifolds with standard norm, recovering the Flow Matching conditional vector field on their respective tangent bundles, which we denote $u_t^{\mathcal{M}'}(m \mid m_1) = \frac{m_1 - m}{1-t}$ for $\mathcal{M}' \in \{\mathcal{A}, \mathcal{L}\}$.

We construct the conditional vector field for a point cloud living on a $n \times 3$-dimensional flat torus invariant to global

translations. First, we construct the naive geodesic path, which may cross the periodic boundary:

$$\exp_{f^i}(\dot{f}^i) := f^i + \dot{f}^i - \lfloor f^i + \dot{f}^i \rfloor, \tag{11}$$

$$\log_{f_0^i}(f_1^i) := \frac{1}{2\pi} \operatorname{atan2}\left[\sin(\omega^i), \cos(\omega^i)\right], \tag{12}$$

$$\omega^i := 2\pi(f_1^i - f_0^i), \tag{13}$$

where $\dot{f}^i \in \mathcal{T}_{f^i}\mathcal{F}^i$ for $i = 1, \ldots, n$. These amount to an atom wise application of $\log_{\boldsymbol{f}_0}$ on $\boldsymbol{f}_1$ and $\exp_{\boldsymbol{f}}$ on $\dot{\boldsymbol{f}} \in \mathcal{T}_{\boldsymbol{f}}\mathcal{M}$ respectively. Specifically, $d(\boldsymbol{f}, \boldsymbol{f}_1) := \|\log_{\boldsymbol{f}_1}(\boldsymbol{f})\|^2$ and $\nabla_{\boldsymbol{f}} d(\boldsymbol{f}, \boldsymbol{f}_1) = -2\log_{\boldsymbol{f}_1}(\boldsymbol{f})$. That would imply a target conditional vector field of $\frac{-\log_{\boldsymbol{f}_1}(\boldsymbol{f})}{1-t}$: a function which is equivariant–*not invariant*–to translation τ! We address this by removing the average torus translation from $\boldsymbol{f}_1$ to $\boldsymbol{f}$:

$$u_t^{\mathcal{F}}(\boldsymbol{f} \mid \boldsymbol{f}_1) := \log_{\boldsymbol{f}_1}(\boldsymbol{f}) - \frac{1}{n}\sum_{i=1}^{n} \log_{f_1^i}(f^i). \tag{14}$$

Our approach is similar to subtracting the mean of a point cloud in Euclidean space; however, it occurs in the tangent space instead of on the manifold. Given the factorization of the inner product on $\mathcal{C}$ (Appendix A), our objective is:

$$\begin{aligned}
&\mathbb{E}_{t, q(\boldsymbol{a}_1, \boldsymbol{f}_1, \boldsymbol{l}_1), p(\boldsymbol{a}_0), p(\boldsymbol{f}_0), p(\boldsymbol{l}_0)} \Bigg[\frac{\lambda_{\boldsymbol{a}}}{hn} \left\| v_t^{\mathcal{A},\theta}(\boldsymbol{c}_t) + \boldsymbol{a}_0 - \boldsymbol{a}_1 \right\|^2 \\
&+ \frac{\lambda_{\boldsymbol{f}}}{3n} \left\| v_t^{\mathcal{F},\theta}(\boldsymbol{c}_t) + \log_{\boldsymbol{f}_1}(\boldsymbol{f}_0) - \frac{1}{n}\sum_{i=1}^{n} \log_{f_1^i}(f_0^i) \right\|^2 \\
&+ \frac{\lambda_{\boldsymbol{l}}}{6} \left\| v_t^{\mathcal{L},\theta}(\boldsymbol{c}_t) + \boldsymbol{l}_0 - \boldsymbol{l}_1 \right\|^2 \Bigg],
\end{aligned} \tag{15}$$

where we've normalized by dimension and $\lambda_{\boldsymbol{a}}, \lambda_{\boldsymbol{f}}, \lambda_{\boldsymbol{l}} \in \mathbb{R}^+$ are hyperparameters and $t \sim \text{Uniform}(0, 1)$. In practice, since we have a closed form geodesic for all of our manifolds, our supervision signals are computed by evaluating conditional flow $\psi_t(\boldsymbol{c} \mid \boldsymbol{c}_1)$ on a minibatch determined by (4) at time t and taking the gradient with automatic differentiation, and in the component on $\mathcal{F}$ we subtract the mean.

Symmetries of the marginal path We show the symmetries of the conditional probability paths and construct the marginal path. Conditional probability path $p_t(\boldsymbol{c} \mid \boldsymbol{c}_1)$ mapping $p_0 = p(\boldsymbol{c})$ to $p_1(\boldsymbol{c} \mid \boldsymbol{c}_1)$ is generated by tuple $u_t(\boldsymbol{c} \mid \boldsymbol{c}_1) := \left(u_t^{\mathcal{A}}(\boldsymbol{a} \mid \boldsymbol{a}_1), u_t^{\mathcal{F}}(\boldsymbol{f} \mid \boldsymbol{f}_1), u_t^{\mathcal{L}}(\boldsymbol{l} \mid \boldsymbol{l}_1)\right)$ formed by direct sum. Conditional vector fields $u_t^{\mathcal{A}}$ and $u_t^{\mathcal{F}}$ are permutation equivariant through relabeling of particles and therefore $p_t(\boldsymbol{a} \mid \boldsymbol{a}_1)$ and $p_t(\boldsymbol{f} \mid \boldsymbol{f}_1)$ are invariant to permutation by Köhler et al. (2020). The representation of $\boldsymbol{l}$ makes $p_t(\boldsymbol{l} \mid \boldsymbol{l}_1)$ invariant to rotation. Finally, by translating away the mean tangent fractional coordinate we relaxed the traditional requirement in Flow Matching that conditional path $p_1(\boldsymbol{f} \mid \boldsymbol{f}_1) = \delta(\boldsymbol{f} - \boldsymbol{f}_1)$ and instead allow p_1 to concentrate on an equivalence class of $\boldsymbol{f}_1$ where all members in the same class are related by a translation. Therefore, $p_1(\boldsymbol{f} \mid \boldsymbol{f}_1)$ remains translation equivariant but the marginal probability path ends up translation invariant (Theorem D.2). Our unconditional vector field u_t, generating unconditional probability path p_t, connecting $p_0 = p$ to $p_1 = q$ is:

$$u_t(\boldsymbol{c}) := \int_{\mathcal{C}} u_t(\boldsymbol{c} \mid \boldsymbol{c}_1) \frac{p_t(\boldsymbol{c} \mid \boldsymbol{c}_1) q(\boldsymbol{c}_1)}{p_t(\boldsymbol{c})} \, d\boldsymbol{c}_1 \tag{16}$$

$$p_t(\boldsymbol{c}) := \int_{\mathcal{C}} p_t(\boldsymbol{c} \mid \boldsymbol{c}_1) q(\boldsymbol{c}_1) \, d\boldsymbol{c}_1. \tag{17}$$

Our construction enforces that $p_t(\boldsymbol{c})$ is invariant to permutation, translation, and rotation, with proof in Appendix D.

Estimated marginal path We specify our model, the unconditional probability path $p_t^\theta(\boldsymbol{c})$, generated by $v_t^\theta(\boldsymbol{c})$,

$$p_t^\theta(\boldsymbol{c}) := \int_{\mathcal{C}} p_t^\theta(\boldsymbol{c} \mid \boldsymbol{c}_1) q(\boldsymbol{c}_1) d\boldsymbol{c}_1, \tag{18}$$

trained by optimizing (15), and when $p_0 = p$ then $p_1 \approx q$. We let $v_t^\theta(\boldsymbol{c})$ be a graph neural network in the style of (Satorras et al., 2021; Jiao et al., 2023) that enforces equivariance to permutation for $v_t^{\mathcal{A},\mathcal{F},\theta}(\boldsymbol{a}, \boldsymbol{f})$ via message passing and invariance to translation in $v_t^{\mathcal{F},\theta}(\boldsymbol{f})$ by featurizing graph edges as displacements between atoms. Invariance to rotation of $v_t^{\mathcal{L},\theta}(\boldsymbol{l})$ is enforced by the representation of $\boldsymbol{l}$. After enforcing these symmetries in our network, we know that $p_t^\theta(\boldsymbol{c})$ has the invariances desired by design. For more details about the graph neural network, see Appendix C.

Inference Anti-Annealing A numerical trick which increased the performance of our neural network on the proxy metrics was to adjust the predicted velocity $v_t^\theta(\boldsymbol{c})$ at inference time. We write the ordinary differential equation and initial condition $\boldsymbol{c} \sim p(\boldsymbol{c})$ that defines the flow,

$$\frac{d}{dt}\psi_t^\theta = s(t) v_t^\theta(\psi_t^\theta(\boldsymbol{c})), \qquad \psi_0^\theta(\boldsymbol{c}) = \boldsymbol{c}, \tag{19}$$

but include a time-dependent velocity scaling term $s(t) := 1 + s't$ where s' is a hyperparameter. We typically found best performance when $0 \leq s' \leq 10$. Notably, we also found that selectively applying the velocity increase to particular variables had a significant effect. In CSP, it was helpful for fractional coordinates but hurtful for lattice parameters $\boldsymbol{l}$. We increased the fractional coordinate velocity for our reported results. For DNG, the trend was not as simple. Further investigation of this effect through ablation study can be found in Appendix B. This effect has been observed in multiple other studies (Yim et al., 2023; Bose et al., 2023).

4. Experiments

We evaluate FlowMM on the two tasks we set out at the beginning of the paper: Crystal Structure Prediction and

De Novo Generation. We apply proxy metrics in all experiments, with a focus on inference-stage efficiency. We additionally investigate the stability of DNG samples by performing extensive density functional theory calculations.

4.1. Crystal Structure Prediction

We perform CSP on all datasets (Perov-5, Carbon-24, MP-20, and MPTS-52) with CDVAE, DiffCSP and FlowMM, evaluating them with proxy metrics computed using `StructureMatcher` (Ong et al., 2013). We present the Match Rate and the Root-Mean-Square Error (RMSE) in Table 1. DiffCSP and FlowMM use exactly the same underlying neural network in an apples-to-apples comparison. FlowMM outperforms competing models on the more challenging & realistic datasets (MP-20 and MPTS-52) on both metrics by a considerable margin. Figure 2 investigates sampling efficiency by comparing the match rate of DiffCSP and FlowMM as a function of number of integration steps. FlowMM achieves a higher match rate with far fewer integration steps, which corresponds to more efficient inference. FlowMM achieves maximum match rate in about 50 steps, at least an order of magnitude decrease in inference time cost compared to the 1000 steps used by DiffCSP. We ablate both inference anti-annealing and the proposed base distribution $p(\boldsymbol{l})$ and confirm that FlowMM is competitive or outperforms other models in terms of Match Rate without those enhancements. We additionally report inference-time uncertainty. Those results are located in Appendix B.

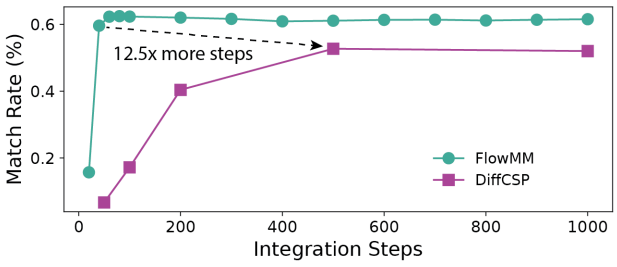


Figure 2. Match rate as a function of number of integration steps on MP-20. FlowMM achieves a higher maximum match rate than DiffCSP overall, and does so ~ 450 steps before DiffCSP. Results with Inference Anti-Annealing ablated are in Appendix B.

4.2. De novo generation

To evaluate de novo generation we trained models on the MP-20 dataset and we generated 10,000 structures from CDVAE and DiffCSP. For FlowMM, we generated 40,000 structures, in batches of 10,000, using a variable number of integration steps: $\{250, 500, 750, 1000\}$. Table 2 shows the proxy metrics along with the stability metrics computed using the generated structures. FlowMM is competitive with the diffusion models on most metrics, but significantly outperforms them on several Wasserstein distance metrics

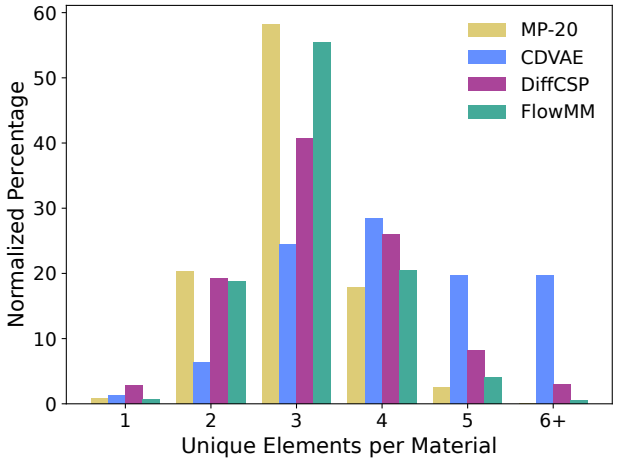


Figure 3. The distribution of number of unique elements per material, or N-ary, for the MP-20 distribution and the generative models. FlowMM matches the MP-20 distribution closest, while CDVAE and DiffCSP generate too many materials with N-ary ≥ 5.

between distributions of properties of generated structures and the test set, specifically: the atomic density ρ and the number of unique elements per crystal N_{el} (same as N-ary).

Table 2 also shows two different stability metrics based on energy above hull (E^{hull}) calculations. To compute E^{hull} for the experiments, we ran structure relaxations with CHGNet (Deng et al., 2023) and density functional theory and used those to determine the distance to the convex hull and thereby stability. Further details are in Appendix A. Conventional methods (Glass et al., 2006; Pickard & Needs, 2011) involve random search and hundreds of expensive density functional theory evaluations. We aim to reduce the computational expense of De Novo Generation. Therefore, S.U.N. Cost is our most important metric as it indicate the expense of finding a new material in terms of integration steps at inference time. From Table 2, it is clear that FlowMM is competitive to DiffCSP on the S.U.N. Rate and Stablity Rate metrics, but significantly better on Cost and S.U.N. Cost due to the reduction in integration steps. This efficiency is typical of flow matching compared to diffusion (Shaul et al., 2023; Yim et al., 2023; Bose et al., 2023). We note the caveat that integration steps are not the only source of computational cost. Training, prerelaxation, and relaxation are all costs worth considering; however, they are slightly more difficult to benchmark. Furthermore, we found them to be approximately equal across models so we focus on the cost of inference, which varies considerably.

We also compare the distribution of the computed E^{hull} values for the various methods in Figure 4. Structures generated by FlowMM are on average much more stable than CDVAE, and are comparable to those generated by DiffCSP.

We compare the distribution of materials according to the

Table 1. Results from crystal structure prediction on unit tests and realistic data sets.

	Perov-5		Carbon-24		MP-20		MPTS-52	
	Match Rate (%) ↑	RMSE ↓	Match Rate (%) ↑	RMSE ↓	Match Rate (%) ↑	RMSE ↓	Match Rate (%) ↑	RMSE ↓
CDVAE	45.31	0.1138	17.09	0.2969	33.90	0.1045	5.34	0.2106
DiffCSP	52.02	**0.0760**	17.54	**0.2759**	51.49	0.0631	12.19	0.1786
FlowMM	**53.15**	0.0992	**23.47**	0.4122	**61.39**	**0.0566**	**17.54**	**0.1726**

Table 2. Results from De Novo generation on the MP-20 dataset.

Method	Integration Steps	Validity (%) ↑		Coverage (%) ↑		Property ↓		Stability Rate† (%) ↑	Cost ↓	S.U.N. Rate ↑	S.U.N. Cost ↓
		Structural	Composition	Recall	Precision	wdist (ρ)	wdist (N_{el})	MP-2023	Steps/Stable†	MP-2023	Steps/S.U.N.
CDVAE	5000	**100.00**	**86.70**	99.15	99.49	0.688	0.278	1.57	31.85	1.43	34.97
DiffCSP	1000	**100.00**	83.25	**99.71**	**99.76**	0.350	0.125	**5.06**	1.98	**3.34**	2.99
FlowMM	250	96.58	83.47	99.48	99.65	**0.261**	**0.107**	4.32	**0.58**	2.38	**1.05**
	500	96.86	83.24	99.38	99.63	**0.075**	**0.079**	4.19	**1.19**	2.45	**2.04**
	750	96.78	83.08	99.64	99.63	**0.281**	**0.097**	4.14	**1.81**	2.22	3.38
	1000	96.85	83.19	99.49	99.58	**0.239**	**0.083**	4.65	2.15	2.34	4.27

Stable† implies $E^{hull} < 0.0$ & N-ary ≥ 2.

arity of the structure. Figure 3 compares the N-ary distribution of each of the models to the MP-20 dataset. FlowMM matches the data distribution significantly better than the diffusion models, this is confirmed numerically with the Wasserstein distance metric N_{el} in table 2. We present a similar distribution for stable structures in Figure 9b.

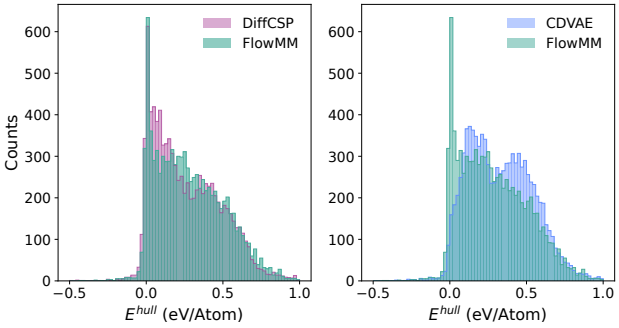


Figure 4. Histogram comparing the distribution of E^{hull} computed after relaxation with DFT for generative models DiffCSP and CDVAE with our proposed FlowMM on the DNG task. After relaxation on for all models, FlowMM generates lower energy structures compared to CDVAE and is competitive with DiffCSP.

5. Conclusion

We introduced a novel method for training continuous normalizing flows using a generalization of Riemannian Flow Matching for generating periodic crystal structures. We empirically tested our model using both CSP and DNG tasks and found strong performance on all proxy-metrics. In CSP, we used exactly the same network as DiffCSP, thereby performing an apples-to-apples comparison. For MP-20, FlowMM was able to outperform DiffCSP, in terms of Match Rate, with as few as 50 integration steps. This represents at least an order of magnitude improvement.

We rigorously evaluated the DNG structures for stability using the energy above hull to determine the Stability Rate, Cost, S.U.N. Rate, and S.U.N. Costs for each model. We found that FlowMM significantly outperforms both CDVAE and DiffCSP on Cost and S.U.N. Cost, and is competitive with DiffCSP on Stability Rate and S.U.N. Rate. This is enabled by FlowMM's 3x more efficient generation, in terms of integration steps, at inference time. The inference time efficiency can be explained by the kinetically optimal paths learned using the Flow Matching objective (Shaul et al., 2023). Resource limitations meant we did not investigate whether FlowMM could generate a similar number of stable structures using only a handful of integration steps. Based on the extremely efficient CSP results in Figure 2, this would be an interesting direction for future work.

Impact Statement

Our paper presents a generative model for predicting the composition and structure of stable materials. Our work may aid in the discovery of novel materials that could catalyze chemical reactions, enable higher energy density battery technology, and advance other areas of materials science and chemistry. The downstream effects are difficult to judge, but the challenges associated with taking a computational prediction to synthesized structure imply the societal impacts are likely going to be limited to new research directions.

Acknowledgements

The authors thank C. Lawrence Zitnick, Kyle Michel, Vahe Gharakhanyan, Abhishek Das, Luis Barroso-Luque, Janice Lan, Muhammed Shuaibi, Brook Wander, and Zachary Ulissi for helpful discussions. Meta provided the compute.

References

Adams, R. P. and Orbanz, P. Representing and learning functions invariant under crystallographic groups. *arXiv*

preprint arXiv:2306.05261, 2023.

AI4Science, M., Hernandez-Garcia, A., Duval, A., Volokhova, A., Bengio, Y., Sharma, D., Carrier, P. L., Koziarski, M., and Schmidt, V. Crystal-gfn: sampling crystals with desirable properties and constraints. *arXiv preprint arXiv:2310.04925*, 2023.

Appl, M. The haber–bosch process and the development of chemical engineering. a century of chemical engineering, 1982.

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and Van Den Berg, R. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

Baird, S. G., Sayeed, H. M., and Riebesell, J. sparks-baird/matbench-genmetrics. https://github.com/sparks-baird/matbench-genmetrics, 2024. [Accessed 03-05-2024].

Bose, A. J., Akhound-Sadegh, T., Fatras, K., Huguet, G., Rector-Brooks, J., Liu, C.-H., Nica, A. C., Korablyov, M., Bronstein, M., and Tong, A. Se (3)-stochastic flow matching for protein backbone generation. *arXiv preprint arXiv:2310.02391*, 2023.

Cao, Z., Luo, X., Lv, J., and Wang, L. Space group informed transformer for crystalline materials generation, 2024.

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

Chavel, I., Randol, B., and Dodziuk, J. (eds.). *Eigenvalues in Riemannian Geometry*, volume 115 of *Pure and Applied Mathematics*. Elsevier, 1984. doi: https://doi.org/10.1016/S0079-8169(08)60810-7. URL https://www.sciencedirect.com/science/article/pii/S0079816908608107.

Cheetham, A. K. and Seshadri, R. Artificial intelligence driving materials discovery? perspective on the article: Scaling deep learning for materials discovery. *Chemistry of Materials*, 2024.

Chen, R. T. and Lipman, Y. Riemannian flow matching on general geometries. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=g7ohDlTITL.

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

Chen, T., Zhang, R., and Hinton, G. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *The Eleventh International Conference on Learning Representations*, 2022.

Choubisa, H., Todorović, P., Pina, J. M., Parmar, D. H., Li, Z., Voznyy, O., Tamblyn, I., and Sargent, E. H. Interpretable discovery of semiconductors with machine learning. *npj Computational Materials*, 9(1):117, 2023.

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

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

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

Falorsi, L. and Forré, P. Neural ordinary differential equations on manifolds. *arXiv preprint arXiv:2006.06663*, 2020.

Flam-Shepherd, D. and Aspuru-Guzik, A. Language models can generate molecules, materials, and protein binding sites directly in three dimensions as xyz, cif, and pdb files. *arXiv preprint arXiv:2305.05708*, 2023.

Geiger, M. and Smidt, T. e3nn: Euclidean neural networks. *arXiv preprint arXiv:2207.09453*, 2022.

Gemici, M. C., Rezende, D., and Mohamed, S. Normalizing flows on riemannian manifolds. *arXiv preprint arXiv:1611.02304*, 2016.

Glass, C. W., Oganov, A. R., and Hansen, N. Uspex—evolutionary crystal structure prediction. *Computer physics communications*, 175(11-12):713–720, 2006.

Grosse-Kunstleve, R. W., Sauter, N. K., and Adams, P. D. Numerically stable algorithms for the computation of reduced unit cells. *Acta Crystallographica Section A: Foundations of Crystallography*, 60(1):1–6, 2004.

Gruver, N., Sriram, A., Madotto, A., Wilson, A. G., Zitnick, C. L., and Ulissi, Z. Fine-tuned language models generate stable inorganic materials as text. *arXiv preprint arXiv:2402.04379*, 2024.

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

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

Hu, E., Liu, C., Zhang, W., and Yan, Q. Machine learning assisted understanding and discovery of co2 reduction reaction electrocatalyst. *The Journal of Physical Chemistry C*, 127(2):882–893, 2023.

Huang, C.-W., Aghajohari, M., Bose, J., Panangaden, P., and Courville, A. Riemannian diffusion models. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022.

Jain, A., Ong, S. P., Hautier, G., Chen, W., Richards, W. D., Dacek, S., Cholia, S., Gunter, D., Skinner, D., Ceder, G., and Persson, K. A. The Materials Project: A materials genome approach to accelerating materials innovation. *APL Materials*, 1(1):011002, 07 2013. ISSN 2166-532X. doi: 10.1063/1.4812323. URL https://doi.org/10.1063/1.4812323.

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

Jiao, R., Huang, W., Liu, Y., Zhao, D., and Liu, Y. Space group constrained crystal generation, 2024.

Köhler, J., Klein, L., and Noé, F. Equivariant flows: exact likelihood generative learning for symmetric densities. In *International conference on machine learning*, pp. 5361–5370. PMLR, 2020.

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

Kondor, R. and Trivedi, S. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *International Conference on Machine Learning*, pp. 2747–2755. PMLR, 2018.

Kresse, G. and Furthmüller, J. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Physical review B*, 54(16):11169, 1996.

Liao, Y.-L., Wood, B., Das, A., and Smidt, T. Equiformerv2: Improved equivariant transformer for scaling to higher-degree representations. *arXiv preprint arXiv:2306.12059*, 2023.

Ling, C. A review of the recent progress in battery informatics. *npj Computational Materials*, 8(1):33, 2022.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2022.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

Mathieu, E. and Nickel, M. Riemannian continuous normalizing flows. *Advances in Neural Information Processing Systems*, 33:2503–2515, 2020.

Merchant, A., Batzner, S., Schoenholz, S. S., Aykol, M., Cheon, G., and Cubuk, E. D. Scaling deep learning for materials discovery. *Nature*, pp. 1–6, 2023.

Miller, B. K., Geiger, M., Smidt, T. E., and Noé, F. Relevance of rotationally equivariant convolutions for predicting molecular properties. *arXiv preprint arXiv:2008.08461*, 2020.

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

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

Passaro, S. and Zitnick, C. L. Reducing so (3) convolutions to so (2) for efficient equivariant gnns. *arXiv preprint arXiv:2302.03655*, 2023.

Perdew, J. P., Burke, K., and Ernzerhof, M. Generalized gradient approximation made simple. *Physical review letters*, 77(18):3865, 1996.

Pickard, C. J. Airss data for carbon at 10gpa and the c+n+h+o system at 1gpa. *Materials Cloud Archive*, 2020.0026/v1, 2020. doi: 10.24435/materialscloud:2020.0026/v1.

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

Potyrailo, R., Rajan, K., Stoewe, K., Takeuchi, I., Chisholm, B., and Lam, H. Combinatorial and high-throughput screening of materials libraries: review of state of the art. *ACS combinatorial science*, 13(6):579–633, 2011.

Riebesell, J. Matbench Discovery v1.0.0. *figshare*, 1 2024. doi: 10.6084/m9.figshare.22715158.v12. URL https://figshare.com/articles/dataset/Matbench_Discovery_v1_0_0/22715158.

Riebesell, J., Goodall, R., Benner, P., Chiang, Y., Lee, A., Jain, A., and Persson, K. Matbench Discovery, August 2023a. URL https://github.com/janosh/matbench-discovery.

Riebesell, J., Goodall, R. E., Jain, A., Benner, P., Persson, K. A., and Lee, A. A. Matbench discovery an evaluation framework for machine learning crystal stability prediction. *arXiv preprint arXiv:2308.14920*, 2023b.

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

Schmidt, J., Hoffmann, N., Wang, H.-C., Borlido, P., Carriço, P. J., Cerqueira, T. F., Botti, S., and Marques, M. A. Large-scale machine-learning-assisted exploration of the whole materials space. *arXiv preprint arXiv:2210.00579*, 2022.

Shaul, N., Chen, R. T., Nickel, M., Le, M., and Lipman, Y. On kinetic optimal probability paths for generative models. In *International Conference on Machine Learning*, pp. 30883–30907. PMLR, 2023.

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

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

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

Wang, H.-C., Botti, S., and Marques, M. A. Predicting stable crystalline compounds using chemical similarity. *npj Computational Materials*, 7(1):12, 2021.

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

Weiler, M., Forré, P., Verlinde, E., and Welling, M. Coordinate independent convolutional networks–isometry and gauge equivariant convolutions on riemannian manifolds. *arXiv preprint arXiv:2106.06020*, 2021.

Wirnsberger, P., Papamakarios, G., Ibarz, B., Racanière, S., Ballard, A. J., Pritzel, A., and Blundell, C. Normalizing flows for atomic solids. *Machine Learning: Science and Technology*, 3(2):025009, 2022.

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

Yang, M., Cho, K., Merchant, A., Abbeel, P., Schuurmans, D., Mordatch, I., and Cubuk, E. D. Scalable diffusion for materials generation. *arXiv preprint arXiv:2311.09235*, 2023.

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

Yim, J., Campbell, A., Foong, A. Y., Gastegger, M., Jiménez-Luna, J., Lewis, S., Satorras, V. G., Veeling, B. S., Barzilay, R., Jaakkola, T., et al. Fast protein backbone generation with se (3) flow matching. *arXiv preprint arXiv:2310.05297*, 2023.

Zeni, C., Pinsler, R., Zügner, D., Fowler, A., Horton, M., Fu, X., Shysheya, S., Crabbé, J., Sun, L., Smith, J., et al. Mattergen: a generative model for inorganic materials design. *arXiv preprint arXiv:2312.03687*, 2023.

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

A. Preliminaries Continued

A.1. Datasets

We consider two realistic datasets and two unit test datasets. The first two are realistic and the second two are unit tests. All datasets are divided into 60% training data, 20% validation data, 20% test data. We use the same splits as Xie et al. (2021) and Jiao et al. (2023).

Materials Project-20 (Xie et al., 2021) Also known as *MP-20*. Contains 45231 samples. Contains all materials with a maximum of 20 atoms per unit cell and within 0.08 eV/atom in the Materials Project database (Jain et al., 2013) from around July 2021. Materials containing radioactive atoms are removed.

Materials Project Time Splits-52 (Baird et al., 2024) Also known as *MPTS-52*. Contains 40476 samples. Uses similar criteria as MP-20, but allows materials with atoms up to 52 in a single unit cell and no elemental filtering is applied. Furthermore, the train, validation, and test splits are organized in chronological order. Therefore, the oldest materials are in the training set and the newest ones are in the test set. Note: this dataset has fewer samples than MP-20 because some materials are entered into the Materials Project database without first publication timestamp information. Those materials are omitted from the dataset.

Perovskite-5 (Castelli et al., 2012) Also known as *perov* or *perov-5*. Contains 18928 samples. All materials have five atoms per unit cell located at the same fractional coordinate values and lattice angles are fixed. Only the lattice lengths and atomic types change.

Carbon-24 (Pickard, 2020) Also known as *carbon*. Contains 10153 samples. Each material contains only carbon atoms, but the other variables are not fixed. This leads to a challenging CSP problem because there are typically multiple geometries for every n-atom set of carbon atoms. This is reflected in depressed match rate scores.

A.2. Proxy metrics

Crystal Structure Prediction Following Jiao et al. (2023), we sampled the CSP model using held out structures as conditioning and measured the *match rate* and *Root-Mean-Square Error (RMSE)*, according to the output of `StructureMatcher` Ong et al. (2013) with settings $stol = 0.5, angle_tol = 10, ltol = 0.3$. Match rate is the number of generated structures that `StructureMatcher` find are within tolerances defined above divided by the total number of held-out structures. RMSE is computed when the held-out and generated structures match (otherwise it does not enter the reported statistics), then normalized by $(V/N)^{1/3}$ as is standard. Unlike DiffCSP, we did not compute multi-sample statistics given the same input composition.

De novo generation A composition is structurally valid when all pairwise distance between atoms are greater than 0.5 Å. A crystal is compositionally valid when a simple heuristic system, SMACT (Davies et al., 2019), determines that the crystal would be charge neutral. Coverage for both COV-R (recall) and COV-P (precision) are standard Recall & Precision metrics computed after on thresholding pairwise distances between 1,000 samples that are both compositionally and structurally valid, and have been featurized by CrystalNN structural fingerprints (Zimmermann & Jain, 2020) and the normalized Magpie compositional fingerprints (Ward et al., 2016).

We also compute two Wasserstein distances on computed properties of crystal samples from the test set and our generated structures. Namely, d_ρ and $d_{N_{el}}$, corresponding to distances between the atomic density: number of atoms divided by unit cell volume and N_{el} which is the number of unique elements in the unit cell, aka N-ary.

A.3. Riemannian Manifolds

Since $\mathcal{C}$ is a product of Riemannian manifolds, it has a natural metric: For any $(\boldsymbol{a}, \boldsymbol{f}, \boldsymbol{l}) \in \mathcal{C}$, the tangent space $(\mathcal{A} \times \mathcal{F} \times \mathcal{L})_{(\boldsymbol{a},\boldsymbol{f},\boldsymbol{l})}$ is canonically isomorphic to the direct sum $\mathcal{A}_{\boldsymbol{a}} \oplus \mathcal{F}_{\boldsymbol{f}} \oplus \mathcal{L}_{\boldsymbol{l}}$. For vectors $\xi, \eta, \in \mathcal{A}_{\boldsymbol{a}}$; $\zeta, \chi \in \mathcal{F}_{\boldsymbol{f}}$; and $\gamma, \omega \in \mathcal{L}_{\boldsymbol{l}}$ we define the inner product of $\xi \oplus \zeta \oplus \gamma$ and $\eta \oplus \chi \oplus \omega$ by $\langle \xi \oplus \zeta \oplus \gamma, \eta \oplus \chi \oplus \omega \rangle_{(\boldsymbol{a},\boldsymbol{f},\boldsymbol{l})} := \lambda_a \langle \xi, \eta \rangle_{\boldsymbol{a}} + \lambda_f \langle \zeta, \xi \rangle_{\boldsymbol{f}} + \lambda_l \langle \gamma, \omega \rangle_{\boldsymbol{l}}$, where the subscripts indicate the tangent space in which the different inner products are calculated and $\lambda := (\lambda_a, \lambda_f, \lambda_l) \in \mathbb{R}^3$ is a hyperparameter. In particular, $\mathcal{A}_{\boldsymbol{a}} \oplus \{0\} \oplus \{0\}$ is orthogonal to $\{0\} \oplus \mathcal{F}_{\boldsymbol{f}} \oplus \{0\}$ and $\{0\} \oplus \{0\} \oplus \mathcal{L}_{\boldsymbol{l}}$. The associated Riemannian measure on $\mathcal{A} \times \mathcal{F} \times \mathcal{L}$ is the product measure determined by $dV_{\mathcal{A}}$, $dV_{\mathcal{F}}$, and $dV_{\mathcal{L}}$ where $dV_{\square}$ denotes the

Lebesgue measure on space $\square$ (Chavel et al., 1984). Since our measure is a product measure, we may define the base probability measures of each space by densities absolutely continuous with respect to their respective Lebesgue measure.

A.4. Specifics for De Novo Generation

Determining number of atoms in the unit cell Above, we describe De Novo Generation via a distribution $p(\boldsymbol{a}, \boldsymbol{f}, \boldsymbol{l})$; however, this omits an important variable: n the number of atoms in the unit cell. This distribution carries an implicit conditional on the number of atoms, namely $p(\boldsymbol{a}, \boldsymbol{f}, \boldsymbol{l} \mid n)$. In other words, we have assumed that n is known beforehand. However, we are interested in generating materials with a variable number of atoms. To solve this problem we follow the method of Hoogeboom et al. (2022) and first sample $n \sim p(n)$ from the empirical distribution of the training set.

Methodology for identifying Stable, Unique, and Novel (S.U.N.) materials Our goal in DNG is to generate stable, unique and novel materials. In that effort, we generated samples from FlowMM, prerelaxed them using CHGNet, and finally relaxed them using density functional theory. Our method for determining whether a material is S.U.N. is as follows:

(S) We determine the stability of our relaxed structures against the `Matbench Discovery` (Riebesell et al., 2023a;b) convex hull, compiled from the Materials Project (Jain et al., 2013), marked as 2nd Feburary 2023. (Although our training data comes from an earlier version of the database, we can still estimate the performance of the models using a later version of the convex hull. In this situation, it becomes more difficult to generate novel structures since those proposed structures may have been added to the database between 2021 and 2023.)

(N) We then take our stable generated structures and search the training data for any structure which contains the same set of elements. We ignore the frequency of the elements during this search, in order to catch similar materials with differently defined unit cells. We do a pairwise comparison between the generated structure and all "element-matching" examples from the training set using `StructureMatcher` (Ong et al., 2013) with default settings. If there is no match, we consider that structure novel.

(U) Finally, we take all stable and novel structures, then use `StructureMatcher` to pairwise compare those structures with themselves. We collect all pairwise matches and group them into "equivalent" structures. This group counts as only one structure for the purpose of S.U.N. computations, thereby enforcing uniqueness.

We want to emphasize that this is not a perfect system. `StructureMatcher` may fail to detect a match, or falsely detect one, due to the default settings of its threshold. Furthermore, without careful application beyond the default settings, `StructureMatcher` does not tell us about chemical function and may not yield matches for materials with extremely similar chemical properties. This could inflate the estimated number of S.U.N. materials (Cheetham & Seshadri, 2024). Additionally, `StructureMatcher` does not define an equivalence relation since it does satisfy the reflexivity property. We treat it like one here anyway since it holds approximately.

Stability metrics explained We are interested in several stability metrics:

$$Stability\ Rate := \frac{N_{\text{stable}}}{N_{\text{gen}}} \tag{20}$$

$$Cost := \frac{N_{\text{int. steps}}}{Stability\ Rate} \tag{21}$$

$$S.U.N.\ Rate := \frac{N_{\text{S.U.N.}}}{N_{\text{gen}}} \tag{22}$$

$$S.U.N.\ Cost := \frac{N_{\text{int. steps}}}{S.U.N.\ Rate} \tag{23}$$

where N_{gen} is the number of generated samples, N_{stable} is the number of generated samples which are stable; $N_{\text{S.U.N.}}$ are the number of generated samples which are stable, unique, and novel; and $N_{\text{int. steps}}$ is the number of integration steps to produce a generated sample. By definition $N_{\text{S.U.N.}} \leq N_{\text{stable}} \leq N_{\text{gen}}$.

A.5. Symmetry

We discuss invariances to symmetry groups for crystal structures. We are interested in estimating a density with invariances to permutation, translation, and rotation as formalized in (1), (2), and (3). We visualize those symmetries in Figure 5.

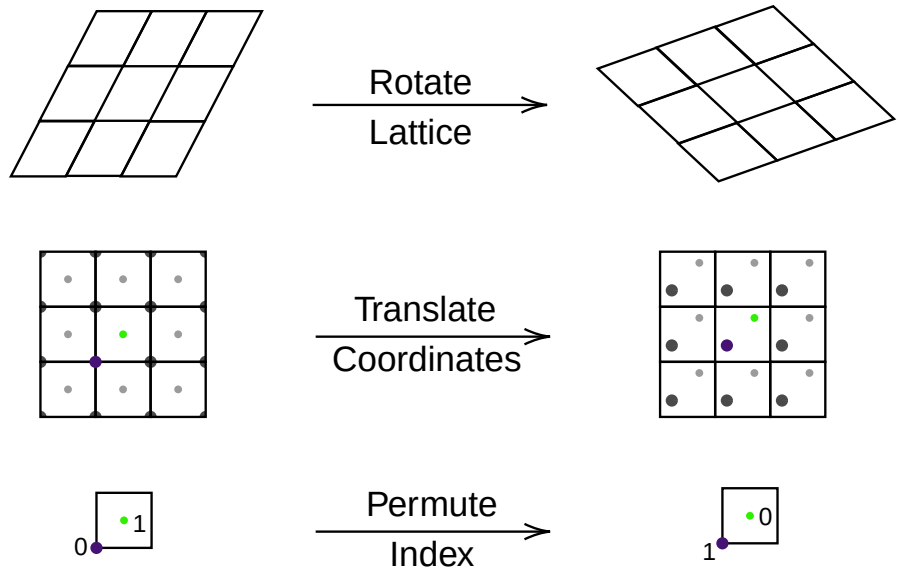


Figure 5. Three symmetry actions are shown above; all of these actions would alter only the representation of the crystal, while leaving its chemical properties intact. (top) Rotation of a lattice formed by a unit cell. (mid) Translation of fractional coordinates within a unit cell. (bot) Permutation of atomic index. Since these images are two-dimensional, they do not capture the symmetry of a three-dimensional crystal. Furthermore, there are additional symmetries that are not represented in these pictures.

There are additional symmetries for crystals that we did not explicitly model in FlowMM. Those are *periodic cell choice invariance*: where the unit cell is skewed by A with $\det A = 1$ and $A \in \mathbb{Z}^{3\times 3}$ and the fractional coordinates are anti-skewed by A^{-1}, and *supercell invariance* where the unit cell is grown to encompass another neighboring "block" and all of the atoms inside. These are discussed in more detail and visualized in Zeni et al. (2023).

A.6. Riemannian Flow Matching visualization

Since flow matching can be rather formal, and perhaps unintuitive when written symbolically, we draw cartoon representations of the regression target from the conditional vector field $u_t(\cdot \mid \cdot_1)$ for the fractional coordinates $\boldsymbol{f}$ and lattice parameters $\boldsymbol{l}$ in Figure 6 and Figure 7, respectively. In both cases, we also represent all necessary components to define the regression target namely the sample from the base distribution $\square_0$, the sample from the target distribution $\square_1$, the conditional path connecting them on the correct manifold, the point at time t along the path where the conditional vector field is evaluated $\square_t$, and the conditional vector itself $u_t(\square_t \mid \square_1)$ where $\square$ represents the relevant variable. We do not show $\boldsymbol{a}$ because it occurs in Euclidean space and behaves like typical flow matching during training.

A.7. Details about Density Functional Theory calculations

For the stability metrics, we applied the Vienna ab initio simulation package (VASP) (Kresse & Furthmüller, 1996) to compute relaxed geometries and ground state energies at a temperature of 0 K and pressure of 0 atm. We used the default settings from the Materials Project (Jain et al., 2013) known as the `MPRelaxSet` with the PBE functional (Perdew et al., 1996) and Hubbard U corrections. These correspond with the settings that our prerelaxation network CHGNet (Deng et al., 2023) was trained on, so prerelaxation should reduce DFT energy, up to the fitting error in CHGNet.

A.8. Limitations of quantifying a computational approach to materials discovery

There are a number of important limitations when it comes to using and quantifying the performance of generative models for materials discovery.

Fundamental limitations for all computational methods include, but are not limited to: (a) Energy and stability computations all occur at nonphysical zero temperature and pressure settings. (b) Our material representation is not realistic since it

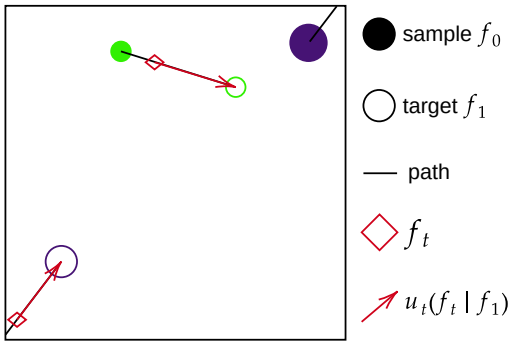


Figure 6. We visualize the necessary components to express a hypothetical Riemannian Flow Matching regression target on a single data point for the fractional coordinates. The sample f_0 is drawn from the base uniform distribution for both points. The target f_1 is from the database of crystals. The path is drawn between the sample and the target following the geodesic path, i.e. wrapping around the boundary. The point f_t along the path at time t is indicated with a diamond. The conditional vector $u_t(f_t \mid f_1)$ at time t is indicated as a vector. This vector is the regression target in Riemannian Flow Matching.

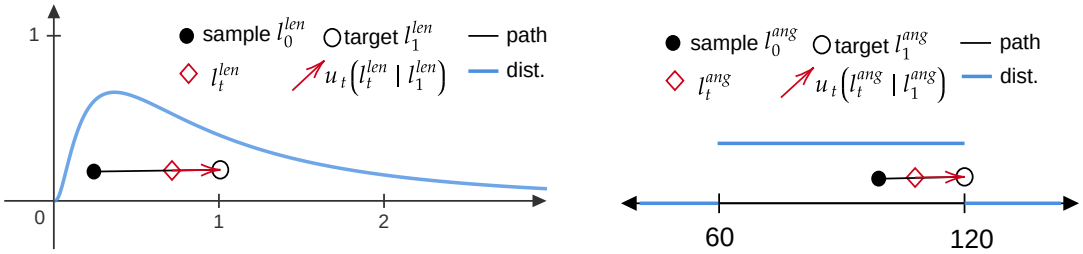


Figure 7. We visualize the necessary components to express a hypothetical Flow Matching regression target on a single data point for the lattice parameters. Specifically, the length parameters are shown on the left and the angle parameters are on the right. The sample, target, path, point-along-path, and vector all follow the description in Figure 6, but are adapted for lattice parameters. Additionally, we display the base distribution as a blue line on this plot to indicate where samples l_0^{len} and l_0^{ang} can appear. Note: This visualization occurs in so-called constrained space for l^{ang}; however, our proposed method does flow matching in the unconstrained space of l^{ang} to avoid the boundaries of the l^{ang} distribution. In this way, this figure visualises the challenges of doing precise flow matching in constrained space and the corresponding difficulty (and lower performance) of the ablated model. Recall that our transformation sends $60 \to -\infty$ and $120 \to \infty$.

assumes complete homogeneity and an infinite crystal structure without disorder. (c) There is a fundamental inaccuracy in density functional theory itself due to the basis set, the energy functional, and computational cost limitations... (etc.)

Generative models learn to fit empirical distributions. We are interested in generating S.U.N. materials which are not in our empirical distribution. In an imprecise way, we expect that FlowMM will generated materials that exist as "interpolations" between existing structures; however, the most interesting and new structures are well outside the existing empirical distribution. We do not expect FlowMM to find these interesting and new structures since it is not trained to do that.

Additionally, one must consider that proposed materials can still be extremely implausible, despite satisfying our definition of stability, or count a new material according to `StructureMatcher`, but a domain expert would not agree. Further discussion of these issues can be found in the work by Cheetham & Seshadri (2024). Furthermore, our tests using `StructureMatcher` rely on it defining an equivalence relation between structures; however, it does not due to its `rtol` parameter which means the reflexive property does not always hold. (However it does hold approximately.)

Finally, we want to emphasize that although we believe our *Cost* metrics to be a good faith attempt to compare models, of course number of integration steps is only one of many dimensions to evaluate the cost of generating a novel material. We did not include training or relaxation time in these computations, for example. (Training time was approximately the same across models and relaxation time is independent of the generation method. Although, relaxation can depend on the

accuracy of a reconstructed/generated structure.)

B. Further Results

B.1. Crystal Structure Prediction (CSP)

To better understand how the components of FlowMM affect the performance on the CSP task, we performed several ablation studies and included estimates of uncertainty. We focused on three aspects in particular: (a) Ablating velocity anti-annealing, (b) ablating our proposed base distribution and unconstrained transformation for lattice parameters $\boldsymbol{l}$, and (c) estimating uncertainty in the inference stage, *not during training*, by rerunning reconstruction with varied random seeds.

Velocity anti-annealing is an inference-time hyperparameter that controls how much to scale the velocity prediction from our learned vector field during integration, see (19). We choose to apply velocity anti-annealing to no variables, fractional coordinates $\boldsymbol{f}$, lattice parameters $\boldsymbol{l}$, or both variables $\boldsymbol{f}, \boldsymbol{l}$. This amounts to a comprehensive ablation of the method. In CSP, we found that it was generally beneficial to apply velocity anti-annealing to $\boldsymbol{f}$, but not $\boldsymbol{l}$. We note that FlowMM typically saw improved performance compared to competitors without the need to scale the velocity with anti-annealing. We report these results in Tables 3 and 4. We also reproduce the match rate as a function of integration steps using FlowMM without Inference Anti-Annealing in Figure 8.

We describe our bespoke parameterization of $\boldsymbol{l}$ in Section 3 including a custom base distribution and a transformation to unconstrained space. Our neural network representation (Appendix C) does not depend on "physical" lattice parameters. Therefore, it is also possible to simply use a typical normal base distribution, without transforming to unconstrained space, and let flow matching take care of learning the target distributions without inductive bias. Note that the ablation of the base distribution for lattice parameters requires training another model. During inference, such a model can produce representations that do not correspond to a real crystal; we will simply consider those generations as having failed. We jointly ablate these inductive biases along with a velocity anti-annealing ablation. We find that no matter the velocity anti-annealing scheme, our lattice parameter inductive biases provide a siginifcant performance boost.

Finally, for each case and dataset, we reran the generation from the corresponding trained model with the same hyperparameters three times. This therefore indicates variation at inference time, but not variation during training. (Although these are newly trained models compared to what is reported in Table 1 for every set of hyperparameter settings.) See Table 3 for the ablation result on the unit test datasets and see Table 4 for the realistic datasets. Some unit test datasets reported the same match rate across three reconstructions when $n = 20$, that's what leads to ± 0.0.

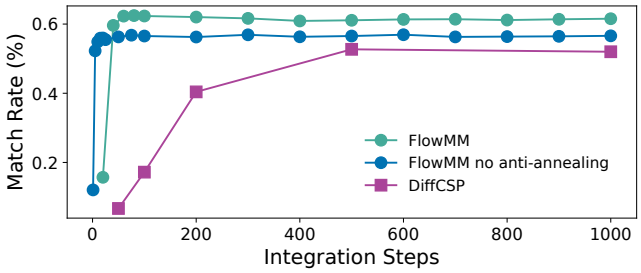
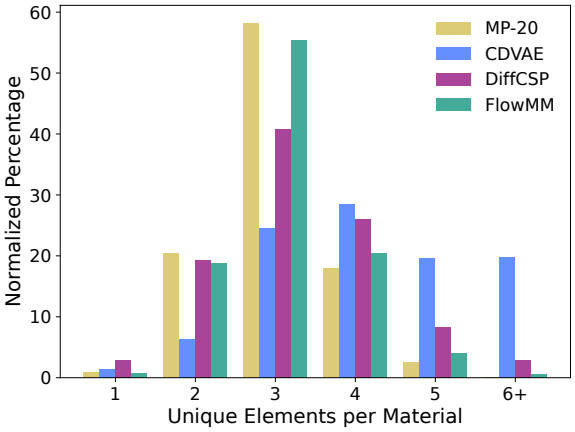


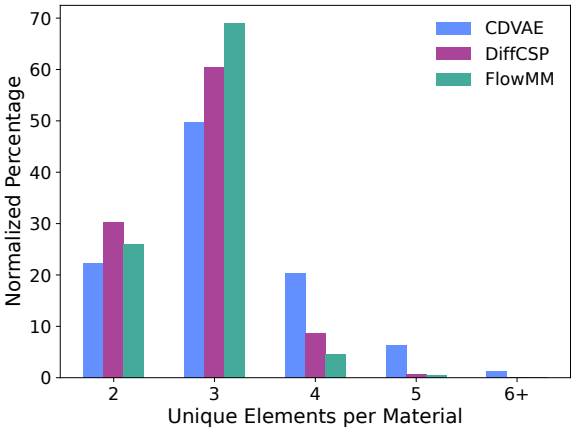
Figure 8. Match rate as a function of number of integration steps on MP-20. FlowMM achieves a higher maximum match rate than DiffCSP overall without without Inference Anti-Annealing (on $\boldsymbol{f}$ not $\boldsymbol{l}$). Even without Inference Anti-Annealing, FlowMM outperforms DiffCSP at every number of integration steps.

B.2. De Novo Generation (DNG)

We present the distribution of generated stable crystals from CDVAE, DiffCSP, and FlowMM trained on MP-20 in Figure 9b. (For context, we include generations without regard to stability in Figure 9a) The number of structures determined to be stable diminishes quickly as a function of N-ary, implying that models generating high N-ary materials do not relax to stable structures after density functional theory calculations.



(a) Materials from MP-20 & generative methods.



(b) Stable materials from generative methods.

Figure 9. (a) Figure 3 repeated for context. It represents a normalized histogram of the number of unique elements per crystal (N-ary) for MP-20 and the generative methods. (b) A normalized histogram of the number of unique elements per *stable* crystal (N-ary) of structures generated by CDVAE, DiffCSP, and FlowMM. All structures are stable and were relaxed using density functional theory. Despite generating a large number of high N-ary structures, CDVAE and DiffCSP find relatively few stable ones after relaxation. The FlowMM columns correspond to generations with 1000 integration steps.

Table 3. Results from ablation study on unit test datasets

				Perov-5		Carbon-24	
				Match Rate (%) ↑	RMSE ↓	Match Rate (%) ↑	RMSE ↓
# of Samples	Lattice $\boldsymbol{l}$ Base Dist.	Anneal Coords	Anneal Lattice				
1	ablated	False	False	52.62 ± 1.06	0.2822 ± 0.0005	15.70 ± 0.85	0.4262 ± 0.0033
			True	0.00 ± 0.00		1.84 ± 0.24	0.4265 ± 0.0041
		True	False	52.07 ± 1.25	0.2189 ± 0.0062	17.01 ± 0.59	0.4213 ± 0.0024
			True	0.00 ± 0.00		1.71 ± 0.12	0.4229 ± 0.0140
	proposed	False	False	57.63 ± 1.03	0.2556 ± 0.0039	21.95 ± 0.51	0.4097 ± 0.0017
			True	56.36 ± 0.42	0.1945 ± 0.0020	15.83 ± 0.51	0.3900 ± 0.0017
		True	False	56.49 ± 1.09	0.1976 ± 0.0025	20.61 ± 0.52	0.3984 ± 0.0027
			True	55.39 ± 0.20	0.1913 ± 0.0036	15.35 ± 0.68	0.3923 ± 0.0055
20	ablated	False	False	98.60 ± 0.00	0.0519 ± 0.0007	76.86 ± 0.29	0.3941 ± 0.0008
			True	0.00 ± 0.00		26.88 ± 1.12	0.4208 ± 0.0038
		True	False	98.60 ± 0.00	0.0372 ± 0.0006	77.93 ± 0.23	0.3870 ± 0.0015
			True	0.00 ± 0.00		26.26 ± 0.97	0.4211 ± 0.0026
	proposed	False	False	98.60 ± 0.00	0.0428 ± 0.0004	79.85 ± 0.36	0.3599 ± 0.0012
			True	98.60 ± 0.00	0.0334 ± 0.0008	75.85 ± 1.31	0.3349 ± 0.0007
		True	False	98.60 ± 0.00	0.0328 ± 0.0007	84.15 ± 0.54	0.3301 ± 0.0037
			True	98.60 ± 0.00	0.0331 ± 0.0007	76.08 ± 0.16	0.3376 ± 0.0035

Table 4. Results from ablation study on realistic datasets

				MP-20		MPTS-52	
				Match Rate (%) ↑	RMSE ↓	Match Rate (%) ↑	RMSE ↓
# of Samples	Lattice $\boldsymbol{l}$ Base Dist.	Anneal Coords	Anneal Lattice				
1	ablated	False	False	43.21 ± 0.47	0.1812 ± 0.0012	4.04 ± 0.21	0.3490 ± 0.0152
			True	0.00 ± 0.00		0.16 ± 0.06	0.4276 ± 0.0306
		True	False	49.15 ± 0.01	0.0866 ± 0.0012	5.27 ± 0.14	0.2567 ± 0.0091
			True	0.00 ± 0.00		0.15 ± 0.03	0.4255 ± 0.0273
	proposed	False	False	56.82 ± 0.42	0.1332 ± 0.0016	12.12 ± 0.36	0.2843 ± 0.0056
			True	59.23 ± 0.08	0.0562 ± 0.0018	14.72 ± 0.45	0.1734 ± 0.0020
		True	False	61.26 ± 0.14	0.0572 ± 0.0014	16.11 ± 0.17	0.1831 ± 0.0021
			True	59.19 ± 0.34	0.0577 ± 0.0008	14.71 ± 0.23	0.1724 ± 0.0012
20	ablated	False	False	73.59 ± 0.10	0.1449 ± 0.0006	15.81 ± 0.36	0.3225 ± 0.0025
			True	0.00 ± 0.00		1.56 ± 0.03	0.4298 ± 0.0014
		True	False	75.68 ± 0.20	0.0791 ± 0.0011	20.63 ± 0.24	0.2581 ± 0.0006
			True	0.00 ± 0.00		1.51 ± 0.05	0.4218 ± 0.0084
	proposed	False	False	76.55 ± 0.09	0.0834 ± 0.0005	28.99 ± 0.04	0.2445 ± 0.0008
			True	70.07 ± 0.04	0.0472 ± 0.0003	28.73 ± 0.25	0.1655 ± 0.0031
		True	False	75.81 ± 0.07	0.0479 ± 0.0004	34.05 ± 0.04	0.1813 ± 0.0012
			True	70.00 ± 0.11	0.0474 ± 0.0004	28.95 ± 0.09	0.1666 ± 0.0028

C. Neural network

We employ a graph neural network from Jiao et al. (2023) that adapts EGNN (Satorras et al., 2021) to fractional coordinates,

$$\boldsymbol{h}^i_{(0)} = \phi_{\boldsymbol{h}_{(0)}}(a^i) \tag{24}$$

$$\boldsymbol{m}^{ij}_{(s)} = \varphi_m(\boldsymbol{h}^i_{(s-1)}, \boldsymbol{h}^j_{(s-1)}, \boldsymbol{l}, \text{SinusoidalEmbedding}(f^j - f^i)), \tag{25}$$

$$\boldsymbol{m}^i_{(s)} = \sum_{j=1}^{N} \boldsymbol{m}^{ij}_{(s)}, \tag{26}$$

$$\boldsymbol{h}^i_{(s)} = \boldsymbol{h}^i_{(s-1)} + \varphi_h(\boldsymbol{h}^i_{(s-1)}, \boldsymbol{m}^i_{(s)}), \tag{27}$$

$$\dot{f}^i = \varphi_{\dot{f}}\left(\boldsymbol{h}^i_{(\max s)}\right) \tag{28}$$

$$\dot{\boldsymbol{l}} = \varphi_{\dot{\boldsymbol{l}}}\left(\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{h}^i_{(\max s)}\right) \tag{29}$$

where $\boldsymbol{m}^{ij}_{(s)}, \boldsymbol{m}^i_{(s)}$ represent messages at layer s between nodes i and j, $\boldsymbol{h}^j_{(s)}$ represents hidden representation of node j at layer s; $\varphi_m, \varphi_h, \phi_{\boldsymbol{h}_{(0)}}, \varphi_{\dot{f}}, \varphi_{\dot{\boldsymbol{l}}}$ represent parametric functions with all parameters noted together as θ. A symbol $\square \in \triangle$ with a dot above it $\dot{\square}$ represents the corresponding velocity components of the learned vector field, i.e. $\dot{\square} := v_t^{\triangle,\theta}(\boldsymbol{c}_t)$. Finally, we define

$$\text{SinusoidalEmbedding}(x) := (\sin(2\pi kx), \cos(2\pi kx))^T_{k=0,\ldots,n_{freq}}, \tag{30}$$

where n_{freq} is a hyperparameter. We standardized the $\boldsymbol{l}$ input to the network with z-scoring. We also standardized the outputs for predicted tangent vectors $\dot{\boldsymbol{f}}, \dot{\boldsymbol{l}}$. Models were trained using the `AdamW` optimizer (Loshchilov & Hutter, 2018).

We parameterize our loss as an affine combination. That means we enforce the following condition for all experiments:

$$\lambda_{\boldsymbol{l}} + \lambda_{\boldsymbol{f}} + \lambda_{\boldsymbol{a}} = 1 \tag{31}$$

enforced by

$$\tilde{\lambda}_{\boldsymbol{l}} + \tilde{\lambda}_{\boldsymbol{f}} + \tilde{\lambda}_{\boldsymbol{a}} := \tilde{\lambda}; \qquad \lambda_{\boldsymbol{l}} = \tilde{\lambda}_{\boldsymbol{l}}/\tilde{\lambda}, \qquad \lambda_{\boldsymbol{f}} = \tilde{\lambda}_{\boldsymbol{f}}/\tilde{\lambda}, \qquad \lambda_{\boldsymbol{a}} = \tilde{\lambda}_{\boldsymbol{a}}/\tilde{\lambda}. \tag{32}$$

In DNG, we introduce an additional loss term. When this term is included, we also include it in the affine combination.

We provide general and network hyperparameters in Table 5 and Table 6. Recall, all datasets use a $60 - 20 - 20$ split between training, validation, and test data. We apply the same split as Xie et al. (2021) and Jiao et al. (2023). More specific details exist in the corresponding experiment sections.

Table 5. General Hyperparameters

	Carbon	**Perov**	**MP-20**	**MPTS-52**
Max Atoms	24	20	20	52
Max Epochs	8000	6000	2000	1000
Total Number of Samples	10153	18928	45231	40476
Batch Size	256	1024	256	64

Table 6. Network Hyperparameters

	Value
Hidden Dimension	512
Time Embedding Dimension	256
Number of Layers	6
Activation Function	silu
Layer Norm	True

Crystal Structure Prediction We employed the network defined above for the CSP experiments. We swept over a grid and selected the model that maximized the match rate on 2,000 reconstructions from (a subset of) the validation set.

We swept learning rate $\in \{0.001, 0.0003\}$, weight decay $\in \{0.003, 0.001, 0.0\}$, gradient clipping = 0.5, $\tilde{\lambda}_{\boldsymbol{l}} = 1$, $\tilde{\lambda}_{\boldsymbol{f}} \in \{100, 200, 300, 400, 500\}$.

We performed multiple reconstructions using various values for the anti-annealing velocity scheduler with coefficient $s' \in \{0, 1, 2, 3.5, 5, 10\}$. We found that the velocity scheduler to be most effective when applied to $\boldsymbol{f}$ alone. Ablation tests of this phenomenon can be found in Appendix B.

Table 7. CSP Hyperparameters

	Carbon	Perov	MP-20	MPTS-52
Learning Rate	0.001	0.0003	0.0001	0.0001
Weight Decay	0.0	0.001	0.001	0.001
$\tilde{\lambda}_{\boldsymbol{f}}$ (Frac Coords)	400	1500	300	300
$\tilde{\lambda}_{\boldsymbol{l}}$ (Lattice)	1.0	1.0	1.0	1.0
s' (Anti-Anneal Slope)	2.0	1.0	10.0	5.0
Anneal $\boldsymbol{f}$	False	False	True	True
Anneal $\boldsymbol{l}$	False	False	False	False

Table 8. DNG Hyperparameters

	Value
Learning Rate	0.0005
Weight Decay	0.005
$\tilde{\lambda}_{\boldsymbol{a}}$ (Atom Type)	300
$\tilde{\lambda}_{\boldsymbol{f}}$ (Frac Coords)	600
$\tilde{\lambda}_{\boldsymbol{l}}$ (Lattice)	1.0
$\tilde{\lambda}_{\text{sce}}$ (Cross Entropy)	20
s' (Anti-Annealing Slope)	5.0
Anneal $\boldsymbol{a}$	False
Anneal $\boldsymbol{f}$	True
Anneal $\boldsymbol{l}$	True

De novo generation For the unconditional experiment, we made some changes to the network above that we found favorable for the featurization of the crystal. The new network and featurization is:

$$\boldsymbol{h}^i_{(0)} = \phi_{\boldsymbol{h}_{(0)}}(a^i) \tag{33}$$

$$\boldsymbol{m}^{ij}_{(s)} = \varphi_m\left(\boldsymbol{h}^i_{(s-1)}, \boldsymbol{h}^j_{(s-1)}, \boldsymbol{l}, \text{SinusoidalEmbedding}\left(\log_{f^i}(f^j)\right), \boldsymbol{z}(n), \frac{\tilde{\boldsymbol{l}}^T\tilde{\boldsymbol{l}}\boldsymbol{f}}{\|\tilde{\boldsymbol{l}}^T\tilde{\boldsymbol{l}}\boldsymbol{f}\|}\right), \tag{34}$$

$$\boldsymbol{m}^i_{(s)} = \sum_{j=1}^{N} \boldsymbol{m}^{ij}_{(s)}, \tag{35}$$

$$\boldsymbol{h}^i_{(s)} = \boldsymbol{h}^i_{(s-1)} + \varphi_h(\boldsymbol{h}^i_{(s-1)}, \boldsymbol{m}^i_{(s)}), \tag{36}$$

$$\dot{f}^i = \varphi_{\dot{f}}\left(\boldsymbol{h}^i_{(\max s)}\right) \tag{37}$$

$$\dot{\boldsymbol{l}} = \varphi_{\dot{\boldsymbol{l}}}\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{h}^i_{(\max s)}, \sum_{i=1}^{n}\boldsymbol{h}^i_{(\max s)}\right) \tag{38}$$

$$\dot{a}^i = \varphi_{\dot{a}}\left(\boldsymbol{h}^i_{(\max s)}\right) \tag{39}$$

$$\tag{40}$$

where $\log_{f^i}(f^j)$ is defined in (12) as the logmap for the flat torus, $\boldsymbol{z}(n)$ represents a learned embedding of the number of atoms n in the crystal's unit cell with parameters concatenated to θ, $\frac{\tilde{\boldsymbol{l}}^T\tilde{\boldsymbol{l}}\boldsymbol{f}}{\|\tilde{\boldsymbol{l}}^T\tilde{\boldsymbol{l}}\boldsymbol{f}\|}$ is the cosine of the angles between the Cartesian edge between atoms and the three lattice vectors (Zeni et al., 2023), $\varphi_{\dot{\boldsymbol{l}}}$ takes in both mean and sum pooling across nodes, and $\varphi_{\dot{a}}$ represents a parametric function with parameters concatenated to θ. A symbol $\square \in \triangle$ with a dot above it $\dot{\square}$ represents the corresponding velocity components of the learned vector field, i.e. $\dot{\square} := v_t^{\triangle,\theta}(\boldsymbol{c}_t)$. The additional edge features in (34) are invariant to translation and rotation. Recall that at $t = 0$, a^i is drawn randomly from the base distribution.

We included an additional loss term with a Lagrange multiplier in our loss function. Namely a version of what Chen et al. (2022) call *sigmoid cross entropy* in Appendix B.2, adapted for atom types represented as analog bits:

$$\mathfrak{L}_{\text{sce}} := -\log\sigma\left(\boldsymbol{a}_1 \cdot \hat{\boldsymbol{a}}_1\right), \tag{41}$$

$$\text{with } \hat{\boldsymbol{a}}_1 := (1-t)\,\dot{\boldsymbol{a}}_t + \boldsymbol{a}_t, \tag{42}$$

where σ is the logistic sigmoid, $\boldsymbol{a}_1 \in \{-1, 1\}^h$ is the target analog bit-style atom type vector, t is the time where the loss is evaluated, $\boldsymbol{a}_t$ is the point along the path between $\boldsymbol{a}_0$ and $\boldsymbol{a}_1$ at time t, $\cdot$ represents the inner product between vectors, and $\dot{\boldsymbol{a}}_t := v_t^{\mathcal{A},\theta}(\boldsymbol{c}_t)$. This represents a one-step numerical estimate of the final predicted position of $\boldsymbol{a}_t$ at $t = 1$. We add this term into the objective (15) in affine combination with unnormalized Lagrange multiplier $\tilde{\lambda}_{\text{sce}}$ suitably normalized as λ_{sce} .

We performed a sweep over the hyperparameters learning rate $\in \{0.0001, 0.0003, 0.0005, 0.0007, 0.001\}$, weight decay $\in \{0.0, 0.0001, 0.0005, 0.001, 0.003, 0.005\}$, gradient clipping = 0.5, $\tilde{\lambda}_{\boldsymbol{l}} = 1$, $\tilde{\lambda}_{\boldsymbol{f}} \in \{40, 100, 200, 300, 400, 600, 800\}$, $\tilde{\lambda}_{\boldsymbol{a}} \in \{40, 100, 200, 300, 400, 600, 800, 1200, 1600\}$, and $\tilde{\lambda}_{\text{sce}} \in \{0, 20\}$, and velocity schedule coefficient $s' \in \{0, 1, 2, 5\}$

We performed model selection using generated samples from each model in the sweep. After computing the proxy metrics on those samples, we collected the ≈ 50 models that were in the top 86th percentile on (both structural & compositional) validity, Wasserstein distance in density (ρ), and Wasserstein distance in number of unique elements (N_{el}). From those models, we prerelaxed the generations using CHGNet (Deng et al., 2023) and took the model which produced the most metastable structures (CHGNet energy above hull < 0.1 eV/atom). We reported the results from the one which then had the best performance on Stability Rate computed using the number of metastable structures.

D. Enforcing G-invariance of marginal probability path

We assume the target distribution q is G-invariant, where G is defined as in the "Symmetries of crystals" paragraph, *i.e.*, for each $g \in G$, where $g = (\sigma, Q, \tau)$ consists of (i) permutation of atoms together with their fractional coordinates, (ii) rotation of the lattice, and (iii) translation of the fractional coordinates. Firstly, we show that generally for marginal probability paths where $p_1 = q$ as in (8), in order to have $p_t(x)$ be G-invariant, it is sufficient to have $p_t(x|x_1)$ satisfy a simple pairwise G-invariant condition.

Theorem D.1. *For pairwise G-invariant conditional probability path $p_t(x|x_1)$, meaning $p_t(g \cdot x|g \cdot x_1) = p_t(x|x_1)\ \forall g \in G,\ x, x_1 \in \mathcal{C}$, the construction in (8) defines a G-invariant marginal distribution $p_t(x)$.*

Proof.

$$
\begin{aligned}
p_t(g \cdot x) &= \int p_t(g \cdot x|x_1) q(x_1) dx_1 && \text{defn. from (8)} \\
&= \int p_t(x|g^{-1} \cdot x_1) q(x_1) dx_1 && \text{pairwise } G\text{-invariance of } p_t(x|x_1) \\
&= \int p_t(x|g^{-1} \cdot x_1) q(g^{-1} \cdot x_1) dx_1 && G\text{-invariance of } q \\
&= \int p_t(x|g^{-1} \cdot x_1) q(g^{-1} \cdot x_1) \underbrace{\left|\det J_{g^{-1}}\right|}_{=1} d(g^{-1} \cdot x_1) && \text{change of variables} \\
&= \int p_t(x|\tilde{x}_1) q(\tilde{x}_1) d\tilde{x}_1 && \tilde{x}_1 = g^{-1} \cdot x_1 \\
&= p_t(x)
\end{aligned}
$$

□

Constructing conditional flows that imply pairwise G-invariant probability paths In order to construct a pairwise G-invariant $p_t(x|x_1)$, we make use of three main approaches. One is to enforce G-equivariant vector fields, which correspond to G-equivariant flows and thus generate G-invariant probabilities, building on the observation of Köhler et al. (2020) to the pairwise case. Another is to simply make use of representations that are G-invariant, resulting in G-invariant probabilities. Finally, we take a novel approach of generalizing the construction of Riemannian Flow Matching to equivalence classes and constructing flows between equivalence classes.

For the first approach, we require a G-invariant base distribution and that $u_t(g \cdot x|g \cdot x_1) = g \cdot u_t(x|x_1)$. This ensures the flow satisfies $\psi_t(g \cdot x_0|g \cdot x_1) = g \cdot \psi_t(x_0|x_1)$ and thus resulting in a pairwise G-invariant probability path. This property is satisfied by the use of regular geodesic paths that we use during the training of Riemannian Flow Matching, because the shortest paths connecting any x and x_1 on the manifolds that we consider here (flat tori and Euclidean) are simply simultaneously transformed alongside x and x_1, for transformations such as permutation and rotation. We use this approach to enforce invariance to permutation of atoms.

The second approach is to bypass the need to enforce invariance in either $u_t(x|x_1)$ or $v_t^\theta(x)$ by instead using a representation of that is bijective with its entire equivalence class. We use this approach to enforce invariance to rotation of the lattice, by directly modeling angles and lengths.

The third approach is enabled by a generalization of the Riemannian Flow Matching framework in the case of a G-invariant q, relaxing the assumption that the conditional probability paths $p_t(x|x_1) = \delta(x - x_1)$ at $t = 1$. Instead, we allow $p_t(x|x_1) = \delta(x - \tilde{x}_1)$ as long as $\tilde{x}_1 \in [x_1]$, the equivalence class of x_1, *i.e.*, $[x] = \{g \cdot x | g \cdot \in G\}$.

Theorem D.2. *Allowing the possibility of G-invariant conditional flow $\psi_t(x_0|x_1)$, meaning $\psi_t(x_0|g \cdot x_1) = \psi_t(x_0|x_1)\ \forall g \in G,\ x, x_1 \in \mathcal{C}$, if $\psi_t(x_0|x_1) \in [x_1]\ \forall x, x_1 \in \mathcal{C}$, then the construction of $u_t(x)$ in (17) is valid and results in a marginal distribution that satisfies $p_1 = q$.*

Proof. For general flow functions $\psi_t(x_0|x_1)$, it follows a Dirac-delta conditional probability $p_t(x|x_0, x_1) = \delta(x - \psi_t)$, where ψ_t is short-hand for $\psi_t(x_0|x_1)$. Then we have that

$$
\begin{aligned}
p_{t=1}(x) &= \int p_{t=1}(x|x_0, x_1)p(x_0)q(x_1)dx_0dx_1 \\
&= \int \delta(x - \psi_{t=1})p(x_0)q(x_1)dx_0dx_1 \\
&= \int \delta(x - g \cdot x_1)p(x_0)q(x_1)dx_0dx_1 && \psi_{t=1} \in [x_1] \\
&= \int \delta(x - g \cdot x_1)p(x_0)q(g \cdot x_1)dx_0d(g \cdot x_1) && G\text{-invariance of } q \text{ \& change of variables} \\
&= \int \delta(x - \tilde{x}_1)p(x_0)q(\tilde{x}_1)dx_0d\tilde{x}_1 \\
&= q(x)
\end{aligned}
$$

□

The main implication of only needing to satisfy $\psi_1 \in [x_1]$ is that this allows us to also impose additional the constraints on our vector fields that were previously not possible. Specifically, we can now allow conditional vector fields that are entry-wise G-invariant, *i.e.*, $u_t(x|g \cdot x_1) = u_t(x|x_1)$ and $u_t(g \cdot x|x_1) = u_t(x|x_1)$. Note this results in flows that satisfy $\psi_t(x_0|g \cdot x_1) = \psi_t(x_0|x_1)$, and importantly, this implies the flow can no longer distinguish x_1 from other elements in its equivalence class; the flow ψ_t is purely a function of equivalence classes $[x_0]$ and $[x_1]$. However, as per above, this is still sufficient for satisfying $p_1 = q$. Simultaneously, allowing such conditional vector fields provides the means to satisfy the pairwise G-invariance condition we need for G-invariance of $p_t(x)$, *i.e.*, $p_t(g \cdot x|g \cdot x_1) = p_t(x|x_1)$.

Translation invariance with periodic boundary conditions On Euclidean space, one typical method of imposing translation invariance of a set of points is to remove the mean of the set and using a "mean-free" representation. This provides the ability to work with a representation that does not contain any information about translation, following the second approach described above. However, on flat tori (*i.e.*, with periodic boundary conditions), this approach is not possible because the mean of a set of points is not uniquely defined. Instead, we make use of the third approach described above and construct $\psi_t(f_0|f_1)$ such that it flows to a set of fractional coordinates that is equivalent to f_1. Since for the flat tori, translations in the tangent plane result in translations on the manifold, we propose simply removing the translation component of the conditional vector field resulting from the geodesic construction. This results in the "mean-free" conditional vector field in (14).