OMatG-flash: An All-Atom Flow Map with Reinforce Adjoint Matching for Scalable Materials Discovery

Anonymous authors
Paper under double-blind review

Abstract

The discovery of novel inorganic materials drives technological breakthroughs in critical fields such as computing and energy storage. Generative AI has promised to accelerate the materials discovery pipeline, but state-of-the-art flow and diffusion models remain bottlenecked by the cost of proposing candidate materials. To address this, we introduce OMatG-flash, an all-atom flow map for inorganic crystal structure prediction (CSP) and de novo generation (DNG). OMatG-flash is a Pareto-optimal inference engine for materials, sampling candidate materials with an order of magnitude fewer inference steps and less wall-clock time than existing flow and diffusion models while demonstrating benchmark performance on par with the state-of-the-art. To enable post-training fine-tuning we apply Reinforce Adjoint Matching to flow maps, further improving performance on the CSP task. OMatG-flash showcases the potential of flow maps to accelerate generation of high-quality candidate inorganic materials and demonstrates a step forward in sample throughput necessary for data-hungry materials discovery workflows.

1 Introduction

A central goal in materials discovery is that of proposing candidate inorganic crystalline materials with desired properties. This has implications in critical fields such as computing (Chen et al., 2019), energy storage (Pielichowska & Pielichowski, 2014), and sustainability (Boyd et al., 2019). Traditional materials discovery pipelines involve repeated synthesis and experimentation loops which are time-consuming and costly to perform Potyrailo et al. (2011). This has inspired the creation of computational algorithms which leverage molecular dynamics and electronic structure to propose materials without the need for lab-based synthesis De Breuck et al. (2025).

Such algorithms aim to solve this problem via repeated first-principles quantum chemical calculations which yield accurate results but are computationally expensive and often impractical at scale Pickard & Needs (2011). Despite unfavorable scaling, the deployment of such methods has culminated in the aggregation of large materials science datasets (Jain et al., 2013; Zagorac et al., 2019; Cavignac et al., 2026) along with a wealth of computational tools (Ong et al., 2013; Hjorth Larsen et al., 2017) for analyzing them. These tools, coupled with advances in machine learning and artificial intelligence, have spurred the development of highly accurate machine learning models for applications in chemistry and physics which can approximate energies and forces with chemical accuracy (Batzner et al., 2022; Batatia et al., 2023; Wood et al., 2025) or propose candidate compositions and crystal structures (Hautier et al., 2010; Merchant et al., 2023). In the domain of generative modeling for materials, diffusion- and flow-based generative models have shown particularly dominant performance Xie et al. (2022); Jiao et al. (2024); Miller et al. (2024); Zeni et al. (2024); Hoellmer et al. (2025). Nevertheless, there is an ever-present need to accelerate the rate at which candidate structures can be proposed. As continued effort and innovation flow into scaling autonomous labs Coley et al. (2019); MacLeod et al. (2020) and accelerated downstream processing Cohen et al. (2025), so too must the scalability of generative algorithms grow to saturate these systems.

Aforementioned flow and diffusion models are fundamentally limited by the need to numerically integrate differential equations to propose candidate materials. To address this we introduce OMatG-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

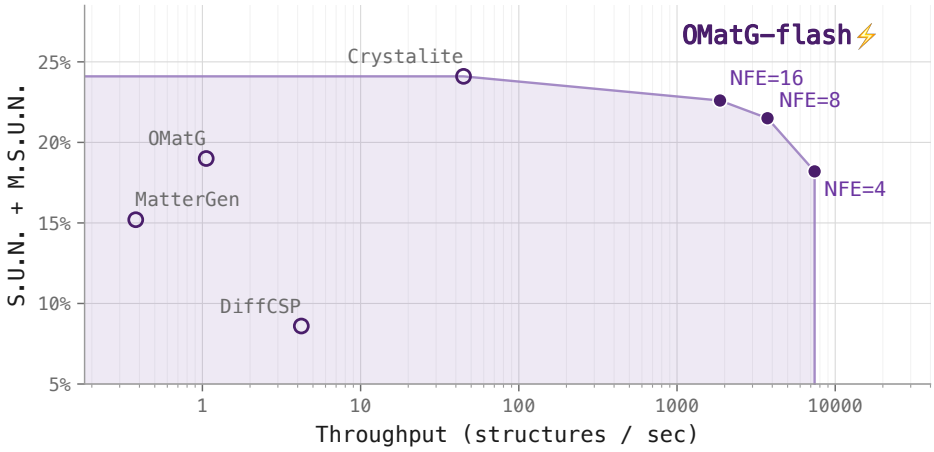


Figure 1: **Pareto-Optimal Inference**. Pareto frontier of combined S.U.N. and M.S.U.N. vs. throughput is shown for existing models. OMatG-flash is Pareto-optimal, surpassing existing models by an order of magnitude in terms of throughput while maintaining competitive raw performance on the DNG task. Solid points indicate OMatG-flash under several number of function evaluation (NFE) budgets.

flash, a transformer-based flow map for CSP and DNG of inorganic materials. OMatG-flash boasts an order of magnitude faster wall-clock inference time and far fewer neural network evaluations than existing flow and diffusion model architectures while maintaining equivalent performance. Furthermore, we extend post-training reinforcement learning methods developed for flow matching to flow maps, attaining performance competitive with respect to the state-of-the-art on the CSP task with few function evaluations. In contrast to conventional flow map distillation methods, OMatG-flash is trained directly from data unlike teacher-distilled flow maps which require access to a pretrained generative model. Furthermore, OMatG-flash can be optimized for a target reward via Reinforce Adjoint Matching while retaining its low inference cost.

Our Contributions:

- We introduce OMatG-flash, a Pareto-optimal all-atom flow map that rivals existing flow- and diffusion-based models on materials-generation benchmarks while substantially improving inference-time scalability via few-step sampling (Figure 1).
- We show that post-training OMatG-flash with Reinforce Adjoint Matching (RAM)—to our knowledge, its first application to flow maps—achieves state-of-the-art performance on polymorph-aware CSP benchmarks compared to existing diffusion- and flow-based models.
- We characterize the key limitations of flow map training for scientific applications and outline prospective future directions for how they may be applied to other problems in materials design.

2 Related Work

Flows and Diffusions Diffusion and flow models formulate generative modeling as a transport problem; mapping a tractable easy-to-sample base distribution to a target distribution that is accessible only through data samples Albergo et al. (2023). The diffusion model approach uses a noising stochastic differential equation (SDE) to construct a regression target for estimating the drift of a reverse denoising SDE. With this learned drift, generation proceeds by sampling an initial noisy state and numerically integrating in time to produce an approximate data sample from the target Sohl-Dickstein et al. (2015); Song et al. (2021). In flow models the transport is framed deterministically in terms of an instantaneous velocity field which satisfies the probability flow ordinary differential equation (ODE) describing the probability path of the distribution ρ_t Liu et al. (2022); Lipman

et al. (2023); Albergo et al. (2023). This velocity field is numerically integrated via deterministic dynamics to draw approximate samples from the target.

Flow Maps and Consistency Models Unfortunately, the iterative numerical integration required to draw samples with flow- and diffusion-based models requires repeated model evaluations, hindering time-sensitive performance. This has inspired the creation of flow maps which learn a transport that requires few neural network evaluations per generated sample Song et al. (2023); Boffi et al. (2024; 2025); Woo et al. (2026). Flow maps have demonstrated performance matching both diffusion and flow models in various domains at a greatly reduced inference cost Yoo et al. (2026); Scarpellini et al. (2026). Flow maps exhibit success in language (Lee et al., 2026), biochemistry (Scarpellini et al., 2026), and physics (Ripken et al., 2026) while retaining the ability to be learned directly from data.

Generative Models for Materials Discovery Improvement in the performance and scalability of generative AI has inspired the creation of machine learning models which can rapidly propose physically stable and novel crystalline materials. Diffusion- and flow-based methods have demonstrated particularly strong performance in this domain Xie et al. (2022); Jiao et al. (2024); Miller et al. (2024); Zeni et al. (2024); Zeng et al. (2026); Lo et al. (2026). Subsequent advances in both the underlying algorithms and network architectures further improved generative accuracy Hoellmer et al. (2025); Cornet et al. (2025); Veljković et al. (2026), while conditioning strategies have enabled the generation of materials with targeted properties Kazeev et al. (2025); Prakash et al. (2026). Current CSP and DNG generative models have thus shifted focus to post-training algorithms with reinforcement learning Park & Walsh (2025); Hoellmer & Martiniani (2026), enabling the generative process to push beyond the underlying dataset it was trained on. Parallel to this, substantial effort has been devoted to training large language models to generate text-based representations of crystal structures Antunes et al. (2024); Xu et al. (2026); Bone et al. (2026).

3 METHODS

Unit Cell Representation We represent an N-atom periodic unit cell as an element $\mathbf{c} \in \mathcal{M}$ of the product manifold

$$\mathbf{c} = (\mathbf{A}, \mathbf{F}, \mathbf{y}) \in \mathcal{M} := \mathbb{R}^{N \times d_A} \times \mathbb{T}^{N \times 3}_{[0,1)} \times \mathbb{R}^6, \tag{1}$$

where $\mathbf{A}$ is the atomic descriptor matrix with embedding dimension d_A, $\mathbf{F}$ represents the fractional coordinates, and $\mathbf{y}$ is a rotation-invariant unit-cell representation obtained from the Cholesky decomposition of the lattice metric tensor $\mathbf{G} = \mathbf{L}\mathbf{L}^\top$ (Appendix A). We define the Riemannian stochastic interpolant and its endpoint-conditioned velocity as

$$\mathbf{c}_t = \exp_{\mathbf{c}_0}\left(\beta_t \log_{\mathbf{c}_0}(\mathbf{c}_1)\right), \qquad \dot{\mathbf{c}}_t = \tfrac{\dot{\beta}_t}{1-\beta_t} \log_{\mathbf{c}_t}(\mathbf{c}_1). \tag{2}$$

The exponential and logarithm maps act componentwise on $\mathcal{M}$ Grenioux et al. (2026), with their definitions and further details regarding the crystal representation provided in Appendix A. We take $\beta_t = t$ to obtain a geodesic interpolant and use a standard normal base distribution $\mathbf{c}_0 \sim \rho_0$, while $\mathbf{c}_1 \sim \rho_1$ is sampled from the dataset.

Flow Matching The corresponding velocity field which bridges ρ_0 and ρ_1 is given by a conditional expectation $\mathbf{v}_t(\mathbf{c}) = \mathbb{E}[\dot{\mathbf{c}}_t \mid \mathbf{c}_t = \mathbf{c}]$. Under suitable regularity conditions, this velocity field generates the marginal probability path ρ_t induced by the stochastic interpolant Lipman et al. (2023). An approximation $\mathbf{v}^\theta_t$ is learned by minimizing

$$\mathcal{L}_{\text{vel}}(\theta) = \int_0^1 \mathbb{E}\left[\left\|\mathbf{v}^\theta_t(\mathbf{c}_t) - \dot{\mathbf{c}}_t\right\|^2\right] \mathrm{d}t \tag{3}$$

where the expectation is taken over $\mathbf{c}_1 \sim \rho_1$ and $\mathbf{c}_0 \sim \rho_0$. Samples are generated by integrating

$$\dot{\mathbf{c}}_t = \mathbf{v}^\theta_t(\mathbf{c}_t), \qquad \mathbf{c}_0 \sim \rho_0, \quad t \in [0,1], \tag{4}$$

giving terminal samples $\mathbf{c}_1 \sim \rho^\theta_1 \approx \rho_1$ approximating the dataset.

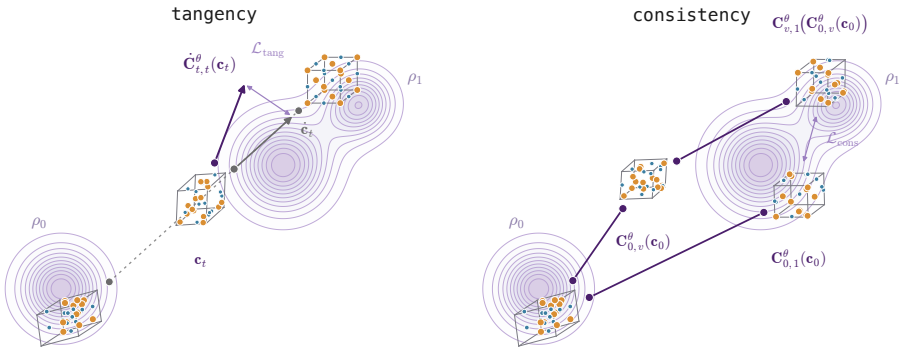


Figure 2: **Flow Map Training.** OMatG-flash training minimizes a tangency loss (**Left**) which, similar to standard flow matching, ensures that, at diagonal time t, the velocity associated with the instantaneous flow map is aligned with the ground truth velocity field. This is done by drawing an interpolant (dotted gray line) between noise and data and regressing onto the analytical velocity (bold gray arrow) $\dot{\mathbf{c}}_t$. In tandem, a consistency loss (**Right**) enforces that flow map steps can be properly composed to cover any given time range.

3.1 Flow Maps

A two-time flow map $\mathbf{C}_{s,u}$ directly transports samples from the marginal ρ_s to ρ_u. For $\mathbf{c}_s \sim \rho_s$, it is defined as the solution to the ODE over the interval $[s, u]$:

$$\mathbf{C}_{s,u}(\mathbf{c}_s) := \mathbf{c}_s + \int_s^u \mathbf{v}_\tau(\mathbf{c}_\tau)\,\mathrm{d}\tau. \tag{5}$$

Rather than learning only the velocity through equation 3, a flow map model directly approximates the solution operator $\mathbf{C}^\theta_{s,u} \approx \mathbf{C}_{s,u}$ in equation 5 Song et al. (2023); Boffi et al. (2024). This enables sampling from ρ_1 using a small number of composed function evaluations. A flow map must satisfy a *tangency condition*;

$$\dot{\mathbf{C}}_{t,t}(\mathbf{c}_t) := \tfrac{\partial}{\partial u}\mathbf{C}_{t,u}(\mathbf{c}_t)\big|_{u=t} = \mathbf{v}_t(\mathbf{c}_t), \tag{6}$$

Ensuring that, in the instantaneous limit, the flow map evolves according to the velocity field that generates the marginal probability path ρ_t. To enable few-step sampling it must also satisfy a *consistency condition*;

$$\mathbf{C}_{s,w}(\mathbf{c}_s) = \mathbf{C}_{v,w}(\mathbf{C}_{s,v}(\mathbf{c}_s))\,, \qquad s < v < w, \tag{7}$$

imposing that a composition of steps from time s to v, then v to w is equivalent to a single step over the interval s to w.

Riemannian Flow Maps We employ Riemannian MeanFlow (Woo et al., 2026) to parameterize the flow map jump $\mathbf{C}^\theta_{s,u}$. Specifically, we use an endpoint prediction network $\hat{\mathbf{c}}^\theta_1$ so that the jump from time s to $u > s$ becomes

$$\mathbf{C}^\theta_{s,u}(\mathbf{c}_s) = \exp_{\mathbf{c}_s}\left(\frac{u-s}{1-s}\log_{\mathbf{c}_s}\left(\hat{\mathbf{c}}^\theta_1(\mathbf{c}_s, s, u)\right)\right). \tag{8}$$

To compute $\hat{\mathbf{c}}^\theta_1$, we apply Crystalite's neural network architecture (Veljković et al., 2026). We combine each atomic descriptor with its corresponding fractional coordinate to form one token per atom and encode the lattice as a separate token, producing $N + 1$ tokens in total. These tokens are then processed by a two-time scalable interpolant transformer (SiT) backbone, foregoing equivariance for speed Ma et al. (2024); Peebles & Xie (2023).

A Riemannian flow map (Woo et al., 2026) is learned by substituting $\dot{\mathbf{C}}^\theta_{t,t} = \frac{\log_{\mathbf{c}_t}(\mathbf{c}^\theta_1(\mathbf{c}_t,t,t))}{1-t}$ for the learned velocity in equation 3 to enforce tangency. We then add a consistency loss $\mathcal{L}_{\mathrm{C}}$ to

enforce equation 7. The total training objective is finally given by a weighted linear combination of a tangency loss,

$$\mathcal{L}_{\mathrm{T}}(\theta) = \int_0^1 \mathbb{E}\left[\gamma_t \left\| \dot{\mathbf{C}}^{\theta}_{t,t}(\mathbf{c}_t) - \dot{\mathbf{c}}_t \right\|^2\right] \mathrm{d}t, \tag{9}$$

and a consistency loss

$$\mathcal{L}_{\mathrm{C}}(\theta) = \int_0^1 \int_0^w \int_s^w \mathbb{E}\Big[\gamma_{s,w} \left\| \log_{\mathbf{c}_s}\left(\mathbf{C}^{\theta}_{s,w}(\mathbf{c}_s)\right) - \mathtt{sg}\left(\log_{\mathbf{c}_s}\left(\mathbf{C}^{\theta}_{v,w}(\mathbf{C}^{\theta}_{s,v}(\mathbf{c}_s))\right)\right)\right\|^2\Big] \,\mathrm{d}v\,\mathrm{d}s\,\mathrm{d}w. \tag{10}$$

A weighted combination of these yields a flow map objective for crystalline unit cells:

$$\mathcal{L}_{\mathrm{tot}}(\theta) = \lambda_T \mathcal{L}_{\mathrm{T}}(\theta) + \lambda_C \mathcal{L}_{\mathrm{C}}(\theta). \tag{11}$$

Here, λ_{T} and λ_{C} are per-field weights controlling the relative contributions of the tangency and consistency losses, respectively, and $\mathtt{sg}$ denotes the stop-gradient operator. γ are time-dependent weighting functions chosen for stability Woo et al. (2026) (Appendix B.3). The expectations are taken over endpoints $\mathbf{c}_0 \sim \rho_0$ and $\mathbf{c}_1 \sim \rho_1$ sampled independently. We illustrate flow map training conceptually in Figure 2.

3.2 Reinforce Adjoint Matching for All-Atom Flow Maps

Flow map inference is not naturally compatible with standard policy-optimization methods such as DDPO Black et al. (2024), PPO Schulman et al. (2017), and GRPO Shao et al. (2024). Flow maps define deterministic generative dynamics and learn finite-time transport rather than a stochastic policy. Policy-gradient methods require a stochastic process to formulate generation as a Markov decision process. Existing methods for flow matching post-training navigate around this hurdle by either converting the ODE in equation 4 to an SDE via the score or by artificially injecting noise into this ODE Liu et al. (2025); Hoellmer & Martiniani (2026). While these approaches have shown promising results, they do not transfer naturally to flow maps, which do not admit a straightforward score-based reformulation as an SDE except for in the instantaneous limit. Reinforce Adjoint Matching (RAM), however, provides a path for fine-tuning a learned flow map Bergmeister et al. (2026).

We adapt RAM to enable flow map reinforcement learning. By exploiting the tangent condition in equation 6 we frame the RAM loss for an all-atom flow map as

$$\mathcal{L}_{\mathrm{RAM}}(\theta) = \int_0^1 \mathbb{E}\left[\left\| \dot{\mathbf{C}}^{\theta}_{t,t}(\mathbf{c}_t) - \mathtt{sg}\left(\dot{\mathbf{C}}^{\mathrm{ref}}_{t,t}(\mathbf{c}_t) + r(\mathbf{c}_1)\left(\dot{\mathbf{c}}_t - \dot{\mathbf{C}}^{\theta}_{t,t}(\mathbf{c}_t)\right)\right)\right\|^2\right] \mathrm{d}t, \tag{12}$$

which we combine with a consistency loss to yield $\mathcal{L}^{\mathrm{FM}}_{\mathrm{RAM}}(\theta) = \lambda^{\star}_{\mathrm{RAM}}\mathcal{L}_{\mathrm{RAM}}(\theta) + \lambda^{\star}_{C}\mathcal{L}_{\mathrm{C}}(\theta)$. Weights $\lambda^{\star}_{\mathrm{RAM}}$ and $\lambda^{\star}_{C}$ are indicated with a star to denote post-training. Minimizing this objective requires no reparameterization of the learned flow map nor does it require any sort of noise injection which would corrupt the generative dynamics. This fine-tunes a model to sample a reward-tilted target $\rho^{\star}_1(\mathbf{c}) = \rho_1(\mathbf{c})e^{r(\mathbf{c})}$.

We implement RAM post-training only for the CSP task using the energy as a reward model, $r(\mathbf{c}_1) = -\lambda_E E(\mathbf{c}_1)$, where λ_E is a hyperparameter. This reward pushes the generated distribution towards more stable regions of the energy surface, improving performance at the CSP task. We use the MACE-MPA-0 foundational machine learned interatomic potential (MLIP) as the chosen energy model for this task to balance high-fidelity energy evaluation with speed Batatia et al. (2023). To stabilize post-training we compute the reward after taking 50 relaxation steps for each generated structure $\mathbf{c}_1$. More detail on the OMatG-flash-RAM implementation is provided in Appendix E.

4 Experiments

We train OMatG-flash on two inorganic materials datasets to benchmark performance on CSP and DNG: **MP-20**, which comprises 45,231 crystal structures curated from the Materials Project (Jain et al., 2013), and **Alex-MP-20**, which is a larger dataset comprising 675,204 structures from the Materials Project and the Alexandria database Cavignac et al. (2026). Each dataset contains crystals with at most 20 atoms in the unit cell and covers a diverse set of compositions and structures. We

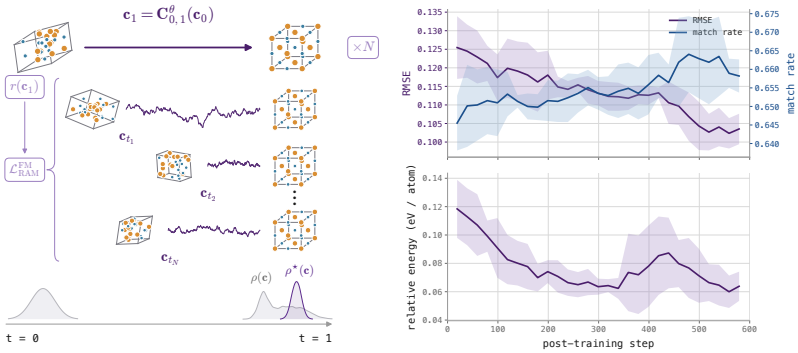


Figure 3: **OMatG-flash-RAM Post-Training.** A conceptual overview (**Left**) of RAM post-training for OMatG-flash is presented. OMatG-flash is used to generate a sample, $\mathbf{c}_1$, using few NFE. N replicas of this structure are made and independently renoised. Renoised samples $\{\mathbf{c}_{t_i}\}_{i=1}^{N}$ and the reward signal $r(\mathbf{c}_1)$ are used to update the model via the RAM objective in equation 12. (**Right**) The evolution of relative energy with respect to known crystal structures, RMSE, and match rate all trend desirably during fine-tuning. Relative energies are reported after relaxation, indicating that RAM post-training enables the model to produce structures which minimize to lower energies. Match rate and RMSE are computed before any relaxation.

evaluate OMatG-flash (pretrained) for both CSP and DNG of inorganic materials and OMatG-flash-RAM (post-trained) for CSP. All reported metrics for competing models are given as reported in existing work Seong et al. (2026); Betala et al. (2026).

OMatG-flash is trained on the canonical data splits for both of these datasets Xie et al. (2022); Zeni et al. (2024) as well as the new polymorph-split **MP-20-ps** Martirossyan et al. (2025). Results with respect to the original data splits are given solely to provide a fair comparison to existing work and we strongly urge future generative modeling efforts to train, evaluate, and compare to the polymorph split (ps) versions of this data using appropriate polymorph-aware metrics as plain MP-20 has known leakage across splits Martirossyan et al. (2025).

4.1 Crystal Structure Prediction

A central challenge in inorganic materials design is to predict crystal structure given only the constituent elements of atoms in the unit cell. The performance of OMatG-flash at CSP is primarily assessed via match rates and root-mean-squared-error (RMSE) between generated and reference crystals using Pymatgen's `StructureMatcher` algorithm which is used to report the fraction of generated crystal structures which match an index-paired test-set structure up to a chosen tolerance. While informative, match rates depend on arbitrary indexing which is flawed when considering the polymorphism inherent to crystalline materials Martirossyan et al. (2025). We therefore report and highlight our results on METRe (match-everyone-to-reference) and cRMSE (corrected RMSE), recently proposed polymorph-aware metrics which improve upon the flawed match rate benchmark. METRe addresses the failures of one-to-one match rate by instead searching for a match among *all* generated structures for each test set structure to remove any dependence on arbitrary indexing. The associated RMSE values measure the similarity between two matched structures in terms of a unitless distance whereas cRMSE adds a tolerance-based penalty for each non-matching case (Appendix F). For completeness we report both one-to-one match rates and RMSE along with METRe and cRMSE, but strongly urge wider adoption of polymorph-aware METRe and cRMSE. We report the performance of OMatG-flash and OMatG-flash-RAM in Tables 1 and 2. Post-training curves of the match rate, RMSE, and relative energy are shown in Figure 3.

4.2 De Novo Generation

DNG is the most general materials design problem, requiring proposal of not only the crystal structure but also the composition of elements within a crystal. We benchmark OMatG-flash's performance at this task by computing stability, uniqueness, and novelty (S.U.N.) and metastability,

Table 1: **CSP Performance on MP-20 and Alex-MP-20.** OMatG-flash and OMatG-flash-RAM demonstrate strong performance on one-to-one match rate where the subscript indicates the NFE used for inference. Values unavailable in the literature are indicated with dashes. k indicates that inference was performed k times and the best match from each generated set is taken. Each slash-separated entry reports `valid or invalid` / `valid`, where the first value is computed over all generated structures and the second only over structures classified as valid. **Bold** indicates best overall and underlined indicates best on a $k = 1$ budget.

		MP-20		Alex-MP-20	
Method	k	MR (%) ↑	RMSE ↓	MR (%) ↑	RMSE ↓
DiffCSP	1	57.82 / 52.51	0.0627 / 0.0600	–	–
FlowMM	1	66.22 / 59.98	0.0661 / 0.0629	–	–
MCFlow	1	70.38 / 64.08	**0.0592** / 0.0561	–	–
OMatG	1	69.83 / 63.75	0.0741 / 0.0720	72.50 / 64.71	0.1261 / 0.1251
Crystalite	1	– / 66.09	– / **0.0337**	– / 68.26	– / **0.0317**
OMatG-flash$_{16}$	1	67.15 / 60.79	0.1172 / 0.1136	66.19 / 59.05	0.1356 / 0.1342
	3	75.47 / 68.62	0.0954 / 0.0930	80.76 / 72.39	0.1055 / 0.1042
	5	78.27 / 71.19	0.0878 / 0.0860	**85.68** / **76.91**	0.0961 / 0.0948
OMatG-flash$_{16}$-RAM	1	68.90 / 62.55	0.0881 / 0.0858	70.52 / 62.87	0.0955 / 0.0939
	3	75.76 / 68.89	0.0743 / 0.0728	81.05 / 72.55	0.0807 / 0.0792
	5	**78.30** / **71.30**	0.0713 / 0.0702	84.84 / 76.08	**0.0764** / 0.0749

uniqueness, and novelty (M.S.U.N.) with respect to a known corpus of reference data Siron et al. (2025); Betala et al. (2026). We rely on LeMat-GenBench Betala et al. (2026) — an open-source leaderboard of generative models for inorganic materials generation — to compute S.U.N. and M.S.U.N. on 2500 sampled structures in Table 3. Stability and metastability are determined by comparing the energy of each generated structure to a convex hull of stable phases. Uniqueness measures the proportion of generated structures which are distinct chemically or structurally from all other generated structures whereas novelty identifies if a structure is distinct with respect to a chosen reference dataset of inorganic crystals. To identify matching crystals for uniqueness and novelty we use pymatgen's `StructureMatcher` algorithm. More information on the definition and practical calculation of these metrics is given in Appendix F. We report metrics for models trained on the MP-20 dataset as this dataset has the most reported entries on the LeMat-GenBench leaderboard.

5 Discussion

5.1 Results

For both CSP and DNG we compare OMatG-flash to unconditional all-atom diffusion- and flow-based models operating on crystalline unit cells. This is done to show how such models for materials can be amortized via a flow map. There are a variety of models which condition on space group to generate more symmetric structures Levy et al. (2025); Chang et al. (2025); Puny et al. (2025). Generative models utilizing different representations such as asymmetric units (Chang et al., 2026) or learned latent spaces (Joshi et al., 2025) have also been shown to yield promising performance for materials. Lastly, language models and unmasking transformer architectures have also demonstrated success for CSP and DNG Antunes et al. (2024); Kazeev et al. (2025); Seong et al. (2026). MaskGXT (Seong et al., 2026) is of this kind and is the current state-of-the-art overall for CSP on MP-20 and MP-20-ps with benchmark performance that exceeds ours. Their inference algorithm performs a double rollout, conditioning the second on the space group representation produced by the first, resulting in improved performance over other CSP models. For DNG, Crystalite (Veljković et al., 2026) is the current state-of-the-art and is reported as it is an unconditional diffusion model. A consensus on which architecture, crystal representation, and choice of conditioning signal is most advantageous for CSP and DNG is still an open question and deserves dedicated study which is out of the scope of this work.

Table 3: **DNG Performance.** OMatG-flash matches or exceeds all but Crystalite (Veljković et al., 2026) on S.U.N. (all NFE budgets) and combined S.U.N. and M.S.U.N. (8 or 16 NFE). The quality of proposed crystals is robust even to very few inference steps. OMatG-flash metrics are computed using LeMat-GenBench code for 2500 structures. Metrics for all other models appear as reported on the leaderboard. Best is **bolded** and second-best is underlined. An asterisk indicates that samples were not prerelaxed before benchmarking according to LeMat-GenBench.

Method	Valid ↑	Novel ↑	Stable ↑	S.U.N. ↑	S.U.N. + M.S.U.N. ↑
DiffCSP*	95.7	66.2	2.3	0.1	8.6
MatterGen	95.7	**70.5**	2.0	0.2	15.2
OMatG	96.4	51.2	11.6	1.0	19.0
MCFlow	**97.2**	52.2	11.9	0.7	19.6
Crystalite	**97.2**	53.2	**12.7**	**1.5**	**24.1**
OMatG-flash$_4$	96.4	66.9	7.2	1.4	18.2
OMatG-flash$_8$	96.8	63.4	7.8	1.0	21.5
OMatG-flash$_{16}$	96.8	61.0	8.8	1.2	22.6

CSP Results Among existing unconditional diffusion- and flow-based generative models OMatG-flash-RAM is state-of-the-art on METRe for MP-20-ps while the base OMatG-flash model is competitive with these methods. Post-training OMatG-flash proves remarkably effective for the CSP task overall, simultaneously improving METRe and cRMSE (Table 2) as well as RMSE in every case (Table 1), demonstrating the effectiveness of the proposed RAM extension and underscoring the potential for flow maps to accelerate high-quality CSP. While our results for $k > 1$ are not meant to be an apples-to-apples comparison with existing results against $k = 1$, we argue that our findings emphasize the benefit of rapid, high-quality inference over more expensive inference with slightly improved raw performance. On MP-20, performing k-inference greatly enhances the one-to-one match rates of OMatG-flash. OMatG-flash enjoys an order of magnitude speedup over existing models. Consequently, $k > 1$ inference can be done faster than $k = 1$ for conventional flow and diffusion models to yield stronger numbers.

Table 2: **CSP Performance on MP-20-ps.** OMatG-flash-RAM demonstrates state-of-the-art performance on METRe with respect to existing flow and diffusion models. Best is **bolded** and second-best is underlined.

Method	METRe (%) ↑	cRMSE ↓
DiffCSP	53.14	0.279
FlowMM	65.18	0.226
MCFlow	70.70	0.195
OMatG	70.50	0.187
Crystalite	70.87	**0.174**
OMatG-flash$_{16}$	68.56	0.250
OMatG-flash$_{16}$-RAM	**71.10**	0.221

Despite high METRe and match rates, the cRMSE and RMSE between ground truth polymorphs and samples produced by OMatG-flash is systematically high for both polymorph-aware and one-to-one matching indicating that, while many structures can be successfully matched to a test set structure, their error relative to that reference is higher than that of a diffusion- or flow-based generative model. We argue that this can be attributed to compounding errors in the model exacerbated by performing very few inference steps, a trend that has been observed in few-step inference for physical science applications Woo et al. (2026). We also highlight that post-training greatly improves cRMSE and k-inference RMSE which suggests that reinforcement with respect to domain-specific rewards can correct for the heightened sensitivity to model error that standard flow maps incur to perform amortized inference. Furthermore, assuming that production inference for materials discovery is always followed by MLIP relaxation as in DNG with LeMat-GenBench, we argue that generating structures within the appropriate basin of attraction is of more immediate concern than completely resolving structure.

DNG Results OMatG-flash exhibits high-quality performance that remains robust with as few as four model calls per inference. The novelty of structures proposed by OMatG-flash is high on average, trailing only MatterGen (Zeni et al., 2024) at 4 NFE while exceeding MatterGen's stability

rate by a large margin. The S.U.N. and combined S.U.N. and M.S.U.N. rates of structures generated by OMatG-flash rival the state of the art at a fraction of the inference cost. Except for DiffCSP, competing results given in Table 3 are among those which are denoted as prerelaxed on LeMat-GenBench to provide a fair comparison between OMatG-flash and existing models.

We highlight that OMatG-flash-RAM is not benchmarked on this task. As identified in previous work, the S.U.N. and M.S.U.N. metrics are capable of being "hacked" by the naive application of RL Chen et al. (2025); Park & Walsh (2025); Hoellmer & Martiniani (2026). A model can quickly learn that certain compositions and classes of structures are broadly stable or metastable, focusing generation on repetitive materials to maximize the reward. For CSP this is less of an issue, as every proposed material is conditioned on a fixed chemical composition which imposes structure on the energy surface that is difficult to exploit. We argue that S.U.N. and M.S.U.N. are more relevant for assessing performance during pretraining but that post-training reinforcement should be targeted to chemical and structural features to target a specific physical property as opposed to stability or novelty in the broadest sense. We do not consider RAM post-training for DNG in this work and leave post-training of OMatG-flash and reward design to this end as future work.

5.2 Pareto-Optimal Materials Generation

Flow and diffusion-based models for materials design have demonstrated remarkable capability for sampling stable and novel materials, but the primary advantage of OMatG-flash is scalability at inference time. OMatG-flash demonstrates performance competitive with the state-of-the-art but with massively improved inference-time throughput. In Figure 1 we construct a Pareto frontier of performance vs. inference speed for the DNG task to illustrate this. OMatG-flash exhibits Pareto-optimal performance with respect to existing flow and diffusion models, surpassing all except Crystalite in terms of performance on LeMat-GenBench with at least an order of magnitude speedup over existing models. We argue that, while improvement in these raw performance indicators is crucial, advancement along other axes such as inference speed and cost is necessary at scale. OMatG-flash sets the state-of-the-art in balancing rapid inference with high-quality generation for CSP and DNG.

5.3 Limitations and Future Work

A key limitation of OMatG-flash is the high reported RMSE between generated samples from the base model and ground truth data. This, coupled with high METRe rates, indicates that, while OMatG-flash is quite capable of isolating stable crystalline motifs, it struggles to resolve fine-grained structure. Improving the quality of the base model is therefore of immediate importance.

A clear future direction is to better understand and stabilize RAM post-training for the flow map setting (Appendix E). A separate future direction for OMatG-flash is as an engine for inference time steering where the aim is to take a pretrained generative model and guide it towards desired rewards without changing the weights themselves Wu et al. (2024); Skreta et al. (2025). Recently, there has been increased interest in using flow maps in conjunction with diffusion or flow-based generative models to bias inference towards regions of feature space with high rewards Potaptchik et al. (2026); Holderrieth et al. (2026); McCallum et al. (2026). OMatG-flash could be extended for such a use case.

6 Conclusion

In this work we introduced OMatG-flash as an all-atom flow map for materials generation and showcased its capability for rapid inference of high-quality inorganic material candidates. Furthermore, we demonstrated post-training fine-tuning of OMatG-flash based on desired rewards. Benchmarking on Materials Project data, we show that OMatG-flash is a Pareto-optimal materials generator for CSP and DNG and pushes the boundaries of inference-time speed while maintaining raw performance on par with the state-of-the-art. As materials discovery shifts toward fully automated agentic workflows and accelerated postprocessing, generation time can no longer be treated as a secondary metric. When one model can generate thousands of candidates in the time another generates one, small differences in per-sample accuracy are overwhelmed by the difference in throughput.

AI USE STATEMENT

In this work, we used generative AI tools to design or provide feedback on research methodology or experiments, implement methods, and to clean or reformat data. We have not used generative AI tools for proposing or refining hypotheses, supporting qualitative and thematic data analysis, interpreting results, or helping develop theoretical models or conceptual frameworks. Formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, assisting with translation, are not applicable to this work. Additionally, we used generative AI tools for creating and modifying scientific figures or images and editing for improved readability. We have reviewed all AI-assisted work. All AI-generated code was reviewed and rigorously tested to ensure faithfulness to the authors' intent. All methods implemented with AI-assistance were either conceived by the authors or drawn from existing literature. The only data transformation done by an LLM was Niggli reduction carried out using well-established scientific software, ASE (Hjorth Larsen et al., 2017), with parameters and specifications checked by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our work concerns generative AI for proposing novel materials and introduces a new tool for materials generation. We do not foresee any direct harm arising from this work.

REPRODUCIBILITY STATEMENT

In an effort to contribute a reproducible work we have taken care to include details regarding model training (Section 3), architecture (Appendix C), and inference in addition to providing detailed discussion on the chosen representation (Appendix A) of the data considered for generative modeling. Furthermore, we include a table of relevant hyperparameters for pretraining, post-training, and inference (F.3) in Table 4. We will make all code associated with the work open-source for reproducibility and extension of the work.

REFERENCES

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic Interpolants: A Unifying Framework for Flows and Diffusions, November 2023.

Brian D. O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, May 1982. ISSN 0304-4149. doi: 10.1016/0304-4149(82)90051-5.

Luis M. Antunes, Keith T. Butler, and Ricardo Grau-Crespo. Crystal structure generation with autoregressive large language modeling. *Nature Communications*, 15(1):10570, December 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-54639-7.

Christopher J. Bartel, Amalie Trewartha, Qi Wang, Alexander Dunn, Anubhav Jain, and Gerbrand Ceder. A critical examination of compound stability predictions from machine-learned formation energies. *npj Computational Materials*, 6(1):97, July 2020. ISSN 2057-3960. doi: 10.1038/s41524-020-00362-y.

Ilyes Batatia, Philipp Benner, Yuan Chiang, Alin M. Elena, Dávid P. Kovács, Janosh Riebesell, Xavier R. Advincula, Mark Asta, William J. Baldwin, Noam Bernstein, Arghya Bhowmik, Samuel M. Blau, Vlad Cărare, James P. Darby, Sandip De, Flaviano Della Pia, Volker L. Deringer, Rokas Elijošius, Zakariya El-Machachi, Edvin Fako, Andrea C. Ferrari, Annalena Genreith-Schriever, Janine George, Rhys E. A. Goodall, Clare P. Grey, Shuang Han, Will Handley, Hendrik H. Heenen, Kersti Hermansson, Christian Holm, Jad Jaafar, Stephan Hofmann, Konstantin S. Jakob, Hyunwook Jung, Venkat Kapil, Aaron D. Kaplan, Nima Karimitari, Namu Kroupa, Jolla Kullgren, Matthew C. Kuner, Domantas Kuryla, Guoda Liepuoniute, Johannes T. Margraf, Ioan-Bogdan Magdău, Angelos Michaelides, J. Harry Moore, Aakash A. Naik, Samuel P. Niblett, Sam Walton Norwood, Niamh O'Neill, Christoph Ortner, Kristin A. Persson, Karsten Reuter, Andrew S. Rosen, Lars L. Schaaf, Christoph Schran, Eric Sivonxay, Tamás K. Stenczel, Viktor Svahn, Christopher Sutton, Cas van der Oord, Eszter Varga-Umbrich, Tejs Vegge, Martin

Vondrák, Yangshuai Wang, William C. Witt, Fabian Zills, and Gábor Csányi. A foundation model for atomistic materials chemistry. 2023.

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

Andreas Bergmeister, Stefanie Jegelka, Nikolas Nüsken, Carles Domingo-Enrich, and Jakiw Pidstrigach. Reinforce Adjoint Matching: Scaling RL Post-Training of Diffusion and Flow-Matching Models, May 2026. URL http://arxiv.org/abs/2605.10759. arXiv:2605.10759 [cs.LG].

Siddharth Betala, Samuel P. Gleason, Ali Ramlaoui, Andy Xu, Georgia Channing, Daniel Levy, Clémentine Fourrier, Nikita Kazeev, Chaitanya K. Joshi, Sékou-Oumar Kaba, Félix Therrien, Alex Hernandez-Garcia, Rocío Mercado, N. M. Anoop Krishnan, and Alexandre Duval. LeMat-GenBench: A Unified Evaluation Framework for Crystal Generative Models, January 2026.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training Diffusion Models with Reinforcement Learning, January 2024. URL http://arxiv.org/abs/2305.13301. arXiv:2305.13301 [cs.LG].

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. Flow Map Matching, June 2024.

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation, October 2025.

Cyprien Bone, Matthew Walker, Bradley A. A. Martin, Kuangdai Leng, Luis M. Antunes, Ricardo Grau-Crespo, Amil Aligayev, Javier Dominguez, and Keith T. Butler. Discovery and recovery of crystalline materials with property-conditioned transformers, June 2026.

Peter G. Boyd, Arunraj Chidambaram, Enrique García-Díez, Christopher P. Ireland, Thomas D. Daff, Richard Bounds, Andrzej Gładysiak, Pascal Schouwink, Seyed Mohamad Moosavi, M. Mercedes Maroto-Valer, Jeffrey A. Reimer, Jorge A. R. Navarro, Tom K. Woo, Susana Garcia, Kyriakos C. Stylianou, and Berend Smit. Data-driven design of metal–organic frameworks for wet flue gas CO2 capture. *Nature*, 576(7786):253–256, December 2019. ISSN 1476-4687. doi: 10.1038/s41586-019-1798-7. URL https://www.nature.com/articles/s41586-019-1798-7.

Théo Cavignac, Jonathan Schmidt, Pierre-Paul De Breuck, Antoine Loew, Tiago F. T. Cerqueira, Hai-Chen Wang, Anton Bochkarev, Yury Lysogorskiy, Aldo H. Romero, Ralf Drautz, Silvana Botti, and Miguel A. L. Marques. AI-driven expansion and application of the Alexandria database. *Journal of Physics: Materials*, 9(2):025014, 2026. doi: 10.1088/2515-7639/ae6620. URL https://doi.org/10.1088/2515-7639/ae6620.

Rees Chang, Angela Pak, Alex Guerra, Ni Zhan, Nick Richardson, Elif Ertekin, and Ryan P. Adams. Space Group Equivariant Crystal Diffusion, October 2025.

Rees Chang, Andrew Novick, Ryan P. Adams, and Elif Ertekin. SLayerGen: A Crystal Generative Model for all Space and Layer Groups, May 2026.

Junwu Chen, Jeff Guo, Edvin Fako, and Philippe Schwaller. Accelerating inverse materials design using generative diffusion models with reinforcement learning, November 2025.

Wei Chen, Janine George, Joel B. Varley, Gian-Marco Rignanese, and Geoffroy Hautier. High-throughput computational discovery of In2Mn2O7 as a high Curie temperature ferromagnetic semiconductor for spintronics. *npj Computational Materials*, 5(1):72, July 2019. ISSN 2057-3960. doi: 10.1038/s41524-019-0208-x. URL https://www.nature.com/articles/s41524-019-0208-x.

Orion Cohen, Janosh Riebesell, Rhys Goodall, Adeesh Kolluru, Stefano Falletta, Joseph Krause, Jorge Colindres, Gerbrand Ceder, and Abhijeet S. Gangan. TorchSim: An efficient atomistic simulation engine in PyTorch, August 2025.

Connor W. Coley, Dale A. Thomas, Justin A. M. Lummiss, Jonathan N. Jaworski, Christopher P. Breen, Victor Schultz, Travis Hart, Joshua S. Fishman, Luke Rogers, Hanyu Gao, Robert W. Hicklin, Pieter P. Plehiers, Joshua Byington, John S. Piotti, William H. Green, A. John Hart, Timothy F. Jamison, and Klavs F. Jensen. A robotic platform for flow synthesis of organic compounds informed by AI planning. *Science*, 365(6453):eaax1566, August 2019. doi: 10.1126/science.aax1566.

François Cornet, Federico Bergamin, Arghya Bhowmik, Juan Maria Garcia Lastra, Jes Frellsen, and Mikkel N. Schmidt. Kinetic Langevin Diffusion for Crystalline Materials Generation, July 2025.

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

Oscar Davis, Michael S. Albergo, Nicholas M. Boffi, Michael M. Bronstein, and Avishek Joey Bose. Generalised Flow Maps for Few-Step Generative Modelling on Riemannian Manifolds, February 2026.

Pierre-Paul De Breuck, Hai-Chen Wang, Gian-Marco Rignanese, Silvana Botti, and Miguel A. L. Marques. Generative AI for crystal structures: a review. *npj Computational Materials*, 11(1): 370, December 2025. ISSN 2057-3960. doi: 10.1038/s41524-025-01881-2. URL https://www.nature.com/articles/s41524-025-01881-2.

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky T. Q. Chen. Adjoint Matching: Fine-tuning Flow and Diffusion Generative Models with Memoryless Stochastic Optimal Control, January 2025.

Louis Grenioux, Leonardo Galliano, Ludovic Berthier, Giulio Biroli, and Marylou Gabrié. Boltzmann generators for amorphous particle systems, July 2026.

Geoffroy Hautier, Chris Fischer, Virginie Ehrlacher, Anubhav Jain, and Gerbrand Ceder. Data Mined Ionic Substitutions for the Discovery of New Compounds. *Inorganic Chemistry*, 50(2):656–663, December 2010. ISSN 0020-1669. doi: 10.1021/ic102031h.

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

Philipp Hoellmer and Stefano Martiniani. Open Materials Generation with Inference-Time Reinforcement Learning, June 2026.

Philipp Hoellmer, Thomas Egg, Maya M. Martirossyan, Eric Fuemmeler, Zeren Shui, Amit Gupta, Pawan Prakash, Adrian Roitberg, Mingjie Liu, George Karypis, Mark Transtrum, Richard G. Hennig, Ellad B. Tadmor, and Stefano Martiniani. Open Materials Generation with Stochastic Interpolants, July 2025.

Peter Holderrieth, Douglas Chen, Luca Eyring, Ishin Shah, Giri Anantharaman, Yutong He, Zeynep Akata, Tommi Jaakkola, Nicholas Matthew Boffi, and Max Simchowitz. Diamond Maps: Efficient Reward Alignment via Stochastic Flow Maps, May 2026.

Aapo Hyvärinen. Estimation of Non-Normalized Statistical Models by Score Matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005. ISSN 1533-7928.

Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, and Kristin a. Persson. The Materials Project: A materials genome approach to accelerating materials innovation. *APL Materials*, 1(1):011002, 2013. ISSN 2166532X. doi: 10.1063/1.4812323. URL http://link.aip.org/link/AMPADS/v1/i1/p011002/s1&Agg=doi.

Rui Jiao, Wenbing Huang, Peijia Lin, Jiaqi Han, Pin Chen, Yutong Lu, and Yang Liu. Crystal Structure Prediction by Joint Equivariant Diffusion, March 2024.

Chaitanya K. Joshi, Xiang Fu, Yi-Lun Liao, Vahe Gharakhanyan, Benjamin Kurt Miller, Anuroop Sriram, and Zachary W. Ulissi. All-atom Diffusion Transformers: Unified generative modelling of molecules and materials, May 2025.

Nikita Kazeev, Wei Nong, Ignat Romanov, Ruiming Zhu, Andrey Ustyuzhanin, Shuya Yamazaki, and Kedar Hippalgaonkar. Wyckoff Transformer: Generation of Symmetric Crystals, June 2025.

Chanhyuk Lee, Jaehoon Yoo, Manan Agarwal, Sheel Shah, Jerry Huang, Aditi Raghunathan, Seunghoon Hong, Nicholas M. Boffi, and Jinwoo Kim. Flow Map Language Models: One-step Language Modeling via Continuous Denoising, May 2026. URL http://arxiv.org/abs/2602.16813. arXiv:2602.16813 [cs.CL].

Daniel Levy, Siba Smarak Panigrahi, Sékou-Oumar Kaba, Qiang Zhu, Kin Long Kelvin Lee, Mikhail Galkin, Santiago Miret, and Siamak Ravanbakhsh. SymmCD: Symmetry-Preserving Crystal Generation with Diffusion Models. https://arxiv.org/abs/2502.03638v3, February 2025.

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

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling, February 2023.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-GRPO: Training Flow Matching Models via Online RL, October 2025.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow, September 2022.

Alston Lo, Luka Mucko, Austin H. Cheng, Andy Cai, Alastair J. A. Price, Wojciech Matusik, and Alán Aspuru-Guzik. Fast Organic Crystal Structure Prediction with Unit Cell Flow Matching, June 2026. URL http://arxiv.org/abs/2606.03199. arXiv:2606.03199 [cs.LG].

Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, and Saining Xie. SiT: Exploring Flow and Diffusion-based Generative Models with Scalable Interpolant Transformers, September 2024. URL http://arxiv.org/abs/2401.08740. arXiv:2401.08740 [cs.CV].

B. P. MacLeod, F. G. L. Parlane, T. D. Morrissey, F. Häse, L. M. Roch, K. E. Dettelbach, R. Moreira, L. P. E. Yunker, M. B. Rooney, J. R. Deeth, V. Lai, G. J. Ng, H. Situ, R. H. Zhang, M. S. Elliott, T. H. Haley, D. J. Dvorak, A. Aspuru-Guzik, J. E. Hein, and C. P. Berlinguette. Self-driving laboratory for accelerated discovery of thin-film materials. *Science Advances*, 6(20):eaaz8867, May 2020. doi: 10.1126/sciadv.aaz8867.

Maya M. Martirossyan, Thomas Egg, Philipp Hoellmer, George Karypis, Mark Transtrum, Adrian Roitberg, Mingjie Liu, Richard G. Hennig, Ellad B. Tadmor, and Stefano Martiniani. All that structure matches does not glitter, September 2025.

Sam McCallum, Zander W. Blasingame, Timothy Herschell, Niklas Rindtorff, Alexander Tong, and James Foster. Strong Stochastic Flow Maps, May 2026.

Amil Merchant, Simon Batzner, Samuel S. Schoenholz, Muratahan Aykol, Gowoon Cheon, and Ekin Dogus Cubuk. Scaling deep learning for materials discovery. *Nature*, 624(7990):80–85, December 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06735-9.

Benjamin Kurt Miller, Ricky T. Q. Chen, Anuroop Sriram, and Brandon M. Wood. FlowMM: Generating Materials with Riemannian Flow Matching, June 2024.

Mark Neumann, James Gin, Benjamin Rhodes, Steven Bennett, Zhiyi Li, Hitarth Choubisa, Arthur Hussey, and Jonathan Godwin. Orb: A Fast, Scalable Neural Network Potential, October 2024.

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

Hyunsoo Park and Aron Walsh. Guiding Generative Models to Uncover Diverse and Novel Crystals via Reinforcement Learning, November 2025.

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers, March 2023.

Chris J. Pickard and R. J. Needs. Ab initio Random Structure Searching. *Journal of Physics: Condensed Matter*, 23(5):053201, February 2011. ISSN 0953-8984, 1361-648X. doi: 10.1088/0953-8984/23/5/053201.

Kinga Pielichowska and Krzysztof Pielichowski. Phase change materials for thermal energy storage. *Progress in Materials Science*, 65:67–123, August 2014. ISSN 0079-6425. doi: 10.1016/j.pmatsci.2014.03.005.

Peter Potaptchik, Adhi Saravanan, Abbas Mammadov, Alvaro Prat, Michael S. Albergo, and Yee Whye Teh. Meta Flow Maps enable scalable reward alignment, June 2026.

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

Pawan Prakash, Jason B. Gibson, Zhongwei Li, Gabriele Di Gianluca, Juan Esquivel, Eric Fuemmeler, Benjamin Geisler, Jung Soo Kim, Adrian Roitberg, Ellad B. Tadmor, Mingjie Liu, Stefano Martiniani, Gregory R. Stewart, James J. Hamlin, Peter J. Hirschfeld, and Richard G. Hennig. Guided diffusion for the discovery of new superconductors. *npj Computational Materials*, May 2026. ISSN 2057-3960. doi: 10.1038/s41524-026-02117-7.

Omri Puny, Yaron Lipman, and Benjamin Kurt Miller. Space Group Conditional Flow Matching, September 2025.

Winfried Ripken, Michael Plainer, Gregor Lied, Thorben Frank, Oliver T. Unke, Stefan Chmiela, Frank Noé, and Klaus-Robert Müller. Learning Hamiltonian Flow Maps: Mean Flow Consistency for Large-Timestep Molecular Dynamics, June 2026. URL http://arxiv.org/abs/2601.22123. arXiv:2601.22123 [cs.LG].

Gianluca Scarpellini, Ron Shprints, Peter Holderrieth, Juno Nam, Pranav Murugan, Rafael Gómez-Bombarelli, Tommi Jaakkola, Maruan Al-Shedivat, Nicholas Matthew Boffi, and Avishek Joey Bose. Few-step Cofolding with All-Atom Flow Maps, June 2026.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, August 2017.

Kiyoung Seong, Nayoung Kim, and Sungsoo Ahn. Discovering Crystal Structure Prediction Algorithms with an AI Co-Scientist, June 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models, April 2024. URL http://arxiv.org/abs/2402.03300. arXiv:2402.03300 [cs].

Martin Siron, Inel Djafar, Ali Ramlaoui, Etienne du Fayette, Amandine Rossello, Edvin Fako, Matthew McDermott, Felix Therrien, Luis Barroso-Luque, Flaviu Cipcigan, Philippe Schwaller, Thomas Wolf, and Alexandre Duval. LeMat-Bulk: Aggregating, and de-duplicating quantum chemistry materials databases, November 2025.

Marta Skreta, Tara Akhound-Sadegh, Viktor Ohanesian, Roberto Bondesan, Alán Aspuru-Guzik, Arnaud Doucet, Rob Brekelmans, Alexander Tong, and Kirill Neklyudov. Feynman-Kac Correctors in Diffusion: Annealing, Guidance, and Product of Experts, June 2025.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep Unsupervised Learning using Nonequilibrium Thermodynamics, November 2015.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations, February 2021.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency Models, May 2023.

G. E. Uhlenbeck and L. S. Ornstein. On the Theory of the Brownian Motion. *Physical Review*, 36(5):823–841, September 1930. doi: 10.1103/PhysRev.36.823. URL https://link.aps.org/doi/10.1103/PhysRev.36.823.

Tin Hadži Veljković, Joshua Rosenthal, Ivor Lončarić, and Jan-Willem van de Meent. Crystalite: A Lightweight Transformer for Efficient Crystal Modeling, July 2026.

Dongyeop Woo, Marta Skreta, Seonghyun Park, Kirill Neklyudov, and Sungsoo Ahn. Riemannian MeanFlow, May 2026.

Brandon M. Wood, Misko Dzamba, Xiang Fu, Meng Gao, Muhammed Shuaibi, Luis Barroso-Luque, Kareem Abdelmaqsoud, Vahe Gharakhanyan, John R. Kitchin, Daniel S. Levine, Kyle Michel, Anuroop Sriram, Taco Cohen, Abhishek Das, Ammar Rizvi, Sushree Jagriti Sahoo, Zachary W. Ulissi, and C. Lawrence Zitnick. UMA: A Family of Universal Models for Atoms, June 2025.

Luhuan Wu, Brian L. Trippe, Christian A. Naesseth, David M. Blei, and John P. Cunningham. Practical and Asymptotically Exact Conditional Sampling in Diffusion Models, November 2024.

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

Andy Xu, Rohan Desai, Larry Wang, Ethan Ritz, and Gabriel Hope. PLaID++: A Preference Aligned Language Model for Targeted Inorganic Materials Design, June 2026.

Jaehoon Yoo, Wonjung Kim, Floor Eijkelboom, Chanhyuk Lee, Nicholas M. Boffi, Seunghoon Hong, and Jinwoo Kim. Self-conditioned Flow Map Language Models via Fixed-point Flows, July 2026.

D. Zagorac, H. Müller, S. Ruehl, J. Zagorac, and S. Rehme. Recent developments in the Inorganic Crystal Structure Database: Theoretical crystal structure data and related features. *Journal of Applied Crystallography*, 52(5):918–925, 2019. ISSN 1600-5767. doi: 10.1107/S160057671900997X.

Cheng Zeng, Harry W. Sullivan, Thomas Egg, Maya M. Martirossyan, Philipp Höllmer, Jirui Jin, Richard G. Hennig, Adrian Roitberg, Stefano Martiniani, Ellad B. Tadmor, and Mingjie Liu. MolCrystalFlow: Molecular Crystal Structure Prediction via Flow Matching, March 2026. URL http://arxiv.org/abs/2602.16020. arXiv:2602.16020 [cs.LG].

Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Sasha Shysheya, Jonathan Crabbé, Lixin Sun, Jake Smith, Bichlien Nguyen, Hannes Schulz, Sarah Lewis, Chin-Wei Huang, Ziheng Lu, Yichi Zhou, Han Yang, Hongxia Hao, Jielan Li, Ryota Tomioka, and Tian Xie. MatterGen: A generative model for inorganic materials design, January 2024.

A Unit Cell Representation

As introduced in Section 3, we consider the materials proposal problem as inference on crystalline unit cells $\mathbf{c} = (\mathbf{A}, \mathbf{F}, \mathbf{y})$. The atomic descriptor matrix $\mathbf{A}$ is obtained by passing the raw atom types $\mathbf{a} \in \mathbb{Z}_+^N$ through a chosen featurization map (Appendix C). $\mathbf{F}$ denotes fractional coordinates derived from 3-D Cartesian atomic coordinates $\mathbf{X} \in \mathbb{R}^{N\times 3}$ using the unit cell $\mathbf{L}$

$$\mathbf{F} = \mathbf{X}\mathbf{L}^{-1}. \tag{13}$$

Given any valid cell matrix $\mathbf{L}' \in \mathrm{GL}^+(3, \mathbb{R})$, we first construct its metric tensor

$$\mathbf{G} = \mathbf{L}'\mathbf{L}'^{\top}.$$

We then compute the lower-triangular Cholesky factor $\mathbf{L}$ such that

$$\mathbf{G} = \mathbf{L}\mathbf{L}^{\top}.$$

The corresponding six-dimensional latent representation $\mathbf{y}$ is then defined by

$$\mathbf{L}(\mathbf{y}) = \begin{bmatrix} e^{y_1} & 0 & 0 \\ y_2 & e^{y_3} & 0 \\ y_4 & y_5 & e^{y_6} \end{bmatrix},$$

or equivalently,

$$\mathbf{y}(\mathbf{L}) = [\log L_{11},\, L_{21},\, \log L_{22},\, L_{31},\, L_{32},\, \log L_{33}] \in \mathbb{R}^6.$$

Manifold Operations for Unit Cell Transport We treat the generation of all components of the crystal unit cell as taking place in continuous space. For $\mathbf{y}$ and $\mathbf{A}$ we operate in Euclidean space with simple Riemannian $\exp$ and $\log$ maps which, for elements x and y, are given by

$$\exp_x(y) = x + y \quad \text{and} \quad \log_x(y) = y - x. \tag{14}$$

For $\mathbf{F}$ we consider generation on the periodic flat torus, $\mathbb{T}^{N\times 3}_{[0,1)} := (\mathbb{R}/\mathbb{Z})^{N\times 3}$ with

$$\exp_x(y) = \texttt{wrap}(x + y) \quad \text{and} \quad \log_x(y) = \texttt{wrap}(y - x + 0.5) - 0.5 \tag{15}$$

and wrap operation as $\texttt{wrap}(z) = z \bmod 1.0$. These operations allow us to write a Riemannian stochastic interpolant for each component of the crystalline unit cell $\mathbf{c}$.

B Generative Modeling as Dynamical Transport

B.1 Diffusions

Diffusion models consider bridging a Gaussian base distribution ρ_T with a target distribution ρ_0 by a stochastic process whose time evolution is given by a forward SDE

$$\mathrm{d}x_t = f_t(x_t)\mathrm{d}t + \sigma_t \mathrm{d}\overrightarrow{W}_t. \tag{16}$$

Here f_t is a linear drift $f_t(x) = A_t x + b_t$, σ_t is a chosen diffusion coefficient, and $\mathrm{d}\overrightarrow{W}_t$ is an infinitesimal Wiener increment where the arrow indicates that this process moves forward in time. This SDE gradually applies noise to data, approaching an isotropic Gaussian as T grows to infinity. The objective in score-based diffusion models is to learn a reverse-time SDE which generates clean data from noise which, subject to mild conditions on f_t and σ_t, takes the form Anderson (1982); Sohl-Dickstein et al. (2015); Song et al. (2021)

$$\mathrm{d}x_t = \left[f_t(x_t) - \sigma_t^2 \nabla_{x_t} \log \rho_t(x_t)\right] \mathrm{d}t + \sigma_t \mathrm{d}\overleftarrow{W}_t. \tag{17}$$

This is generally done by minimizing a score matching objective Hyvärinen (2005). If we parameterize a neural network s_t^θ approximating the score $s_t^\theta \approx \nabla_x \log \rho_t$ then we can write the objective

$$\mathcal{L}_{\mathrm{SM}}(\theta) = \int_0^T \mathbb{E}\left[\left\| s_t^\theta(x_t) - \nabla_{x_t} \log \rho_t(x_t|x_0) \right\|^2\right] \mathrm{d}t \tag{18}$$

Since the forward SDE has an affine-linear drift and a state-independent diffusion coefficient, its transition kernel $\rho_t(x_t|x_0)$ is Gaussian and given by the usual solution to the SDE in equation 16 Uhlenbeck & Ornstein (1930). With access to an estimator for the score, equation 17 can be integrated backwards in time to generate data from independent realizations of ρ_T.

B.2 FLOWS

Following the success of diffusion models, methods have been formulated to learn a deterministic transport between noise and clean data. In this flow-based generative modeling framework a coupling between a base distribution ρ_0 and a target ρ_1 is given by a velocity field Lipman et al. (2023); Albergo et al. (2023); Liu et al. (2022).[1] The typical approach is to bridge noise and data with an interpolant as in Section 3,

$$x_t = \alpha_t x_0 + \beta_t x_1, \tag{19}$$

where x_1 is sampled from the target and x_0 is sampled from noise. Both α_t and β_t are scheduling functions which determine how data is noised in time.

Interpolation between noise and data allows for the construction of the flow matching algorithm which aims to learn an instantaneous velocity field v_t. The desired velocity is that which solves a transport equation

$$\frac{\partial \rho_t}{\partial t} + \nabla \cdot (v_t \rho_t) = 0. \tag{20}$$

This transport equation describes the evolution of a time-dependent density from $t = 0$ to $t = 1$. It can be shown that this velocity can be written in terms of the interpolant in equation 19

$$v_t(x) = \mathbb{E}[\dot{x}_t | x_t = x] \tag{21}$$

We approximate this velocity with a neural network v_t^θ by minimizing the conditional flow matching loss

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \int_0^1 \mathbb{E}\left[\left\| v_t^\theta(x_t) - \dot{x}_t \right\|^2\right] \mathrm{d}t \tag{22}$$

where the expectation is taken over (x_0, x_1) pairs. Access to an approximate v_t allows for the generation of new data samples through integration of an ODE forwards in time

$$\frac{\mathrm{d}x}{\mathrm{d}t} = v_t, \quad \text{s.t.} \quad x_0 \sim \rho_0. \tag{23}$$

Flows and diffusions are closely related paradigms for generative modeling Albergo et al. (2023). If ρ_0 is a standard Gaussian, it is possible to simulate data via a diffusion SDE similar to equation 17 given access to a flow velocity field v_t using the following relation

$$\nabla_x \log \rho_t(x) = \alpha_t^{-1} \frac{\beta_t v_t(x) - \dot{\beta}_t x}{\dot{\beta}_t \alpha_t - \beta_t \dot{\alpha}_t}. \tag{24}$$

This relationship is used for deriving Flow-GRPO Liu et al. (2025). The relationship between the score and the velocity is similarly used in the derivation of RAM (Appendix E).

B.3 FLOW MAPS

The success of flows and diffusions has inspired exploration into flow maps and consistency models Song et al. (2023); Boffi et al. (2024; 2025). Given a base and target distribution bridged by equation 20 a flow map is a two-time map $X_{s,u}$ which satisfies

$$X_{s,u}(x_s) = x_u \tag{25}$$

where $s, u \in [0, 1]$.

A flow map must satisfy a tangent and consistency condition. The tangent condition enforces that, in the instantaneous limit, the flow map velocity matches the velocity in equation 23

[1] Literature on flows and diffusions is often inconsistent in terms of which direction in time points from noise to data. In diffusion model literature, noise is also attributed to a steady-state realized at $T = \infty$ whereas $T = 0$ is a clean initial condition that is progressively noised. In flow-based models, $t = 0$ often corresponds to noise and $t = 1$ to data. We choose this convention throughout the paper as our method is more directly related to flow models.

$$\dot{X}_{s,s}(x_s) := \frac{\partial}{\partial u} X_{s,u}(x_s)\Big|_{u=s} = v_s(x_s). \tag{26}$$

A flow map must also be consistent through time. There are three conditions that can be used to enforce consistency:

- **Lagrangian**
$$\frac{\partial X_{s,u}(x_s)}{\partial u} = v_u(X_{s,u}(x_s)) \tag{27}$$
- **Eulerian**
$$\frac{\partial X_{s,u}(x_s)}{\partial s} + \nabla X_{s,u}(x_s) v_s(x_s) = 0 \tag{28}$$
- **Semigroup**
$$X_{v,w}(X_{s,v}(x_s)) = X_{s,w}(x_s) \tag{29}$$

where $s < v < w$. Flow maps have been successfully generalized for generative modeling on Riemannian manifolds Davis et al. (2026); Woo et al. (2026). As outlined in Section 3 we employ Riemannian MeanFlow with a semigroup consistency objective for training OMatG-flash. Following this prescription (Woo et al., 2026), we parameterize OMatG-flash as a two-time map

$$X_{s,u}(x_s) = \exp_{x_s}\left((u-s) v_{s,u}(x_s)\right) \tag{30}$$

where $\exp_x y$ is the exponential map associated with a manifold $\mathcal{M}$ and $s < u$. We find strong performance by parameterizing the average velocity field $v_{s,u}$ with endpoint prediction. Specifically, given x_s and an estimator of the endpoint $x_1(x_s, s, u)$ we write the two-time velocity

$$v_{s,u}(x_s) = \frac{\log_{x_s} x_1(x_s, s, u)}{1-s} \tag{31}$$

where $\log_x y$ is the logarithm map on $\mathcal{M}$. Composing these two equations above we arrive at the parameterization of OMatG-flash in equation 8. For both tangency and consistency losses we employ weighting functions to stabilize the loss $\gamma_t = \frac{1-t}{\max(1-t,\epsilon)}$ and $\gamma_{s,w} = \frac{\gamma_s}{(w-s)^2}$. We set $\epsilon = 0.1$.

C OMatG-flash Architecture

Atom Type Encoding OMatG-flash applies Crystalite's subatomic tokenization scheme (Veljković et al., 2026). For atomic number Z_i, Crystalite constructs the 34-dimensional descriptor

$$\mathbf{d}(Z_i) = \left[\mathbf{e}^{(7)}_{r(Z_i)} \,\middle\|\, \mathbf{e}^{(19)}_{g(Z_i)} \,\middle\|\, \mathbf{e}^{(4)}_{b(Z_i)} \,\middle\|\, \left(\tfrac{n_s}{2}, \tfrac{n_p}{6}, \tfrac{n_d}{10}, \tfrac{n_f}{14}\right)\right], \tag{32}$$

where $\mathbf{e}^{(k)}$ denotes a k-dimensional one-hot encoding, r, g, and b denote the element's period, group, and block, and (n_s, n_p, n_d, n_f) are its ground-state valence-shell occupancies. After standardization, group weighting, PCA projection, and normalization, each descriptor is mapped to $\mathbf{a}(Z_i) \in \mathbb{R}^{d_A}$ with $d_A = 24$. The atomic descriptor matrix is formed by stacking these vectors:

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}(Z_1)^\top \\ \vdots \\ \mathbf{a}(Z_N)^\top \end{bmatrix} \in \mathbb{R}^{N \times d_A}. \tag{33}$$

This representation effectively describes the space of elements and circumvents the need for a discrete diffusion because the state is part of a continuous manifold.

Atom Type Decoding After integration is complete, each generated atomic descriptor is snapped to the closest valid element descriptor in PCA space. Specifically, for a generated descriptor $\hat{\mathbf{A}}_{i,1}$, we assign the atomic number

$$\hat{Z}_i = \underset{Z \in \{1,\ldots,100\}}{\arg\max} \frac{\left\langle \hat{\mathbf{A}}_{i,1}, \mathbf{a}(Z) \right\rangle}{\left\| \hat{\mathbf{A}}_{i,1} \right\|_2 \left\| \mathbf{a}(Z) \right\|_2}. \tag{34}$$

Decoding is just a nearest-neighbor projection under cosine similarity to the set of valid PCA-projected element descriptors.

Neural Network Architecture We parameterize OMatG-flash as an all-atom transformer, again building off of the Crystalite architecture to do so Veljković et al. (2026).

The atoms in a unit cell $\mathbf{c}$ are tokenized into N tokens, one per atom, with one additional token for the lattice. The tokenization of an atom indexed i is a linear combination of a coordinate and atom type embedding

$$\mathbf{h}_i^{\text{atom}} = E_{\text{comp}}(\mathbf{A}_i) + E_{\text{coord}}(\text{SinEmb}(\mathbf{F}_i; N_{\text{freq}}^{\text{coord}})) \tag{35}$$

where SinEmb featurizes the coordinates via a Fourier expansion,

$$\text{SinEmb}(x; N_{\text{freq}}) = [\sin(2\pi k x), \cos(2\pi k x)]_{k=1}^{N_{\text{freq}}}, \tag{36}$$

and $E_{\text{comp}}, E_{\text{coord}}$ are two-layer MLPs with a SiLU activation. The lattice token is similarly obtained by passing $\mathbf{y}$ through a two-layer MLP embedding layer

$$\mathbf{h}_{N+1}^{\text{lattice}}(\mathbf{y}) = E_{\text{lattice}}(\mathbf{y}) \tag{37}$$

yielding a sequence of tokens $\mathbf{t}^{(0)} = [\mathbf{h}_1^{\text{atom}}, \ldots, \mathbf{h}_N^{\text{atom}}, \mathbf{h}_{N+1}^{\text{lattice}}]$ which the all-atom transformer processes through layers of self-attention. The superscript indicates that this is the initial tokenized embedding before any transformer layers are applied. OMatG-flash is adapted from the diffusion transformer architecture Peebles & Xie (2023). Our conditioning vector is derived using the left and right foot times. For a source time s and a destination time u this conditioning vector is predicted via a two-layer MLP on s and $\Delta = u - s$

$$\mathbf{h}_{\text{conditioning}}(s, u) = E_{\text{cond}}\left(\text{SinEmb}(s; N_{\text{freq}}^{\text{time}}), \text{SinEmb}(\Delta; N_{\text{freq}}^{\text{time}})\right) \tag{38}$$

$\mathbf{h}_{\text{conditioning}}$ along with $\mathbf{t}^{(0)}$ are passed into the model. $\mathbf{h}_{\text{conditioning}}$ is used to predict scale and shift parameters which normalize the tokens via adaptive layer normalization (adaLN) over the course of several transformer layers.

We enable physics-informed self-attention within the model using Crystalite's geometry enhancement module. This module computes a block-diagonal bias mask, $\mathbf{B}(\mathbf{c})$, that modulates attention only between atom tokens i and j by first computing the approximate minimum-image displacement using $\Delta\mathbf{f}_{ij}(\mathbf{r}) = \mathbf{f}_j - \mathbf{f}_i + \mathbf{r}$,

$$\mathbf{r}_{ij}^{\star} = \arg\min_{\mathbf{r}\in\{-1,0,1\}^3} \Delta\mathbf{f}_{ij}(\mathbf{r})\mathbf{G}(\mathbf{y})\Delta\mathbf{f}_{ij}^{\top}(\mathbf{r}), \tag{39}$$

then

$$\Delta\mathbf{f}_{ij}^{\text{mi}} = \mathbf{f}_j - \mathbf{f}_i + \mathbf{r}_{ij}^{\star}. \tag{40}$$

The normalized minimum-image Cartesian distance becomes

$$d_{ij}^{\text{mi}} = \frac{\|\Delta\mathbf{f}_{ij}^{\text{mi}}\mathbf{L}(\mathbf{y})\|_2}{s(\mathbf{y})} \tag{41}$$

which is normalized by a cell scale divisor $s(\mathbf{y}) = (a + b + c)/3$ derived from the lattice vector lengths.

The pairwise geometry is featurized as

$$\mathbf{e}_{ij} = \left[\text{SinEmb}(\Delta\mathbf{f}_{ij}^{\text{mi}}; N_{\text{freq}}^{\text{edge}}) \,\middle\|\, \left\{\exp[-\gamma(d_{ij}^{\text{mi}} - \mu_q)^2]\right\}_{q=1}^{N_{\text{rbf}}} \,\middle\|\, \mathbf{y}^{\text{Y1}}(\mathbf{y})\right]. \tag{42}$$

Here, $N_{\text{freq}}^{\text{edge}}$ and N_{rbf} are the numbers of Fourier and radial basis features, respectively; μ_q are uniformly spaced radial centers, γ is their inverse squared spacing, and $\mathbf{y}^{\text{Y1}}(\mathbf{y})$ denotes the Y1 lattice features computed from $\mathbf{y}$.

The bias for attention head h is

$$[\mathbf{B}(\mathbf{c})]_{hij} = g_h(s)\left(-\text{softplus}(w_h)d_{ij}^{\text{mi}} + [E_{\text{edge}}(\mathbf{e}_{ij})]_h\right), \tag{43}$$

where w_h is a learned head-specific distance parameter, E_{edge} is a two-layer MLP with hidden dimension d_{edge}, and

$$g_h(s) = \text{sigmoid}\left(\text{softplus}(\alpha_h)[-\log(\max(s, \epsilon))] + \beta_h\right). \tag{44}$$

Here, s is the left-foot time supplied to the DiT/SiT backbone, α_h and β_h are learned head-specific parameters, and $\epsilon > 0$ is a numerical-stability constant. The diagonal of the edge bias and all entries involving padded atoms or the lattice token are set to zero.

The geometry bias is added directly to the scaled dot-product attention logits. For attention head h,

$$\mathrm{Attn}_h = \mathrm{softmax}\left(\frac{\mathbf{Q}_h\mathbf{K}_h^\top}{\sqrt{d_h}} + \mathbf{B}_h(\mathbf{c}) + \mathbf{M}\right)\mathbf{V}_h, \tag{45}$$

where $\mathbf{Q}_h$, $\mathbf{K}_h$, and $\mathbf{V}_h$ are the query, key, and value projections, d_h is the head dimension, and $\mathbf{M}$ is the key-padding mask, taking value $-\infty$ for padded key tokens and zero otherwise. The remainder of the architecture follows standard SiT/DiT token-based setups, using adaLN-conditioned self-attention and MLP residual blocks.

For predicting endpoints which are periodic in the domain $[0, 1)$ we parameterize the endpoint prediction of denoised fractional coordinates,

$$\hat{\mathbf{F}}_1 = \frac{\mathrm{atan2}(\hat{\mathbf{u}}_1, \hat{\mathbf{v}}_1)}{2\pi} \bmod 1.0, \tag{46}$$

Here, $\hat{\mathbf{u}}_1$ and $\hat{\mathbf{v}}_1$ are predicted by the fractional-coordinate head. We use this so-called circular prediction mode for all realizations of OMatG-flash. $\hat{\mathbf{A}}_1$ and $\hat{\mathbf{y}}_1$ are predicted by linear layer output heads.

Data Augmentation OMatG-flash is not rotation-equivariant nor is it translation invariant however, since $\mathbf{F}$ and $\mathbf{A}$ are naturally rotation invariant, we need only the lattice in handling rotations. Our choice of modeling $\mathbf{y}$ is rotation-invariant and, thus, we do not consider data augmentation for global rotations of the crystal. OMatG-flash is, however, not translation-invariant and $\mathbf{F}$ is sensitive to this. We therefore augment the data with random translations in 3-D during training.

D Hyperparameter Selection

Table 4: **OMatG-flash hyperparameters.**

Hyperparameter	Value
Architecture	
Atomic descriptor dimension d_A	24
Supported atomic numbers Z	1–100
Transformer width d	512
Transformer layers	18
Attention heads	16
Coordinate Fourier frequencies	32
Time-embedding dimension	256
Flow map objective	
Atomic weights $(\lambda_{\mathrm{T},A}, \lambda_{\mathrm{C},A})$	$(2.5, 2.5)$
Coordinate weights $(\lambda_{\mathrm{T},F}, \lambda_{\mathrm{C},F})$	$(1.0, 1.0)$
Lattice weights $(\lambda_{\mathrm{T},y}, \lambda_{\mathrm{C},y})$	$(0.05, 0.05)$
Optimization	
Batch size	256
Optimizer	AdamW
Learning rate	5×10^{-4}
AdamW coefficients (β_1, β_2)	$(0.9, 0.999)$
AdamW ϵ_{opt}	10^{-8}
Weight decay	10^{-2}
EMA decay	0.999
Gradient clipping threshold	1.0
RAM post-training	
Atomic weights $(\lambda_{\mathrm{RAM},A}, \lambda_{\mathrm{C},A})$	$(-, -)$
Coordinate weights $(\lambda_{\mathrm{RAM},F}, \lambda_{\mathrm{C},F})$	(0.520, 0.002)
Lattice weights $(\lambda_{\mathrm{RAM},y}, \lambda_{\mathrm{C},y})$	(0.074, 0.002)
RAM energy weight λ_E	0.2
RAM number groups	16
RAM group size	64
RAM noising replicas per sample	4
RAM learning rate	1.0×10^{-4}
RAM training steps	250

OMatG-flash's hyperparameters are summarized in Table 4. We minimize flow map and RAM losses per-field using the weights specified. The channel-specific tangency and consistency weights are normalized internally, so only their relative values matter. Hyperparameters for RAM were fine-tuned using Ray Tune Liaw et al. (2018); initialization for pretraining was derived from intuition. We use a standard normal base distribution which, after wrapping the fractional coordinate onto the flat torus, is effectively uniform. At inference time we use a uniform time grid.

E Reinforce Adjoint Matching

Reinforce adjoint matching frames fine-tuning an ODE velocity field in terms of stochastic optimal control. The central objective is to draw samples from a tilted distribution

$$\rho^\star(x) \propto \rho(x) e^{r(x)} \tag{47}$$

where r is a scalar-valued reward function Potaptchik et al. (2026). In the stochastic dynamical transport setting, samples from $\rho^\star$ can be drawn by sampling random Gaussian noise x_0 and integrating an SDE similar to equation 17 but written forwards in time

$$dx_t = \left[f_t(x_t) + \sigma_t^2 \nabla_{x_t} \log \rho_t(x_t) + \sigma_t^2 u_t^\star(x_t) \right] \mathrm{d}t + \sigma_t \mathrm{d}\overrightarrow{W}_t. \tag{48}$$

where $u_t^\star$ is a control field which, to sample from equation 47, should be chosen to maximize a terminal reward regularized by a path cost term

$$u^\star = \arg\max_u \mathbb{E}\left[r(x_1) - \frac{1}{2}\int_0^1 \sigma_\tau^2 \|u_\tau(x_\tau)\|^2 \,\mathrm{d}\tau\right]. \tag{49}$$

We write the optimal controller as $u_t^\star(x) = \nabla_x V_t(x)$ where V_t is the value function Domingo-Enrich et al. (2025)

$$\nabla_x V_t(x) = \mathbb{E}\left[r(x_1)\nabla_{x_t}\log\rho_{1|t}(x_1^{u^\star}|x_t)\middle|x_t = x\right] - \frac{1}{2}\mathbb{E}\left[\nabla\int_t^1 \sigma_\tau^2\|u_\tau^\star(x_\tau)\|^2\,\mathrm{d}\tau\middle|x_t = x\right]. \tag{50}$$

Dropping the path cost term, we arrive at an approximate value function gradient used in RAM:

$$\nabla_x V_t(x) \approx \mathbb{E}\left[r(x_1)\nabla_{x_t}\log\rho_{1|t}(x_1^{u^\star}|x_t)\middle|x_t = x\right]. \tag{51}$$

The superscript $u^\star$ indicates that the sample was generated using the controller.

RAM Fixed Point Objective In order to estimate the control in equation 51 we recall the uncontrolled SDE drift

$$b_t(x) = v_t(x) + \frac{\sigma_t^2}{2}\nabla_x \log\rho_t(x), \tag{52}$$

and controlled SDE drifts

$$b_t(x) + \sigma_t^2 u_t^\star(x) = v_t^\star(x) + \frac{\sigma_t^2}{2}\nabla_x\log\rho_t^\star(x), \tag{53}$$

Subtracting gives

$$\sigma_t^2 u_t^\star(x) = (v_t^\star(x) - v_t(x)) + \frac{\sigma_t^2}{2}\left(\nabla_x\log\rho_t^\star(x) - \nabla_x\log\rho_t(x)\right). \tag{54}$$

Provided that ρ_0 is Gaussian and that we use a memoryless schedule, we can write $\nabla_x\log\rho_t(x) = 2\sigma_t^{-2}(v_t(x) - \dot\beta_t\beta_t^{-1}x)$. Using this score identity, one can show that $v_t^\star$ satisfies a fixed point

$$\sigma_t^2 u_t^\star(x) = 2(v_t^\star(x) - v_t(x)) \tag{55}$$

$$v_t^\star(x) \approx v_t(x) + \frac{\sigma_t^2}{2}\mathbb{E}\left[r(x_1)\nabla_{x_t}\log\rho_{1|t}(x_1^{u^\star}|x_t)\middle|x_t = x\right] \tag{56}$$

The last step is to invoke the Bayes bridge score identity shown in the RAM derivation for the linear interpolant schedule ($\alpha_t = 1 - t, \beta_t = t$) Bergmeister et al. (2026),

$$\nabla_{x_t}\log\rho_{1|t}(x_1|x_t) = \frac{t}{1-t}((x_1 - x_0) - v_t^\star(x_t)), \tag{57}$$

and to enforce $\sigma_t^2/2 = (1-t)/t$ giving the RAM fixed point condition[2].

$$v_t^\star(x) - v_t(x) = \mathbb{E}\left[r(x_1)(\dot x_t - v_t^\star(x_t))|x_t = x\right]. \tag{58}$$

Application to OMatG-flash We highlight that the application of RAM to our OMatG-flash flow map improves performance notably (Section 4). We do note, however, that in the limit of long training times, we encounter instability and reward collapse and we illustrate this effect in Figure 4. Applying relaxation steps to the generated structure before reward computation improved stability of the algorithm. The numbers for OMatG-flash-RAM reported in Tables 1 and 2 are obtained using the checkpoint which exhibited the lowest average relative energy during training with relaxation applied before reward computation.

We do note that on $\mathbb{T}^{N\times 3}_{[0,1)}$ the application of the velocity-score reparameterization given in equation 24 is heuristic and we do not claim an extension of RAM to arbitrary manifolds is theoretically rigorous. Rewards are normalized to group-relative advantages following (Bergmeister et al., 2026).

[2]We emphasize that, in the original RAM literature, the problem is framed in reverse time where ρ_0 is clean data and ρ_1 is noise. Thus, many of the relations in their text have been adjusted to reflect our time convention.

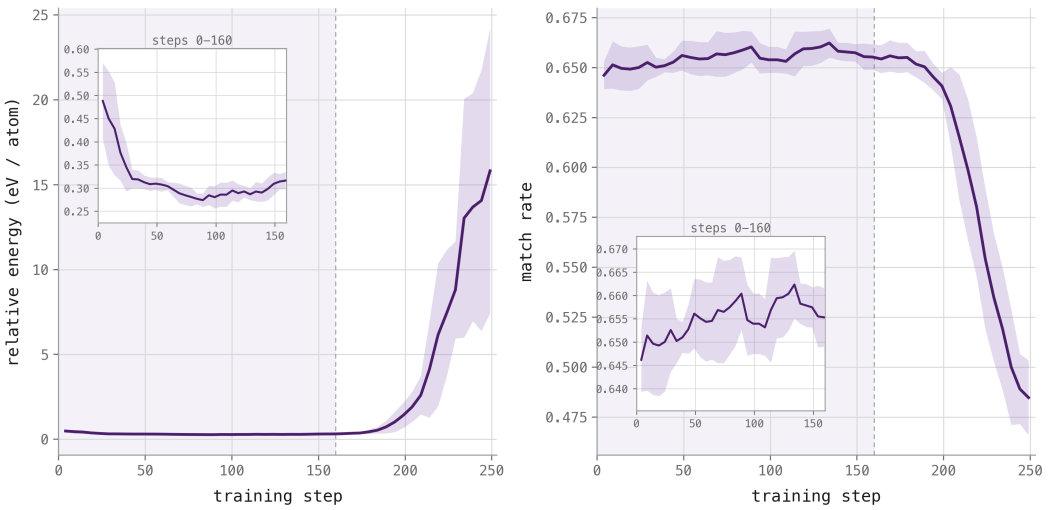


Figure 4: **Instability in RAM Minimization.** Early during post-training optimization with no relaxation applied the energy of generated samples decreases by over 0.2 eV/atom on average as shown in the inset. The one-to-one match rate trends slightly upward. Later, the RAM and consistency losses become unstable and model performance collapses.

F Benchmark Metrics

F.1 Crystal Structure Prediction

Match Rate and RMSE Match rate and RMSE are two canonical methods for measuring the performance of generative models on the CSP task. Both rely on the pymatgen `StructureMatcher` algorithm which determines a match within some specified tolerance by performing translations, rotations, and unimodular lattice operations. If a match is found, an RMSE can be computed which is then normalized by $\sqrt[3]{V/N_{\text{atoms}}}$ with V the volume of the cell and N_{atoms} the number of atoms. For CSP metrics we use the settings most often used in the generative modeling literature: `stol`=0.5, `ltol`=0.3, `angle_tol`=10.0. This metric is generally computed one-to-one where each generated structure is compared to its index-matched test set reference.

METRe and cRMSE It has been observed that one-to-one match rate is a suboptimal metric for determining CSP performance Martirossyan et al. (2025). This is due to the phenomenon of polymorphism in crystal structures, where a single composition can yield many diverse crystal structures. METRe attempts to combat this by instead comparing each reference structure against *every* generated structure. cRMSE is then a corrected version of RMSE where non-matched structures appear as an explicit penalty in the cRMSE computation

$$\text{cRMSE}\left(\{\mathbf{c}_{\text{gen},j}\}_{j=1}^{N_{\text{gen}}};\{\mathbf{c}_{\text{ref},i}\}_{i=1}^{N_{\text{ref}}}\right)=\frac{1}{N_{\text{ref}}}\sum_{i\in\mathcal{X}}\min_{j\in\mathcal{J}_i}\text{RMSE}(\mathbf{c}_{\text{gen},j},\mathbf{c}_{\text{ref},i})+\texttt{stol}\cdot\frac{\left(N_{\text{ref}}-N_{\text{ref}}^{\text{match}}\right)}{N_{\text{ref}}} \tag{59}$$

where $\mathbf{c}_{\text{gen}}$ is a generated crystal, $\mathbf{c}_{\text{ref}}$ is a reference crystal, N_{ref} is the number of reference test set structures, and $N_{\text{ref}}^{\text{match}}$ is the number of matched structures. We consider $\mathcal{X}$ as

$$\mathcal{X}=\{i\in\{1,...,N_{\text{ref}}\}\ :\ \mathcal{J}_i\neq\emptyset\} \tag{60}$$

where we have $\mathcal{J}_i$ as

$$\mathcal{J}_i=\{j\in\{1,...,N_{\text{gen}}\}\ :\ \mathbf{c}_{\text{gen},j}\ \text{matches}\ \mathbf{c}_{\text{ref},i}\}. \tag{61}$$

We illustrate METRe vs. one-to-one matching in Figure 5.

Validity For Table 1 we report results for all structures (`valid or invalid`) along with values restricted to only those passing certain validity checks (`valid`). Validity is assessed first via

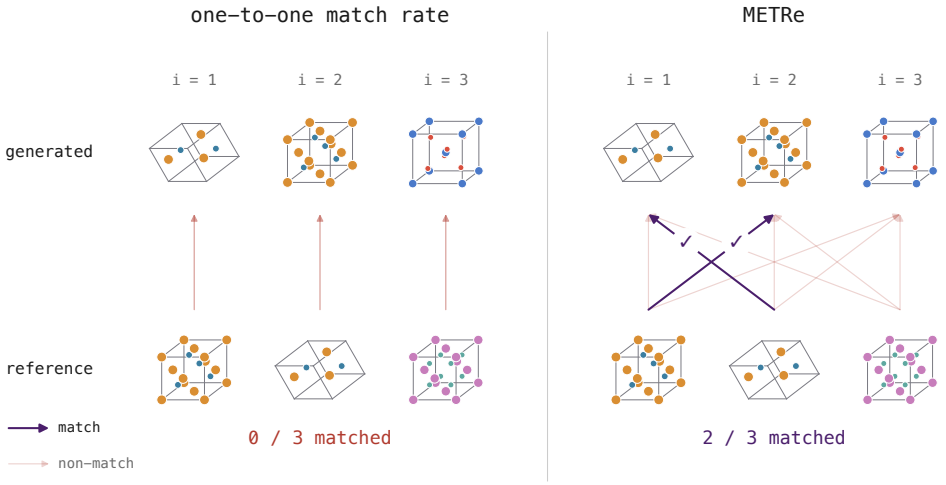


Figure 5: **Concept of METRe vs. One-To-One Matching.** The one-to-one method (**Left**) counts three non-matches despite the presence of two pairs of structures that *could* indeed be matched. METRe (**Right**) correctly scans the generated set for each reference structure, determining that two matches are present in the generated set.

structural assessment followed by SMACT chemical validity. Structural checks ensure that no interparticle distances are less than 0.5Å, the cell volume is greater than 0.1Å^3, and that a polar sine cutoff

$$V/(a \times b \times c) \geq 10^{-3} \tag{62}$$

is not violated where V is the volume of the cell and a, b, c are lattice vector lengths. If these are satisfied, a crystal structure must then pass a SMACT compositional validity check (Davies et al., 2019) followed by a chemical fingerprint validity check.

F.2 De Novo Generation

All DNG benchmark metrics are computed with code provided by LeMat-GenBench in their open source implementation using the `scripts/run_benchmarks.py` script with `comprehensive_multi_mlip_hull` configuration flag and `structure-matcher` fingerprinting Betala et al. (2026). The subsequent results are extracted with `scripts/extract_benchmark_metrics.py`. We prerelax our 2500 generated structures using MACE for up to 1000 steps with a force tolerance of 0.02 eV/Å before benchmarking with LeMat-GenBench. To expand on our results in Table 3 we include raw numbers as reported by LeMat for all DNG metrics in Table 5

Stability Stability is computed using a series of foundation model interatomic potentials. In order to determine if a material is stable or not it is input to UMA Wood et al. (2025), ORB Neumann et al. (2024), and MACE Batatia et al. (2023). The average energy-above-hull returned by these models is compared with the LeMat convex hull of stable phases corresponding to that reduced composition Bartel et al. (2020). A material is stable if and only if its energy-above-hull, $E_{\text{hull}} \leq 0.0$ eV/atom. Metastability for the purposes of M.S.U.N. computation is similarly defined, triggering if $0.0 < E_{\text{hull}} \leq 0.1$ eV/atom. Additionally, a material can only be considered if at least two of the three MLIPs in the ensemble return an energy for that entry.

Uniqueness and Novelty Uniqueness and novelty are computed in the same way but using different reference data. For both metrics pymatgen's `StructureMatcher` algorithm is used. The tolerances used by LeMat-GenBench are stricter than we use for computing CSP and are set `stol`=0.3, `ltol`=0.1, `angle_tol`=5.0. Uniqueness is determined by comparison to all structures in the generated set. Novelty, on the other hand, is determined against a held-out LeMat-Bulk reference set Siron et al. (2025).

Table 5: **DNG Raw Numbers.** LeMat-GenBench reports public leaderboard results using one decimal place. For transparency we report the raw numbers computed by the LeMat-GenBench evaluator and extractor scripts. Numbers are computed for an initial pool of 2500 structures. We take numbers for competing models from the public LeMat-GenBench leaderboard.

Method	Valid ↑	Novel ↑	Stable ↑	S.U.N. ↑	S.U.N. + M.S.U.N. ↑
DiffCSP	2392	1654	58	3	215
MatterGen	2392	1762	49	6	380
OMatG	2361	1255	284	24	464
MCFlow	2429	1306	298	18	490
Crystalite	2430	1331	317	38	604
OMatG-flash$_4$	2409	1672	180	34	455
OMatG-flash$_8$	2420	1584	196	26	537
OMatG-flash$_{16}$	2420	1524	221	29	564

F.3 Wall-Clock Inference Speed

We measure sampling speed using the conventions established in the literature. To measure results in accordance with the literature, we use one NVIDIA H100 GPU choosing a batch size of 8192 structures for the DNG task with the hyperparameters as reported in Appendix C. We use `bfloat16` inference which is consistent with how OMatG-flash was trained. We define generation time as the time required to sample $\mathbf{c}_0$, perform inference with OMatG-flash, and decode the data into a standard unit cell representation, excluding I/O. For reporting the Pareto frontier in Figure 1 we use baseline timing measurements as computed by Crystalite Veljković et al. (2026). For reporting Crystalite itself we use their "standard" inference setting as they indicate that this is the primary timing one should consider. We emphasize that, even with their reported optimized inference time of 5.14 seconds per thousand structures, our model at NFE=16 is an order of magnitude faster, recording 0.535 seconds per thousand structures.