OPEN MATERIALS GENERATION WITH INFERENCE-TIME REINFORCEMENT LEARNING

Philipp Höllmer & Stefano Martiniani
New York University

ABSTRACT

Continuous-time generative models for crystalline materials enable inverse materials design by learning to predict stable crystal structures, but incorporating explicit target properties into the generative process remains challenging. Policy-gradient reinforcement learning (RL) provides a principled mechanism for aligning generative models with downstream objectives but typically requires access to the score, which has prevented its application to flow-based models that learn only velocity fields. We introduce Open Materials Generation with Inference-time Reinforcement Learning (OMatG-IRL), a policy-gradient RL framework that operates directly on the learned velocity fields and eliminates the need for the explicit computation of the score. OMatG-IRL leverages stochastic perturbations of the underlying generation dynamics preserving the baseline performance of the pretrained generative model while enabling exploration and policy-gradient estimation at inference time. Using OMatG-IRL, we present the first application of RL to crystal structure prediction (CSP). Our method enables effective reinforcement of an energy-based objective while preserving diversity through composition conditioning, and it achieves performance competitive with score-based RL approaches. Finally, we show that OMatG-IRL can learn time-dependent velocity-annealing schedules, enabling accurate CSP with order-of-magnitude improvements in sampling efficiency and, correspondingly, reduction in generation time.

1 INTRODUCTION

The discovery and development of novel crystalline materials with targeted properties are fundamental to technological progress and machine-learning methods are rapidly emerging as a powerful tool of modern materials discovery (Sanchez-Lengeling & Aspuru-Guzik, 2018; Schmidt et al., 2019; Noh et al., 2020; Liu et al., 2023; Park et al., 2024; Long et al., 2024; Handoko & Made, 2025; De Breuck et al., 2025). Central to this development are generative models capable of realizing an inverse-design approach by suggesting stable and novel structures with predefined target properties.

Generative models for crystalline materials learn the chemical rules underlying crystal-structure datasets by approximating the high-dimensional distribution from which the data is drawn. Since the datasets typically comprise experimentally realized or computationally relaxed structures (Xie et al., 2022; Zeni et al., 2025), chemical (meta-)stability is implicitly learned as the primary target property of the generated samples. Accordingly, generative models are commonly benchmarked with respect to (meta-)stability. In the *de novo generation* (DNG) task, where crystal structure and composition are generated jointly, stability is typically evaluated directly (Betala et al., 2025; Szymanski & Bartel, 2025). In the *crystal structure prediction* (CSP) task, where a structure is generated for a given composition, stability is instead assessed by proxy: Computationally efficient structure-matching metrics such as match rate (Xie et al., 2022) and METRe/cRMSE (Martirossyan et al., 2025) compare generated structures to known stable structures from the crystal-structure datasets.

Beyond the implicit stability requirement, generative models in an inverse materials-design pipeline should also be able to align generation with explicit target properties such as desired mechanical, electronic, and magnetic properties. *Reinforcement learning* (RL) provides a flexible framework for directly optimizing generative models with respect to downstream objectives using black-box reward functions (Fan & Lee, 2023; Black et al., 2024). In an inverse materials-design setting, rewards can also render implicit constraints such as chemical (meta-)stability explicit for systematic

optimization. More broadly, RL overcomes a fundamental limitation of likelihood-based training of generative models, as it prioritizes task-specific objectives rather than the accurate approximation of the data log-likelihood. RL has proven highly effective, for example, for large language models, computer vision, and molecule design (Cao et al., 2025). In contrast, the application of RL to generative models for crystalline materials remains comparatively underexplored (De Breuck et al., 2025).

1.1 Related Works

A rapidly growing body of work has developed generative frameworks for inorganic crystalline materials using a wide range of representations, architectures, and generative paradigms [see, e.g., the recent review by De Breuck et al. (2025)]. An important class of continuous-time approaches considers joint generative processes over atomic (fractional) coordinates, lattice parameters, and atom types to produce periodic structures in real space without explicit space-group constraints [for symmetry-aware variants we refer to De Breuck et al. (2025)]. Many of these frameworks rely on score-based diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021) which learn the gradient of the log probability density and generate samples via reverse diffusion processes (Jiao et al., 2023; Yang et al., 2024; Zeni et al., 2025; Joshi et al., 2025; Cornet et al., 2025; Tangsongcharoen et al., 2025). In contrast, flow-matching models (Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023; Chen & Lipman, 2024; Lipman et al., 2024) learn a velocity field and generate samples by integrating an ordinary differential equation (Miller et al., 2024; Sriram et al., 2024; Luo et al., 2025). Finally, OMatG (Höllmer et al., 2025) builds on *stochastic interpolants* (SI) (Albergo et al., 2025), a unifying framework that encompasses both flow-matching and diffusion-based methods as special cases through specific choices of interpolants. Flow-matching and SI models can initiate generation from arbitrary base distributions, which has been shown to substantially improve performance (Miller et al., 2024; Sriram et al., 2024; Höllmer et al., 2025). Moreover, they typically require significantly fewer integration steps during inference, leading to markedly higher sampling efficiency (Miller et al., 2024; Luo et al., 2025; Höllmer et al., 2025).

Some of these frameworks are able to align generation with explicit target properties. This is typically achieved via direct input augmentation to the underlying network, either in a single conditional model (Luo et al., 2025), or by combining unconditional and conditional models through classifier-free guidance (Ho & Salimans, 2022; Yang et al., 2024; Zeni et al., 2025; Tangsongcharoen et al., 2025; Prakash et al., 2025). These approaches require sufficiently large and diverse labeled datasets for the desired target properties and remain constrained by the support of the underlying training distribution. Only recently, MatInvent (Chen et al., 2025) and Chemeleon2 (Park & Walsh, 2025) applied policy-gradient RL frameworks to the DNG task, with the latter adopting group-relative policy optimization (GRPO) (Shao et al., 2024) in conjunction with proximal policy optimization (PPO) (Schulman et al., 2017) to ensure stable optimization. Notably, both approaches had to incorporate explicit diversity rewards to prevent mode collapse during training.

MatInvent and Chemeleon2 apply RL to (latent) diffusion models, which provide explicit access to the score required by policy-gradient RL methods. Flow-matching frameworks and much of the SI framework in OMatG, however, only learn velocity fields without computing the score. Flow-GRPO addresses this limitation for the specific case of Gaussian base distributions and linear interpolant paths, where the score can be related to the learned velocity field (Liu et al., 2025). Still, a substantial portion of the SI design space in OMatG, whose systematic exploration previously led to state-of-the-art benchmark results, remains inaccessible to RL. This motivates the development of RL methods that operate across the full SI design space without requiring explicit score representations.

1.2 Our Contribution

We introduce *Open Materials Generation with Inference-time Reinforcement Learning* (OMatG-IRL), a policy-gradient RL framework for continuous-time flow-based generative models, and apply it to the CSP task. Our method operates directly on the learned velocity field and does not require an explicit score computation, while also naturally extending to score-based models when available. OMatG-IRL is inspired by the empirical observation that standard CSP evaluation metrics are robust to the introduction of stochastic perturbations into the underlying ordinary differential equation, enabling their use for exploration and policy-gradient estimation (see Fig. 1). Taking advantage of OMatG's flexibility to learn both velocity fields and scores, we directly compare score-based and

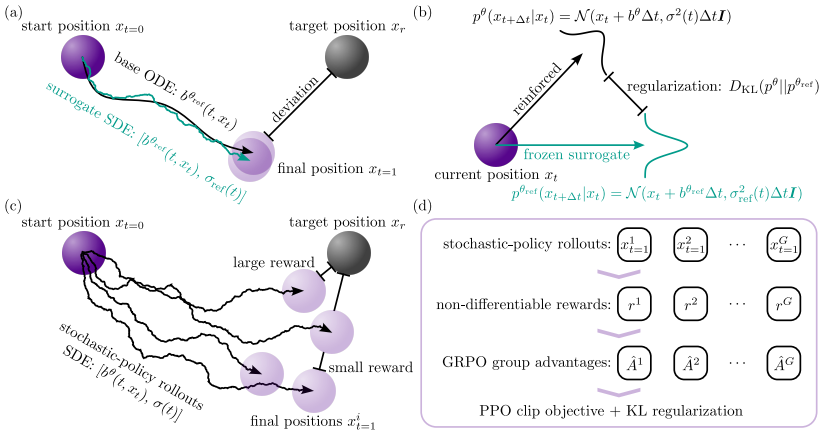

Figure 1: Inference-time RL for CSP in velocity-based OMatG-IRL. (a) The deterministic base ODE with pretrained velocity field $b^{\theta_{\text{ref}}}(t, x_t)$ is augmented with a small noise schedule $\sigma_{\text{ref}}(t)$, yielding a surrogate SDE that leaves evaluation metrics (e.g., deviation from a reference structure) of the final samples $x_{t=1}$ virtually unchanged. (b) The frozen surrogate defines a reference policy for KL regularization, while stochastic exploration is performed using a reinforced velocity field $b^{\theta}(t, x_t)$ and a (potentially different) noise schedule $\sigma(t)$. (c) GRPO compares terminal rewards $r^i = r(x^i_{t=1})$ obtained from multiple stochastic-policy rollouts under identical conditioning. (d) These rewards are transformed into GRPO group advantages and used, together with KL regularization, in a PPO-style clipped objective to update the policy.

velocity-based OMatG-IRL and demonstrate that both achieve comparable reinforcement performance.

This work also presents the first application of RL to the CSP task. We show that energy-based objectives can be effectively reinforced without the need for explicit diversity rewards; in contrast to DNG models, diversity in the CSP task naturally emerges from conditioning on composition. Finally, we demonstrate that our RL framework can be used to learn a time-dependent velocity-annealing schedule, replacing handcrafted annealing schemes and enabling accurate CSP with an order-of-magnitude reduction in the number of integration steps.

2 Background

2.1 Equivariant Crystal Representation

OMatG represents a crystalline material by its unit cell, which serves as the fundamental building block of the infinite periodic crystal. A unit cell containing N atoms is represented by three components $\{\boldsymbol{A}, \boldsymbol{X}, \boldsymbol{L}\}$. Here, $\boldsymbol{A} \in \mathbb{N}^N_{>0}$ denotes the atomic numbers of the atoms, $\boldsymbol{X} \in [0, 1)^{N\times 3}$ their fractional coordinates on a torus, and $\boldsymbol{L} \in \mathbb{R}^{3\times 3}$ the lattice matrix whose rows correspond to the three lattice vectors. The lattice vectors span a parallelepiped with volume $|\det \boldsymbol{L}| > 0$. Cartesian coordinates of the atoms within the parallelepiped are obtained as $\boldsymbol{X}\boldsymbol{L}$, and periodic replication of the unit cell generates the infinite crystal.

The crystal structure is invariant under joint permutations of atom indices in $\boldsymbol{A}$ and $\boldsymbol{X}$, global rotations of the lattice (which induce rigid rotations of the entire crystal), and translations of the fractional coordinates $\boldsymbol{X}$ (modulo 1). OMatG employs CSPNet (Jiao et al., 2023) as its model architecture, an $E(n)$-equivariant graph neural network (Satorras et al., 2021) whose internal representation is equivariant to permutations and rotations, and invariant to translations. As a result, the outputs of the neural network preserve the same symmetries.

2.2 Stochastic Interpolants

OMatG adopts the SI framework to model the continuous lattice matrices $\boldsymbol{L}$ and fractional coordinates $\boldsymbol{X}$. For CSP, the atomic numbers $\boldsymbol{A}$ are solely fixed conditional inputs during the generative

process. While the model output depends on the full triplet $\{\boldsymbol{A}, \boldsymbol{X}, \boldsymbol{L}\}$ during both training and generation, only the continuous variables $\boldsymbol{X}$ and $\boldsymbol{L}$ are interpolated and integrated in parallel by the SI dynamics. In the following, we use the symbol x to denote a generic continuous configuration variable, that is, either $\boldsymbol{X}$ or $\boldsymbol{L}$, with the understanding that model inputs refer to the full triplet.

A stochastic interpolant defines a stochastic process that bridges an arbitrary base distribution ρ_0 and the data distribution ρ_1 (Albergo et al., 2025):

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

Here, the time $t \in [0, 1]$ evolves a continuous random variable x_t from a sample $x_0 \sim \rho_0$ from the base distribution to a sample $x_1 \sim \rho_1$ from the training data. This construction requires, among other constraints (Albergo et al., 2025), $\alpha(0) = \beta(1) = 1$ and $\alpha(1) = \beta(0) = \gamma(0) = \gamma(1) = 0$. The random variable z is drawn from a standard Gaussian $\mathcal{N}(0, \boldsymbol{I})$ and injects stochasticity into the interpolant path.

The conditional velocity $\partial_t x_t$ from Eq. (1) is used to train an unconditional velocity field $b^\theta(t, x_t)$ that depends only on the current time t and configuration x_t by minimizing the loss function

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

The expectation is taken independently over $z \sim \mathcal{N}(0, \boldsymbol{I})$, $x_0 \sim \rho_0$, $x_1 \sim \rho_1$, and $t \sim \mathcal{U}(0, 1)$ where $\mathcal{U}(0, 1)$ is the uniform distribution between 0 and 1. Remarkably, the minimizer of Eq. (2) yields a velocity field whose marginal distributions ρ_t match those of the stochastic process x_t in Eq. (1). In particular, integrating the ordinary differential equation (ODE) $\mathrm{d}X_t = b^\theta(t, X_t)\, \mathrm{d}t$ from an initial sample $x_0 \sim \rho_0$ at $t = 0$ to $t = 1$ produces a sample x_1 from the data distribution ρ_1. This ODE is usually numerically integrated with the forward Euler method with N_t time steps of size Δt:

$$x_{t+\Delta t} = x_t + b^\theta(t, x_t)\, \Delta t. \tag{3}$$

In addition to the velocity field, one can optionally learn to predict the denoiser $z^\theta(t, x_t)$ at a given time t and configuration x_t by minimizing the loss function

$$\mathcal{L}_z(\theta) = \mathbb{E}_{t,z,x_0,x_1} \left\| z^\theta(t, x_t) - z \right\|^2. \tag{4}$$

The denoiser $z^\theta(t, x_t)$ is related to the score via $\nabla \log \rho^\theta(t, x_t) = -z^\theta(t, x_t)/\gamma(t)$. Equation (4) enables generative modeling by integrating the stochastic differential equation (SDE)

$$\mathrm{d}X_t = \left[b^\theta(t, X_t) - \frac{\sigma^2(t)}{2\gamma(t)} z^\theta(t, X_t) \right] \mathrm{d}t + \sigma(t)\, \mathrm{d}W_t, \tag{5}$$

where $\mathrm{d}W_t$ denotes Wiener process increments and $\sigma(t)$ controls the stochasticity. This SDE is usually numerically integrated from $t = 0$ to $t = 1$ with N_t Euler–Maruyama updates of size Δt:

$$x_{t+\Delta t} = x_t + \left[b^\theta(t, x_t) - \frac{\sigma^2(t)}{2\gamma(t)} z^\theta(t, x_t) \right] \Delta t + \sigma(t) \sqrt{\Delta t}\, \xi, \tag{6}$$

where $\xi \sim \mathcal{N}(0, \boldsymbol{I})$. This update induces an isotropic Gaussian conditional distribution $p^\theta(x_{t+\Delta t} | x_t)$ with the mean given by the drift term and covariance $\sigma^2(t) \Delta t\, \boldsymbol{I}$.

2.3 Policy-Gradient Reinforcement Learning

To apply policy-gradient RL to *stochastic* generative models, we formulate their iterative numerical integration as a Markov decision process $(\mathcal{S}, \mathcal{A}, \phi_0, P, R)$ with state space $\mathcal{S}$, action space $\mathcal{A}$, initial-state distribution ϕ_0, transition kernel P, and reward function R (Black et al., 2024). The state at time t is $s_t \coloneqq (t, x_t) \in \mathcal{S}$, and the initial-state distribution is $\phi_0 \coloneqq (\delta_0, \rho_0)$, where δ_y denotes a Dirac delta distribution centered at y. At each time step, an agent samples an action $a_t \in \mathcal{A}$ from its policy $\pi^\theta(a_t | s_t) \coloneqq p^\theta(x_{t+\Delta t} | x_t)$, corresponding to selecting the next configuration $a_t \coloneqq x_{t+\Delta t}$. The environment transition is deterministic: $P(s_{t+\Delta t} | s_t, a_t) \coloneqq (\delta_{t+\Delta t}, \delta_{x_{t+\Delta t}})$. Rewards are assigned only at the terminal state, with $R(s_t, a_t) \coloneqq r(s_{t=1})$ if $t = 1$ and $R(s_t, a_t) \coloneqq 0$ otherwise, where $r(s) = r(x)$ is an arbitrary configuration-dependent black-box reward function. The policy-gradient RL objective is to maximize the expected terminal reward over trajectories $\tau = (s_0, a_0, \ldots, s_1, a_1)$ sampled from the policy (denoted here as $\tau \sim \pi^\theta$):

$$\mathcal{J}_{\mathrm{RL}}(\theta) = \mathbb{E}_{\tau \sim \pi^\theta}\, r(s_{t=1}). \tag{7}$$

To update the current policy $\pi^{\theta_{\text{old}}} \rightarrow \pi^{\theta}$ to maximize $\mathcal{J}_{\text{RL}}(\pi^{\theta})$, we adopt GRPO (Shao et al., 2024), a policy-gradient method that directly compares rewards across multiple trajectories produced under identical conditioning. For the CSP task, this corresponds to rolling out G trajectories $\{\tau^i \sim \pi^{\theta_{\text{old}}}\}_{i=1}^{G}$ for the same composition, which may or may not start from different initial configurations $x_{t=0}^i$. The terminal rewards $r^i = r(x_{t=1}^i)$ are then compared within the group to compute group-relative advantages

$$\hat{A}^i = \frac{r^i - \text{mean}(\{r^i\}_{i=1}^{G})}{\text{std}(\{r^i\}_{i=1}^{G})}. \tag{8}$$

Since rewards are only defined at the terminal time, the same group-relative advantage $\hat{A}^i$ is used for all time steps along trajectory τ^i. The RL objective in Eq. (7) is then optimized by maximizing a surrogate objective inspired by PPO (Schulman et al., 2017), averaged over the group and all time steps:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \frac{\alpha}{GN_tS} \sum_{i,t} \min\Big[q_t^i(\theta)\hat{A}^i, \text{clip}\big(q_t^i(\theta), \varepsilon\big)\hat{A}^i\Big]. \tag{9}$$

Here, α is a weighting coefficient, $\text{clip}(x, \varepsilon) := \text{clip}(x, 1-\varepsilon, 1+\varepsilon)$ clamps x to the interval $[1-\varepsilon, 1+\varepsilon]$, and $q_t^i(\theta) := p^{\theta}(x_{t+\Delta t}^i | x_t^i) / p^{\theta_{\text{old}}}(x_{t+\Delta t}^i | x_t^i)$ is the probability ratio between the updated policy and the old policy that generated the trajectories. The normalization factor S accounts for variable numbers of atoms per structure and prevents policy updates from being biased toward larger crystals (see Appendix A). Given a fixed group of trajectories, the objective in Eq. (9) is typically optimized for multiple gradient steps (referred to as PPO epochs). To stabilize training, the clipping operation limits large policy updates in a trust-region-like manner controlled by the hyperparameter ε. The group-relative normalization of the advantages in Eq. (8) removes the need for a learned value function and stabilizes optimization across heterogeneous reward scales.

Typically, one includes an additional KL-regularization term in the maximized objective:

$$\mathcal{J}_{\text{KL}}(\theta) = -\frac{\beta}{GN_tS} \sum_{i,t} D_{\text{KL}}\big[p^{\theta}(x_{t+\Delta t}^i | x_t^i) \,||\, p^{\theta_{\text{ref}}}(x_{t+\Delta t}^i | x_t^i)\big]. \tag{10}$$

Here, $D_{\text{KL}}(P||Q)$ denotes the Kullback–Leibler divergence between distributions P and Q, and β is a weighting coefficient. The regularization keeps the reinforced policy close to the reference policy $p^{\theta_{\text{ref}}}$ of the pretrained generative model, which is critical for preserving its learned inductive biases.

2.4 Datasets and Metrics

In this work, we reinforce OMatG models trained on the *MP-20* dataset, which contains $45\,229$ structures from the Materials Project with at most 20 atoms per unit cell, spanning a wide range of structures and compositions across 89 atomic species (Jain et al., 2013; Xie et al., 2022). All structures in the dataset are relaxed using density-functional theory and together comprise most experimentally known inorganic materials with up to 20 atoms per unit cell.

As performance metrics, we consider the *match rate*, *RMSE*, *relative energy per atom*, and *invalid-energy rate*. For the match rate (Xie et al., 2022), we generate one structure for each composition in a reference dataset not seen during training and compare the generated and reference structures using Pymatgen's `StructureMatcher` with tolerances `stol=0.5`, `ltol=0.3`, and `angle_tol=10.0` (Ong et al., 2013). We then report the fraction of successful matches. The RMSE is computed as the average root-mean-square displacement between matched structures, normalized by $(V/N)^{1/3}$, where V is the matched volume and N is the number of atoms (Xie et al., 2022). The energy per atom of generated structures is evaluated using the MACE-MPA-0 model (Batatia et al., 2025), and we report energies relative to the reference structure of the same composition. Since the predicted energy can diverge for unphysical structures (e.g., due to atomic overlaps; see Appendix B), such structures are excluded from the energy statistics. We report the fraction of these samples as the invalid-energy rate.

We additionally consider an OMatG model trained on the *MP-20-polymorph-split* dataset (Martirossyan et al., 2025), a re-splitting of MP-20 in which polymorphs (i.e., distinct crystal structures sharing the same composition) are grouped into the same split. For this setting, we evaluate performance using *METRe* and *cRMSE* instead of match rate and RMSE. METRe measures how well generated structures cover the reference dataset by identifying the best match for each reference

structure, while cRMSE reports the average normalized root-mean-square displacement between best-matching pairs, with unmatched reference structures penalized using `stol`.

3 INFERENCE-TIME REINFORCEMENT LEARNING

OMatG can be trained to predict only the velocity field $b^\theta(t, x_t)$ via Eq. (2), or to jointly predict the velocity field and the denoiser $z^\theta(t, x_t)$ via Eq. (4). When the denoiser—and thus the score—is available, policy-gradient RL can be applied directly. In contrast, when only the velocity field is learned and no explicit score is available at inference time, a novel treatment is required.

3.1 SCORE-BASED OMATG-IRL

The Euler–Maruyama updates in Eq. (6) based on $b^\theta(t, x_t)$ and $z^\theta(t, x_t)$ define a stochastic policy $\pi^\theta(a_t|s_t) = p^\theta(x_{t+\Delta t}|x_t)$, which makes the policy-gradient RL framework of Section 2.3 directly applicable. Compared to existing RL applications in (latent) diffusion (Park & Walsh, 2025) or flow-matching frameworks (Liu et al., 2025), policy updates here jointly modify both the velocity field $b^\theta(t, x_t)$ and the denoiser $z^\theta(t, x_t)$, rather than acting on only a single component. In addition to the KL regularization in Eq. (10), we introduce an explicit denoiser-distillation regularization term:

$$\mathcal{J}_{\text{Dist}}(\theta) = -\frac{\delta}{GN_tS} \sum_{i,t} \left\| z^\theta(t, x_t) - z^{\theta_{\text{ref}}}(t, x_t) \right\|^2. \tag{11}$$

Increasing the weight δ constrains the reinforced denoiser to remain close to the pretrained model. While the KL objective constrains the full drift distribution [which depends jointly on $b^\theta(t, x_t)$ and $z^\theta(t, x_t)$], the denoiser-distillation term acts directly on $z^\theta(t, x_t)$ and thus explicitly controls the score.

3.2 VELOCITY-BASED OMATG-IRL

The Euler updates in Eq. (3) based on $b^\theta(t, x_t)$ are deterministic, which is problematic for policy-gradient RL: Likelihood-ratio computation typically requires expensive divergence estimation, and the absence of per-step stochasticity severely limits exploration (Liu et al., 2025). A natural remedy would be to convert the ODE into an SDE that preserves the same marginal distributions ρ_t while providing the stochasticity required for RL. Within the SI framework, this role is played by the SDE in Eq. (5) but in the setting of this section, the required denoiser $z^\theta(t, x_t)$ is not available.

Instead, we introduce a surrogate stochastic process by augmenting the dynamics with a noise schedule $\sigma_{\text{ref}}(t)$:

$$x_{t+\Delta t} = x_t + b^{\theta_{\text{ref}}}(t, x_t)\Delta t + \sigma_{\text{ref}}(t)\sqrt{\Delta t}\,\xi. \tag{12}$$

The omitted score correction is subdominant, $\mathcal{O}(\sigma_{\text{ref}}^2)$, so by Girsanov's theorem the deviation from the true marginal is bounded by $\mathcal{O}(\sigma_{\text{ref}}^2)$ (Øksendal, 2013). Empirically, evaluation metrics on the final samples confirm that the distribution remains virtually unchanged for sufficiently small $\sigma_{\text{ref}}(t)$ (see Section 4). The stochastic surrogate dynamics both enable exploration and allow policy-gradient updates to the velocity field $b^{\theta_{\text{ref}}}(t, x_t) \to b^\theta(t, x_t)$ (see Fig. 1). Importantly, the construction in Eq. (12) defines a well-posed reference policy for KL regularization. During RL, however, we are not restricted to the noise schedule $\sigma_{\text{ref}}(t)$ and may use alternative schedules $\sigma(t)$ to enhance exploration.

3.3 VELOCITY-ANNEALING OMATG-IRL

It has been empirically observed that velocity annealing improves the performance of flow-matching and SI models (Yim et al., 2023; Bose et al., 2024; Miller et al., 2024; Höllmer et al., 2025). In this approach, the velocity is modified during generation as $b^\theta(t, x_t) \to (1 + st)\, b^\theta(t, x_t)$, where s is a scalar hyperparameter. In OMatG-IRL, we introduce a learned, time-dependent velocity-annealing schedule $s^\theta(t)$. The reference policy is given by

$$x_{t+\Delta t} = x_t + b^{\theta_{\text{ref}}}\,\Delta t + \sigma_{\text{ref}}(t)\, b^{\theta_{\text{ref}}}\sqrt{\Delta t}\,\xi, \tag{13}$$

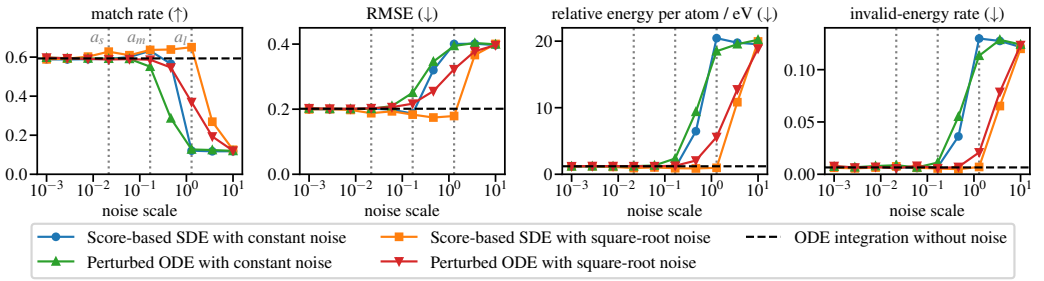

Figure 2: Test-set evaluation metrics for score-based SDE and perturbed velocity-based ODE integration of the atomic positions under different noise schedules. Small, medium, and large noise scales are denoted by a_s, a_m, and a_l, respectively.

where $b^{\theta_{\text{ref}}} = b^{\theta_{\text{ref}}}(t, x_t)$ is the frozen velocity field of the pretrained generative model and $\xi \sim \mathcal{N}(0, 1)$ is a standard Gaussian random variable. Policy-gradient RL then reinforces a zero-initialized schedule $s^\theta(t)$ (using a simple multilayer perceptron) in the velocity-annealed update $x_{t+\Delta t} = x_t + [1 + s^\theta(t)]b^{\theta_{\text{ref}}}\,\Delta t + \sigma(t)\,b^{\theta_{\text{ref}}}\sqrt{\Delta t}\,\xi$.

4 Experiments

We use OMatG-IRL to present the first application of policy-gradient RL to CSP. We show that the relative energy per atom can be effectively reinforced, achieving reductions of approximately $0.5\,\text{eV}$ per atom. The flexible design of OMatG further allows us to demonstrate that the velocity-only RL approach that we introduced in Section 3.2 achieves performance comparable to the score-based RL approach in Section 3.1. Finally, we show that velocity-annealing OMatG-IRL can replace handcrafted annealing schedules while enabling accurate CSP with at least one order of magnitude fewer integration steps.

4.1 Energy-Based Objective

The CSP task is arguably harder to reinforce than the DNG task. In DNG, a generative model can quickly focus on generating specific compositions that optimize the reward, effectively "hacking" the objective. The existing approaches in MatInvent (Chen et al., 2025) and Chemeleon2 (Park & Walsh, 2025) explicitly counteract this behavior by introducing diversity rewards. In the CSP setting considered here, the composition is fixed throughout generation, and diversity is naturally enforced through composition conditioning.

Common structure-based CSP metrics such as match rate, METRe, or cRMSE are ill-suited as reinforcement signals. These metrics measure similarity to specific ground-truth structures in the training set, an objective that is closely aligned with the original training signal. As a result, the additional learning signal by RL would be limited beyond what is already captured during pretraining. Furthermore, these metrics are inherently composition-dependent as each composition corresponds to a different set of target structures. This leads to sparse and poorly transferable rewards that do not generalize well across compositions. Optimizing such rewards would require repeated sweeps over large portions of the training dataset, making this approach computationally prohibitive.

In contrast, the relative energy per atom provides a physically meaningful and transferable reinforcement objective. The chemical rules that govern energy minimization (such as removing atomic overlaps) generalize across compositions. Reinforcing energy therefore promotes chemically reasonable structures independent of the specific target structure for a given composition. It also renders explicit a requirement that is already implicit in structure-based metrics, which assess stability only by comparing generated structures to known stable references. Energy-based reinforcement directly optimizes this underlying notion of chemical (meta-)stability, making it a natural and effective choice for RL in the CSP task.

4.2 INFERENCE-TIME ENERGY REINFORCEMENT

To compare the score-based OMatG-IRL approach of Section 3.1 to the velocity-based OMatG-IRL approach of Section 3.2, we use the publicly available pretrained Trig-SDE-Gamma model of OMatG, which was trained for the CSP task on the MP-20 dataset (Höllmer et al., 2025). For the atomic positions, this model predicts both the velocity field $b^\theta(t, x_t)$ and the denoiser $z^\theta(t, x_t)$ and can be integrated either in SDE mode via Eq. (6) or in perturbed ODE mode via Eq. (12). All modifications discussed below apply exclusively to the atomic positions, which constitute the harder-to-learn component of the CSP task. The lattice vectors are always integrated with an unperturbed ODE via Eq. (3) based on the frozen velocity field $b^{\theta_{\text{ref}}}(t, x_t)$.

In its original configuration, the Trig-SDE-Gamma OMatG model achieves a competitive match rate of approximately $69\,\%$ but relies on strong velocity annealing and a large number $N_t = 740$ of integration steps. For OMatG-IRL, we reduce the number of integration steps to $N_t = 50$ to limit computational cost and reduce variance in the terminal rewards $r^i = r(x^i_{t=1})$. We also remove velocity annealing in the pretrained model, as we find that keeping annealing while reducing the number of integration steps leads to severe performance degradation, indicating that annealing and time discretization are tightly coupled in the pretrained model (see Appendix C).

We consider two noise schedules for stochastic integration: a constant schedule $\sigma(t) = a$, and, inspired by Flow-GRPO, a square-root schedule $\sigma(t) = a\sqrt{(1-t)/t}$ which decays to zero as $t \to 1$. For both schedules, increasing the noise scale a beyond a moderate range deteriorates the performance of the pretrained model (see Fig. 2). Score-based SDE integration tolerates substantially larger noise levels before performance degrades due to the explicit drift correction provided by the learned score. In contrast, perturbed ODE integration is more sensitive to noise and require smaller perturbations to preserve baseline performance. Independently of the integration mode, the square-root noise schedule remains stable for larger noise scales a compared to the constant schedule.

Based on these observations, we use the square-root noise schedule for all policy-gradient RL experiments. For velocity-based OMatG-IRL, we compare small and medium reference noise scales ($a_{\text{ref}} = a_s$ and $a_{\text{ref}} = a_m$) to define the KL-regularized reference policy, and use the medium noise scale a_m during policy rollouts for exploration (see Fig. 2). For score-based OMatG-IRL, we use the medium noise scale a_m, which yields the best RL performance and thus ensures a fair comparison with the velocity-based approach (see Appendix D). As the reward, we use a clipped relative energy per atom to prevent outliers from dominating the group-normalized advantages and to keep policy updates stable (see Appendix E).

We run a hyperparameter sweep on the validation reward to determine the optimal configurations for both the score-based and velocity-based OMatG-IRL approaches (see Appendix F). We find that both approaches are able to effectively reinforce the relative energy per atom (see Fig. 3). At their respective validation optima, both methods achieve relative energies that are approximately $0.5\,\text{eV}$ per atom lower than the pretrained Trig-SDE-Gamma OMatG model evaluated with optimal velocity annealing and $N_t = 740$ integration steps. Despite operating with only $N_t = 50$ integration steps, energy reinforcement also reduces the RMSE and maintains the match rate. These results demonstrate that energy-based inference-time RL can significantly improve CSP performance while reducing the computational cost of generation by more than an order of magnitude.

Importantly, the velocity-based OMatG-IRL approach matches the performance of the score-based variant, showing that effective policy-gradient RL is possible without access to an explicit score at inference time and thus applicable to a broader class of flow-based generative models.

4.3 INFERENCE-TIME VELOCITY-ANNEALING REINFORCEMENT

We apply the velocity-annealing variant of OMatG-IRL introduced in Section 3.3 to the best-performing CSP model of OMatG: the Linear-ODE model trained on the MP-20-polymorph-split dataset (Martirossyan et al., 2025). This model achieves a METRe score of approximately $70\,\%$ and a cRMSE of around 0.19, but only when using a large number of integration steps ($N_t = 950$) together with carefully tuned velocity annealing on both atomic positions and lattice vectors.

Similar to Section 4.2, we remove the handcrafted velocity annealing from the pretrained model and reduce the number of integration steps to $N_t = 100$ for velocity-annealing OMatG-IRL. We

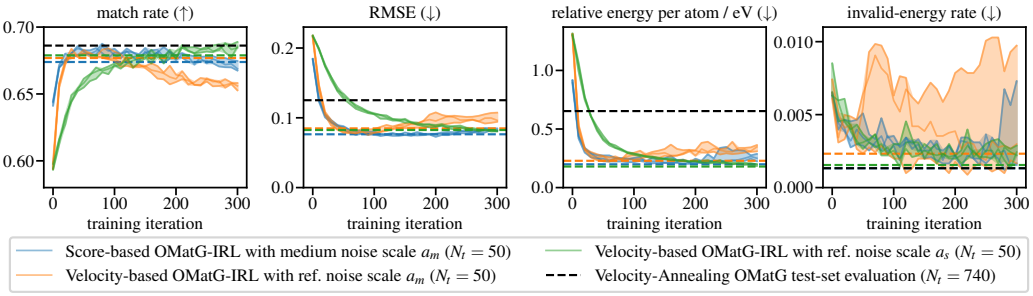

Figure 3: Evolution of validation metrics for score-based and velocity-based OMatG-IRL as a function of RL training iteration, shown for three random seeds of the same setup ($N_t = 50$). The colored dashed lines indicate the test-set performance of the OMatG-IRL checkpoint selected by the validation optimum. For reference, we also show the test-set performance of the original velocity-annealed OMatG model evaluated with $N_t = 740$ integration steps. The velocity-based OMatG-IRL setup with small reference noise scale a_s (green) uses fewer PPO epochs per update, resulting in slower but more stable reinforcement.

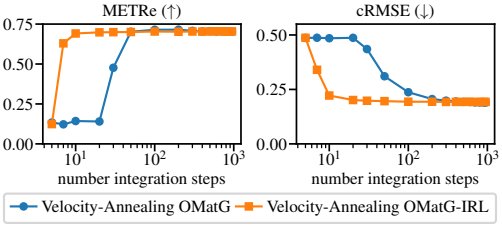

Figure 4: Test-set evaluation metrics for velocity-annealing OMatG and OMatG-IRL as a function of the number of integration steps N_t, highlighting the improved robustness of OMatG-IRL to aggressive time discretization.

first identify an appropriate (constant) reference noise schedule $\sigma_{\text{ref}}(t)$ in Eq. (13), and then apply policy-gradient RL to learn a time-dependent velocity-annealing schedule $s^\theta(t)$ for both atomic positions and lattice vectors. Because the learned annealing schedule depends only on time (and not on composition), and because velocity annealing is known to directly improve structural metrics, we can reinforce the model using a cRMSE-like objective in this setting (see Appendix G for details).

Despite using only 100 integration steps, velocity-annealing OMatG-IRL fully recovers the performance of the original OMatG model that requires 950 steps. More remarkably, the learned velocity-annealing schedule $s^\theta(t)$ yields robust performance across a wide range of integration step counts (see Fig. 4), allowing the number of steps to be reduced as far as 10 with only minor degradation. This behavior stands in sharp contrast to the original OMatG model, whose performance rapidly deteriorates unless the number of integration steps is at least an order of magnitude larger. We verified that a hyperparameter sweep over the handcrafted velocity-annealing schedule at the integration step count of 10 does not recover the performance achieved by the learned velocity-annealing schedule, indicating that the improvement indeed arises from the learned annealing policy (see Appendix H).

5 Conclusion

We introduced OMatG-IRL, a policy-gradient RL framework for continuous-time generative models of crystalline materials, and demonstrated its effectiveness on CSP. A key result is that effective policy-gradient RL is possible even in velocity-only flow-based models, without access to an explicit score. Extending OMatG-IRL to DNG within OMatG requires addressing the discrete flow-matching implementation for the atomic species (Campbell et al., 2024). Conversely, our results suggest that highly accurate, diverse, and reinforced CSP models, when combined with a separate composition predictor, could also serve as powerful building blocks for scalable DNG in inverse materials design pipelines.

ACKNOWLEDGEMENTS

The authors acknowledge funding from NSF Grant OAC-2311632. This work was also supported by a grant from the Simons Foundation [MPS-T-MPS-00839534, MET].

We gratefully acknowledge use of the research computing resources of the Empire AI Consortium, Inc, with support from Empire State Development of the State of New York, the Simons Foundation, and the Secunda Family Foundation (Bloom et al., 2025). Moreover, the authors gratefully acknowledge the additional computational resources and consultation support provided by the IT High Performance Computing at New York University.

REFERENCES

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A Next-generation Hyperparameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019. URL https://doi.org/10.1145/3292500.3330701.

Michael Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic Interpolants: A Unifying Framework for Flows and Diffusions. *Journal of Machine Learning Research*, 26:1–80, 2025. URL http://jmlr.org/papers/v26/23-1605.html.

Michael Samuel Albergo and Eric Vanden-Eijnden. Building Normalizing Flows with Stochastic Interpolants. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=li7qeBbCR1t.

Ilyes Batatia, Philipp Benner, Yuan Chiang, Alin M. Elena, Dávid P. Kovács, Janosh Riebesell, Xavier R. Advincula, Mark Asta, Matthew Avaylon, William J. Baldwin, Fabian Berger, Noam Bernstein, Arghya Bhowmik, Filippo Bigi, Samuel M. Blau, Vlad Cărare, Michele Ceriotti, Sanggyu Chong, James P. Darby, Sandip De, Flaviano Della Pia, Volker L. Deringer, Rokas Elijošius, Zakariya El-Machachi, Edvin Fako, Fabio Falcioni, Andrea C. Ferrari, John L. A. Gardner, Mikołaj J. Gawkowski, Annalena Genreith-Schriever, Janine George, Rhys E. A. Goodall, Jonas Grandel, Clare P. Grey, Petr Grigorev, Shuang Han, Will Handley, Hendrik H. Heenen, Kersti Hermansson, Cheuk Hin Ho, Stephan Hofmann, Christian Holm, Jad Jaafar, Konstantin S. Jakob, Hyunwook Jung, Venkat Kapil, Aaron D. Kaplan, Nima Karimitari, James R. Kermode, Panagiotis Kourtis, Namu Kroupa, Jolla Kullgren, Matthew C. Kuner, Domantas Kuryla, Guoda Liepuoniute, Chen Lin, 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, Louise A. M. Rosset, Lars L. Schaaf, Christoph Schran, Benjamin X. Shi, Eric Sivonxay, Tamás K. Stenczel, Christopher Sutton, Viktor Svahn, Thomas D. Swinburne, Jules Tilly, Cas van der Oord, Santiago Vargas, Eszter Varga-Umbrich, Tejs Vegge, Martin Vondrák, Yangshuai Wang, William C. Witt, Thomas Wolf, Fabian Zills, and Gábor Csányi. A foundation model for atomistic materials chemistry. *The Journal of Chemical Physics*, 163:184110, 2025. URL https://doi.org/10.1063/5.0297006.

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, 2025. URL https://arxiv.org/abs/2512.04562.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training Diffusion Models with Reinforcement Learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=YCWjhGrJFD.

Stacie Bloom, Joshua C. Brumberg, Ian Fisk, Robert J. Harrison, Robert Hull, Melur Ramasubramanian, Krystyn Van Vliet, and Jeannette Wing. Empire AI: A new model for provisioning AI and HPC for academic research in the public good. In *Practice and Experience in Advanced Research Computing (PEARC '25)*, pp. 4, 2025. URL https://doi.org/10.1145/3708035.3736070.

Joey Bose, Tara Akhound-Sadegh, Guillaume Huguet, Kilian Fatras, Jarrid Rector-Brooks, Cheng-Hao Liu, Andrei Cristian Nica, Maksym Korablyov, Michael M. Bronstein, and Alexander Tong. SE(3)-Stochastic Flow Matching for Protein Backbone Generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=kJFIH23hXb.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative Flows on Discrete State-Spaces: Enabling Multimodal Flows with Applications to Protein Co-Design. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=kQwSbv0BR4.

Yuanjiang Cao, Quan Z. Sheng, Julian McAuley, and Lina Yao. Reinforcement Learning for Generative AI: A Survey, 2025. URL https://arxiv.org/abs/2308.14328.

Junwu Chen, Jeff Guo, Edvin Fako, and Philippe Schwaller. Accelerating inverse materials design using generative diffusion models with reinforcement learning, 2025. URL https://arxiv.org/abs/2511.03112.

Ricky T. Q. Chen and Yaron Lipman. Flow Matching on General Geometries. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=g7ohDlTITL.

François R J Cornet, Federico Bergamin, Arghya Bhowmik, Juan Maria Garcia-Lastra, Jes Frellsen, and Mikkel N. Schmidt. Kinetic Langevin Diffusion for Crystalline Materials Generation. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=7J1kwZY72h.

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*, 11:370, 2025. URL https://doi.org/10.1038/s41524-025-01881-2.

Ying Fan and Kangwook Lee. Optimizing DDPM sampling with shortcut fine-tuning. In *Proceedings of the 40th International Conference on Machine Learning*, 2023. URL https://dl.acm.org/doi/10.5555/3618408.3618793.

Albertus Denny Handoko and Riko I Made. Artificial Intelligence and Generative Models for Materials Discovery: A Review. *World Scientific Annual Review of Functional Materials*, 3:2540001, 2025. URL https://doi.org/10.1142/S2810922825400018.

Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance, 2022. URL https://arxiv.org/abs/2207.12598.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020. URL https://dl.acm.org/doi/abs/10.5555/3495724.3496298.

Philipp Höllmer, Thomas Egg, Maya Martirossyan, Eric Fuemmeler, Zeren Shui, Amit Gupta, Pawan Prakash, Adrian Roitberg, Mingjie Liu, George Karypis, Mark Transtrum, Richard Hennig, Ellad B. Tadmor, and Stefano Martiniani. Open Materials Generation with Stochastic Interpolants. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=gHGrzxFujU.

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

Rui Jiao, Wenbing Huang, Peijia Lin, Jiaqi Han, Pin Chen, Yutong Lu, and Yang Liu. Crystal Structure Prediction by Joint Equivariant Diffusion. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=DNdN26m2Jk.

Chaitanya K. Joshi, Xiang Fu, Yi-Lun Liao, Vahe Gharakhanyan, Benjamin Kurt Miller, Anuroop Sriram, and Zachary Ward Ulissi. All-atom Diffusion Transformers: Unified generative modelling of molecules and materials. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=89QPmZjIhv.

Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E. Gonzalez, and Ion Stoica. Tune: A Research Platform for Distributed Model Selection and Training, 2018. URL https://arxiv.org/abs/1807.05118.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow Matching for Generative Modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow Matching Guide and Code, 2024. URL https://arxiv.org/abs/2412.06264.

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. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=oCBKGw5HNf.

Yue Liu, Zhengwei Yang, Zhenyao Yu, Zitu Liu, Dahui Liu, Hailong Lin, Mingqing Li, Shuchang Ma, Maxim Avdeev, and Siqi Shi. Generative artificial intelligence and its applications in materials science: Current situation and future perspectives. *Journal of Materiomics*, 9:798–816, 2023. URL https://doi.org/10.1016/j.jmat.2023.05.001.

Teng Long, Yixuan Zhang, and Hongbin Zhang. Generative deep learning for the inverse design of materials, 2024. URL https://arxiv.org/abs/2409.19124.

Xiaoshan Luo, Zhenyu Wang, Qingchang Wang, Xuechen Shao, Jian Lv, Lei Wang, Yanchao Wang, and Yanming Ma. CrystalFlow: a flow-based generative model for crystalline materials. *Nature Communications*, 16:9267, 2025. URL https://doi.org/10.1038/s41467-025-64364-4.

Maya Martirossyan, Thomas Egg, Philipp Höllmer, George Karypis, Mark Transtrum, Adrian Roitberg, Mingjie Liu, Richard Hennig, Ellad B. Tadmor, and Stefano Martiniani. All that structure matches does not glitter. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. URL https://openreview.net/forum?id=ig9ujp50D4.

Benjamin Kurt Miller, Ricky T. Q. Chen, Anuroop Sriram, and Brandon M Wood. FlowMM: Generating Materials with Riemannian Flow Matching. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=W4pB7VbzZI.

Juhwan Noh, Geun Ho Gu, Sungwon Kim, and Yousung Jung. Machine-enabled inverse design of inorganic solid materials: promises and challenges. *Chemical Science*, 11:4871–4881, 2020. URL https://www.doi.org/10.1039/D0SC00594K.

Bernt Øksendal. *Stochastic Differential Equations*. Springer, sixth edition, 2013.

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, 2013. URL https://doi.org/10.1016/j.commatsci.2012.10.028.

Hyunsoo Park and Aron Walsh. Guiding Generative Models to Uncover Diverse and Novel Crystals via Reinforcement Learning, 2025. URL https://arxiv.org/abs/2511.07158.

Hyunsoo Park, Zhenzhu Li, and Aron Walsh. Has generative artificial intelligence solved inverse materials design? *Matter*, 7:2355–2367, 2024. URL https://doi.org/10.1016/j.matt.2024.05.017.

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, 2025. URL https://arxiv.org/abs/2509.25186.

Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361:360–365, 2018. doi: https://www.doi.org/10.1126/science.aat2663.

Víctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) Equivariant Graph Neural Networks. In *Proceedings of the 38th International Conference on Machine Learning*, 2021. URL https://proceedings.mlr.press/v139/satorras21a.html.

Jonathan Schmidt, Mário R. G. Marques, Silvana Botti, and Miguel A. L. Marques. Recent advances and applications of machine learning in solid-state materials science. *npj Computational Materials*, 5:83, 2019. URL https://doi.org/10.1038/s41524-019-0221-0.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

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, 2024. URL https://arxiv.org/abs/2402.03300.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Anuroop Sriram, Benjamin Kurt Miller, Ricky T. Q. Chen, and Brandon M Wood. FlowLLM: Flow Matching for Material Generation with Large Language Models as Base Distributions. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=0bFXbEMz8e.

Nathan J. Szymanski and Christopher J. Bartel. Establishing baselines for generative discovery of inorganic crystals. *Materials Horizons*, 12:8000–8011, 2025. URL https://www.doi.org/10.1039/D5MH00010F.

Krit Tangsongcharoen, Teerachote Pakornchote, Chayanon Atthapak, Natthaphon Choomphonanomakhun, Annop Ektarawong, Björn Alling, Christopher Sutton, Thiti Bovornratanaraks, and Thiparat Chotibut. CrystalGRW: Generative Modeling of Crystal Structures with Targeted Properties via Geodesic Random Walks, 2025. URL https://arxiv.org/abs/2501.08998.

Tian Xie, Xiang Fu, Octavian-Eugen Ganea, Regina Barzilay, and Tommi S. Jaakkola. Crystal Diffusion Variational Autoencoder for Periodic Material Generation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=03RLpj-tc_.

Sherry Yang, KwangHwan Cho, Amil Merchant, Pieter Abbeel, Dale Schuurmans, Igor Mordatch, and Ekin Dogus Cubuk. Scalable Diffusion for Materials Generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=wm4WlHoXpC.

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

Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Zilong Wang, Aliaksandra Shysheya, Jonathan Crabbé, Shoko Ueda, Roberto Sordillo, Lixin Sun, Jake Smith, Bichlien Nguyen, Hannes Schulz, Sarah Lewis, Chin-Wei Huang, Ziheng Lu, Yichi Zhou, Han Yang, Hongxia Hao, Jielan Li, Chunlei Yang, Wenjie Li, Ryota Tomioka, and Tian Xie. A generative model for inorganic materials design. *Nature*, 639:624–632, 2025. URL https://doi.org/10.1038/s41586-025-08628-5.

A Normalization for Variable-Size Structures

Crystalline-material datasets contain structures with varying numbers of atoms, which causes both the GRPO objective in Eq. (9) and the KL regularization in Eq. (10) to scale with structure size. Without correction, this would bias policy updates toward larger structures. To prevent such size-dependent effects, we either divide the advantages by the number of atoms in the structure, or we compute PPO likelihood ratios per atom and average them to obtain a structure-level policy loss. For the KL regularization [and similarly for the denoiser-distillation regularization in Eq. (11)], we similarly average the per-atom KL divergence to obtain a structure-level penalty. The choice of normalization strategy is treated as a hyperparameter and part of our hyperparameter sweep (see Appendix F).

B Invalid-Energy Rate

We exclude invalid generated structures from the energy statistics and instead report the fraction of such invalid samples. In OMatG-IRL, a generated structure is considered invalid if it satisfies at least one of the following criteria:

1. The unit-cell volume is smaller than $0.1\,\text{Å}^3$.
2. Any pairwise atomic distance (accounting for periodic boundary conditions) is smaller than $0.5\,\text{Å}$.
3. The polar sine of the lattice vectors is smaller than 10^{-3}, indicating a nearly degenerate unit cell.

C Sensitivity of OMatG to Integration Steps and Velocity Annealing

In Table 1, we show that the performance of the pretrained Trig-SDE-Gamma model in OMatG is largely unchanged when switching between SDE integration via Eq. (6) and ODE integration via Eq. (3). However, when the number of integration steps is reduced while retaining velocity annealing, performance deteriorates severely. This effect is illustrated for the ODE-based integration, where no additional noise schedule needs to be tuned. In contrast, when velocity annealing is removed, reducing the number of integration steps has only a minor impact on performance, indicating that velocity annealing and number of integration steps are tightly coupled in the pretrained model.

Table 1: Match rate and RMSE of the Trig-SDE-Gamma OMatG model for different integration modes, velocity-annealing settings, and integration step counts, illustrating the strong coupling between annealing and step number.

Integration type	Velocity annealing	Integration steps	Match rate (%) ↑	RMSE ↓
SDE	✓	740	68.62	0.1252
ODE	✓	740	67.30	0.1145
ODE	✓	100	55.07	0.3644
ODE	✓	50	13.39	0.3932
ODE	✗	740	60.59	0.1749
ODE	✗	100	60.05	0.1891
ODE	✗	50	59.40	0.2024

D Score-Based OMatG-IRL at Different Noise Scales

In Fig. 5, we show the evolution of validation metrics during RL training, analogous to Fig. 3, for the score-based variant of OMatG-IRL under different noise scales of the square-root schedule (see

Fig. 2). The medium noise scale a_m yields the most effective reinforcement behavior, with rapid convergence to a minimum of the relative energy per atom. The small noise scale a_s leads to slower improvement due to insufficient exploration, while the large noise scale a_l is counterproductive for energy minimization. This behavior is consistent with observations reported in Flow-GRPO (Liu et al., 2025). Based on these results, we use the score-based OMatG-IRL model with noise scale a_m for comparison with the velocity-only approach in the main experiments.

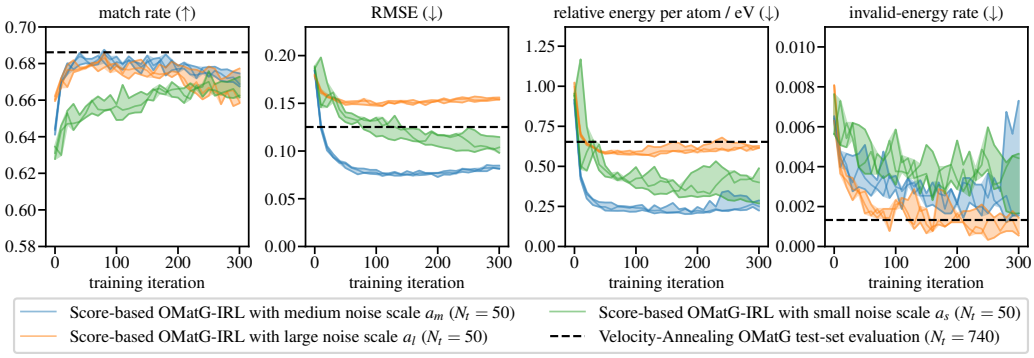

Figure 5: Evolution of validation metrics for score-based OMatG-IRL with different noise scales as a function of RL training iteration, shown for three random seeds of the same setup ($N_t = 50$). For reference, we also show the test-set performance of the original velocity-annealed OMatG model evaluated with $N_t = 950$ integration steps.

E Energy-Based Reward

As the energy-based rewards for a GRPO group of structures in OMatG-IRL, we first identify invalid structures (see Appendix B) and assign them a penalty energy of 3 eV per atom, noting that typical energies per atom in the MP-20 dataset are well below 0 eV. For all remaining valid structures, we estimate the energy per atom using MACE-MPA-0 (Batatia et al., 2025) and clip the resulting values to a band of three standard deviations around the group mean to prevent outliers from dominating the advantages. The reward is then defined as the negative of the (clipped) energy per atom, such that lower energies correspond to higher rewards. Since all structures within a GRPO group share the same composition, we do not need to compute explicit relative energies with respect to a reference structure. Any constant energy offset cancels out in the computation of group-relative advantages in Eq. (8).

F Hyperparameter Optimization

We optimize the hyperparameters of the policy-gradient RL setup in OMatG-IRL using Ray Tune (Liaw et al., 2018) with Optuna (Akiba et al., 2019) as the underlying sampler. For the RL experiments in Section 4.2, which adjust the velocity field $b^\theta(t, x_t)$ of the atomic positions, we tune the following hyperparameters (when applicable), sampled from the distributions below:

- GRPO group size $G \sim \mathrm{Choice}(32, 64)$ (with a fixed number of 16 groups).
- Sharing initial positions within a GRPO group $\sim \mathrm{Choice}(\mathrm{True}, \mathrm{False})$.
- Number of PPO epochs $\sim \mathrm{Choice}(1, 2, 3, 4)$.
- PPO clipping parameter $\varepsilon \sim \mathrm{Uniform}(0.05, 0.3)$.
- Policy loss weight $\sim \mathrm{LogUniform}(0.1, 10.0)$.
- KL regularization weight $\sim \mathrm{Choice}(0.0, 0.00001, 0.00003, 0.0001, 0.0003, 0.001, 0.003, 0.01)$.
- Denoiser-distillation weight $\sim \mathrm{Choice}(0.0, 0.000001, 0.000003, 0.00001, 0.00003, 0.0001, 0.0003, 0.001)$.

- Adam learning rate $\sim \mathrm{LogUniform}(0.00001, 0.001)$.
- Choice of normalization strategy for variable-size structures (see Appendix A).

For the experiments in Section 4.3, which learn time-dependent scaling functions $s^\theta(t, x_t)$ for both positions and lattice parameters, we use a more aggressive search space (in particular, without any regularization):

- GRPO group size $G \sim \mathrm{Choice}(32, 64)$ (with a fixed number of 16 groups).
- Sharing initial positions within a GRPO group $\sim \mathrm{Choice}(\mathrm{True}, \mathrm{False})$.
- Number of PPO epochs $\sim \mathrm{Choice}(1, 2, 3, 4, 5)$.
- PPO clipping parameter $\varepsilon \sim \mathrm{Uniform}(0.1, 0.5)$.
- Position-policy loss weight $\sim \mathrm{LogUniform}(0.01, 100.0)$.
- Adam learning rate $\sim \mathrm{LogUniform}(0.00001, 0.01)$.
- Hidden dimension of the multilayer perceptron of $s^\theta(t) \sim \mathrm{Choice}(32, 64, 128)$.
- Number of hidden layers of the multilayer perceptron of $s^\theta(t) \sim \mathrm{Choice}(1, 2, 3)$.
- Shared trunk for the atomic-position and lattice-vector scale prediction in $s^\theta(t) \sim \mathrm{Choice}(\mathrm{True}, \mathrm{False})$.

The policy weight of the lattice component is fixed to 1.0. Only in this case, the position and lattice weights are rescaled to sum to one.

All hyperparameter optimization runs were conducted under a fixed computational budget, and final configurations were selected based on the validation reward.

G Details on Velocity-Annealing Reinforcement

To learn a time-dependent velocity-annealing schedule $s^\theta(t)$ (parameterized by a simple multilayer perceptron) with velocity-annealing OMatG-IRL, we first perturb the ODE dynamics of the pretrained Linear-ODE OMatG model via Eq. (13) using both constant and square-root noise schedules at different noise scales, applied to both the atomic positions and lattice vectors. Increasing the noise scale beyond a moderate range deteriorates the performance of the pretrained model in all cases. Based on these observations, we use a constant noise schedule with a reference noise scale a_{ref} both to define the KL-regularized reference policy and to provide exploration noise during policy rollouts (see Fig. 6).

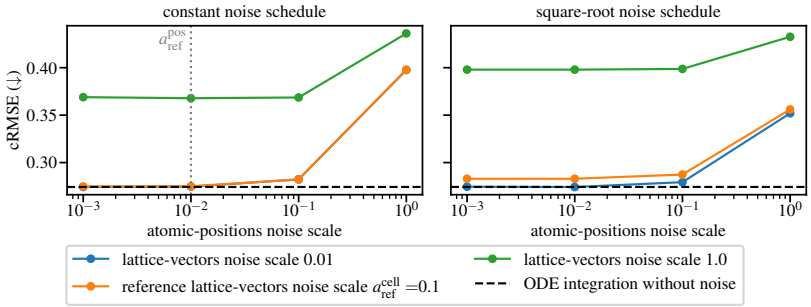

Figure 6: Test-set evaluation metrics for perturbed velocity-annealing ODE integration of the atomic positions and lattice vectors under different noise schedules and noise scales. The chosen reference noise scales for positions and lattice vectors are denoted by $a_{\mathrm{ref}}^{\mathrm{pos}}$ and $a_{\mathrm{ref}}^{\mathrm{pos}}$, respectively.

As the reward for policy-gradient RL, we use a cRMSE-like objective (see Section 2.4). For each generated structure of a given composition, we identify the best-matching polymorph of the same composition in the training set. The reward is defined as $r(x_{t=1}^i) = 0.5 - \mathrm{cRMSE}^i$, where cRMSE^i is the normalized root-mean-square displacement to this best match, with missing matches penalized in the mean using `stol=0.5`. This choice ensures that lower cRMSE values correspond to higher

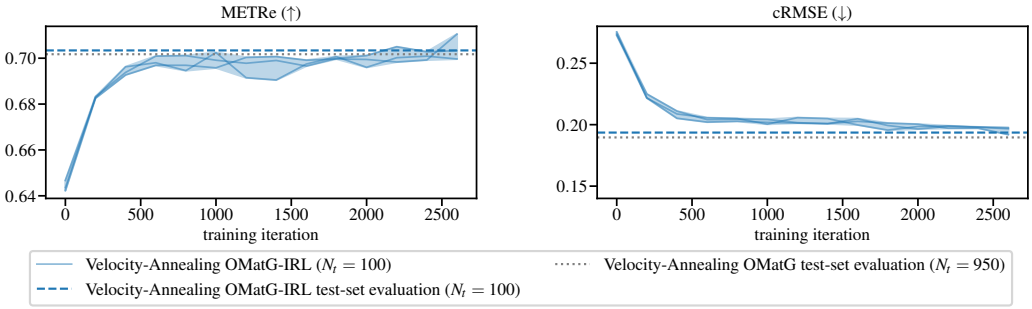

Figure 7: Evolution of validation metrics for velocity-annealing OMatG-IRL as a function of RL training iteration, shown for three random seeds of the same setup ($N_t = 100$). The dashed lines indicate the test-set performance of the OMatG-IRL checkpoint selected by the validation optimum. For reference, we also show the test-set performance of the original velocity-annealed OMatG model evaluated with $N_t = 950$ integration steps.

rewards. This formulation effectively inverts the standard cRMSE computation, which identifies best matches from the generated set for each reference structure.

A hyperparameter sweep on the validation reward is used to determine the optimal configuration for velocity-annealing OMatG-IRL (see Appendix F). Notably, because this setup only rescales velocities without changing their direction, we find that explicit KL regularization is unnecessary. Velocity-annealing OMatG-IRL is able to effectively reinforce the cRMSE metric and, at its validation optimum, recovers the performance of the pretrained Linear-ODE OMatG model while using an order of magnitude fewer integration steps ($N_t = 100$ vs. $N_t = 950$; see Fig. 7).

H Velocity-Annealing Hyperparameter Sweep

To verify that the performance achieved by velocity-annealing OMatG-IRL at $N_t = 10$ cannot be reproduced by tuning velocity annealing alone, we perform an explicit hyperparameter sweep over the velocity-annealing parameters of the pretrained OMatG model. The sweep is conducted using Ray Tune (Liaw et al., 2018) with Optuna (Akiba et al., 2019) as the underlying sampler, and explores a uniform range between 0 and 15 for the velocity-annealing coefficients of both the atomic positions and lattice vectors.

We find that velocity-annealed OMatG at $N_t = 10$ achieves a METRe score of $65.62\,\%$ and a cRMSE of 0.2671, which is substantially worse than velocity-annealing OMatG-IRL, which reaches a METRe score of $69.12\,\%$ and a cRMSE of 0.2220. Despite the explicit hyperparameter sweep over handcrafted velocity-annealing schedules, no configuration recovers the performance achieved by velocity-annealing OMatG-IRL at such low integration step counts, indicating that the improvement arises from the learned annealing policy.

In Fig. 8, we compare the learned and handcrafted velocity-annealing schedules. For the lattice vectors, the learned schedule $s^\theta(t)$ quickly drops to a constant value, whereas the handcrafted schedule is constrained to increase monotonically in time with a slope that appears largely independent of the number of integration steps N_t. For the atomic positions, in contrast, the slope of the handcrafted schedule depends strongly on N_t, while the learned schedule initially increases but then decreases after $t \approx 0.5$, eventually becoming negative. Importantly, $1 + s^\theta(t)$ remains positive at all times, so that the effective velocity $[1 + s^\theta(t)]\, b^\theta(t, x_t)$ is never reversed.

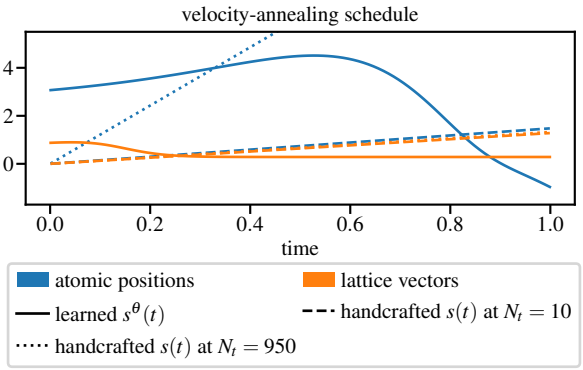

Figure 8: Velocity-annealing schedules for the atomic positions and lattice vectors, either learned with OMatG-IRL or obtained from a hyperparameter sweep over a handcrafted schedule at a given number of integration steps. The learned schedules adapt qualitatively differently from handcrafted ones.