Diffusion Tree Sampling for Inference Time Adaptation of Crystal Generative Models

Daniel T. Levy* **Vineet Jain*** **Tara Akhound-Sadegh**
Sékou-Oumar Kaba **Siamak Ravanbakhsh**

McGill University
Mila - Quebec Artificial Intelligence Institute

Abstract

Diffusion-based generative models have become strong priors over plausible crystal structures, but turning that prior into promising candidates requires steering toward several conflicting properties at once. Fine-tuning is expensive and produces a new model for every new objective, while differentiable guidance is not applicable to many real-world tasks. We instead adapt these generative models at *inference time* using Diffusion Tree Sampling (DTS), a Monte Carlo tree search procedure over the denoising trajectory. We make two technical contributions on top of DTS that are particularly relevant for AI-for-science applications. First, we show how to use existing property-conditional generative models (e.g., models fine-tuned for classifier-free guidance) as guided proposals inside DTS, without distorting the target distribution. Second, we extend DTS to multi-objective sampling, so that a single search produces samples spanning an entire Pareto front. We instantiate both adaptations on de novo crystal generation: improving the stability of crystals generated by different models, using pretrained conditional models to generate crystals belonging to rare space groups or with high bulk modulus, and generating crystals along a Pareto front of high dielectric constant and high bandgap materials. Across these tasks, DTS-based inference-time adaptation successfully improves model outputs without modifying the models themselves.

1 Introduction

Foundation models are flourishing in AI for science due to the superior generalization of large models trained on massive datasets [Kaplan et al., 2020, Bommasani et al., 2021]. In materials discovery, this has motivated powerful diffusion-based generative models that capture the distribution of plausible crystal structures: MatterGen [Zeni et al., 2025] and DiffCSP [Jiao et al., 2023] jointly model atom types, lattice, and coordinates, and can sample chemically plausible crystals. Yet, despite strong sample quality, these models have rarely translated into confirmed new discoveries.

A key barrier is that unconditional generation over all possible crystals is generally not directly useful. Candidate materials should not just be sampled from the manifold of all materials, but rather obtained through exploration of structures that are stable, sit in a particular property window, and trade off several conflicting properties (functionality, cost, element availability). Existing strategies for diffusion models typically modify the base model: fine-tuning property-conditional variants [Zeni et al., 2025], reward-conditioned RL [Chen et al., 2025], or active learning loops [Li et al., 2025]. Each requires labeled data, training compute, and produces a new model per design problem, which is at odds with the foundation-model paradigm.

*Equal contribution. Correspondence: daniel.levy6@mail.mcgill.ca. Code is available at: `https://github.com/danielTLevy/MaterialDTS`

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

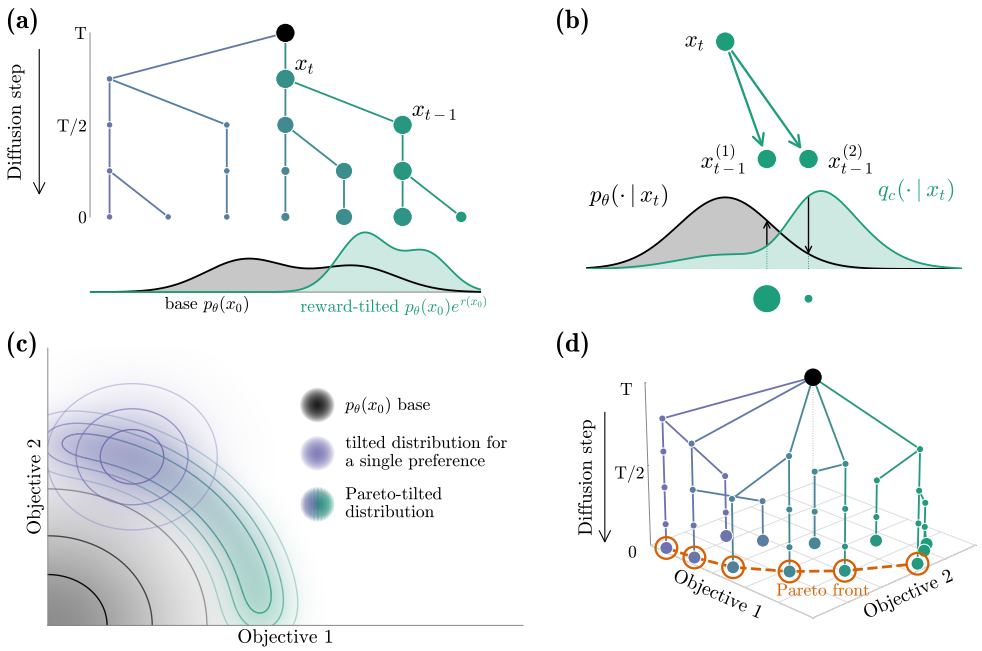


Figure 1: Overview of the method. **(a)**: DTS treats intermediate states of denoising diffusion as nodes in a tree, and steers sampling towards a reward-tilted distribution. Each node is attached to an estimate of expected reward (denoted here by node size). **(b)**: We may use a different model q_c that gives better denoising proposals, but we need to correct its influence by reweighting nodes by the proposal's ratio with the base model. **(c)**: In the multi-objective setting, we wish to sample from a distribution obtained by integrating reward-tilted distributions over different preferences between objectives. **(d)**: We use a single DTS tree to sample from this multi-objective target, conditioning each pass through the tree on a different preference.

Inference-time adaptation offers a different path: leave the pretrained model fixed, and steer its sampling toward the user's specific goals [Dhariwal and Nichol, 2021, Ho and Salimans, 2022, Yeh et al., 2024, Tang et al., 2024, Jain et al., 2025, Uehara et al., 2025]. This is well suited to materials, where the validation of candidate stability with density functional theory (DFT) calculations and the subsequent attempt to synthesize them is orders of magnitude more expensive than generation. Therefore, allocating inference compute to candidate quality is far more valuable than producing high volumes of low-quality structures. Among inference-time methods, Diffusion Tree Sampling (DTS) [Jain et al., 2025] casts denoising as a tree of partial trajectories, allocating compute adaptively to promising branches via Monte Carlo Tree Search (MCTS).

Two features of materials candidate generation are not addressed by the original DTS formulation, and motivate this work. First, in many cases, there *are* property-conditional models – e.g., MatterGen has variants for individual properties – which are dramatically more effective than the unconditional model at finding samples with a given property. Naively swapping in a conditional sampler inside DTS tilts towards the reward twice, resulting in a different distribution from the target. Second, real design problems are rarely single-objective: properties trade off, and the appropriate target is the *Pareto front* of the conflicting objectives [Miettinen, 1999, Deb et al., 2016, Riebesell et al., 2024].

Contributions. **(1)** *Conditional generative models inside DTS.* We show how to use property-conditional diffusion models (e.g. fine-tuned variants of MatterGen) as guided proposals within DTS, while keeping the reward-tilted unconditional base model as the target. This uses an importance-corrected backup and a corrected selection rule. **(2)** *Multi-objective DTS.* We formalize multi-objective sampling from a diffusion model as sampling from a mixture of reward-tilted distributions, each corresponding to a different preference over objectives. We extend DTS to this setting by maintaining per-objective soft value estimates and selecting expansions through sampled preferences, so that a single search samples the Pareto front while sharing computation across preferences and across different conditioning. **(3)** *Application to materials discovery.* We use the method to improve de novo

crystal generation across multiple tasks: we show improved results in generating stable crystals, we demonstrate the ability to generate crystals with specific features using pretrained conditional models, and we demonstrate how to sample crystals along a Pareto front of conflicting objectives, all without having to modify any model weights or use reward gradients.

2 Background

2.1 Diffusion models for materials discovery

Conditional materials generation aims to propose previously unknown crystalline compounds with desirable physical properties. The space of possible structures is combinatorially large, spanning composition, stoichiometry, lattice geometry, and atomic coordinates; only a small fraction is thermodynamically stable, and a smaller fraction still exhibits any given target property. We focus on *de novo generation*, which jointly proposes composition, lattice, and coordinates. Stability of candidates can be validated computationally by DFT, which is orders of magnitude more expensive than generation. Classical pipelines combine heuristic generation with first-principles relaxation [Oganov and Glass, 2006, Pickard and Needs, 2011]. Some methods search using basin hopping [Wales and Doye, 1997, Wales and Scheraga, 1999] – a structure with a natural analogue in the tree-search method we use (see Appendix B).

A recent line of work trains generative models directly on databases of known crystals; diffusion models are a strong fit since they jointly model the discrete (atom types) and continuous (lattice and coordinates) components of a crystal. We focus on two models. **DiffCSP** [Jiao et al., 2023] parameterizes a joint diffusion process over fractional coordinates, lattice, and atom types, and supports time-conditional classifier-based guidance. **MatterGen** [Zeni et al., 2025] is trained on a substantially larger dataset and provides adapter modules for efficient fine-tuning into property-conditional models, sampled with classifier-free guidance [Ho and Salimans, 2022]. Both suffer from a high rate of false positives: only a small fraction of generated structures are predicted to be stable, and a smaller fraction yet satisfies property constraints. Existing strategies modify the base model via classifier-based guidance [Jiao et al., 2023], fine-tuned conditional variants [Zeni et al., 2025], RL [Chen et al., 2025], and active learning [Li et al., 2025]. Our work is complementary: it steers a fixed pretrained generative model at inference time, and extends this steering to settings where conditional variants exist and where multiple objectives must be balanced.

2.2 Inference-time steering and Diffusion Tree Sampling

Let p_θ denote a pretrained diffusion model with reverse transitions $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$, inducing a prior $\mathbf{x}_0 \sim p_\theta(\mathbf{x}_0)$. Inference-time steering introduces a reward $r(\mathbf{x}_0)$ available only after full denoising, and aims to sample from the *reward-tilted* target $\pi^\star(\mathbf{x}_0) = \frac{1}{Z} p_\theta(\mathbf{x}_0) \exp(\lambda r(\mathbf{x}_0))$, which is the KL-regularized optimum of expected reward relative to the base model with inverse temperature λ. A useful characterization of expected reward at a noisy state $\mathbf{x}_t$ is the *soft value function* $v_t(\mathbf{x}_t) = \frac{1}{\lambda} \log \mathbb{E}[\exp(\lambda r(\mathbf{x}_0)) \mid \mathbf{x}_t]$ with $v_0(\mathbf{x}_0) = r(\mathbf{x}_0)$, which satisfies the soft Bellman recursion

$$v_t(\mathbf{x}_t) \;=\; \frac{1}{\lambda} \log \mathbb{E}_{p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)}[\exp(\lambda v_{t-1}(\mathbf{x}_{t-1}))] \,. \tag{1}$$

The optimal aligned reverse transition is $\pi_t^\star(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \propto p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \exp(\lambda v_{t-1}(\mathbf{x}_{t-1}))$. Most inference-time steering methods approximate v_{t-1} via gradient guidance, SMC, or local search over denoising proposals [Uehara et al., 2025].

A central difficulty is *value estimation at high noise*: common approximations become unreliable early in the reverse process. *Diffusion Tree Sampling* (DTS) [Jain et al., 2025] addresses this by casting denoising as a finite-horizon tree whose nodes are states $\mathbf{x}_t$ with value estimates $\hat{v}_t(\mathbf{x}_t)$ and whose edges are reverse transitions. The algorithm repeatedly builds a tree by starting at a root node $\mathbf{x}_{T+1}$ and performing (1) *selection* of children $\propto \exp(\lambda \hat{v})$ until reaching a node that should expand, (2) *expansion* by sampling a new child from p_θ, (3) *rollout* of the new child to a terminal $\mathbf{x}_0$, assigning $\hat{v}(\mathbf{x}_0) = r(\mathbf{x}_0)$ and (4) *backup* via the soft Bellman log-sum-exp. Reusing these backed-up values yields an anytime procedure whose empirical leaf distribution converges to $\pi^\star$; a greedy variant replaces sampling with argmax selection.

3 Adapting Diffusion Tree Sampling to Crystal Generative Models

Standard DTS assumes (i) a single scalar reward, and (ii) that the reverse transitions used during expansion and rollout are exactly the unconditional base model. Both assumptions are limiting in the materials setting. In this section we adapt DTS to relax them: Section 3.1 shows how to use existing property-conditional generative models as guided proposals inside DTS without changing the target distribution; Section 3.2 extends the search to multi-objective sampling. The full algorithm is given in Appendix A.

3.1 Adaptation 1: leveraging conditional models inside DTS

When available, a *property-guided* diffusion model – e.g., a conditional model trained for classifier-free guidance (CFG) – can be much more effective at *locally* steering denoising steps toward narrow high-reward regions than a purely reward-based search on top of the unconditional model. For example, MatterGen provides property-conditional variants for individual properties (bandgap, space group, etc.), and we would like to use them as proposals inside DTS without losing the well-defined target distribution induced by the unconditional base model.

We denote the unconditional reverse transition by $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ and a guided/conditional transition by $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, c)$, where c encodes a target (e.g. a desired property). In our setting, we assume that providing conditioning information c helps with optimizing some scalar reward $r(\cdot)$ (e.g. conditioning on a target property).

The double-tilt problem. A naive attempt would be to replace the base model inside DTS with the conditional model and then tilt by $\exp(\lambda r(\mathbf{x}_0))$. This would target $\pi_{\text{naive}}(\mathbf{x}_0) \propto p_\theta(\mathbf{x}_0 \mid c)\, \exp(\lambda r(\mathbf{x}_0))$, which would tilt twice: the conditional model already biases mass toward the properties encoded by c, and the additional Boltzmann tilt amplifies the same preference, yielding a different target than the original $\pi^\star$. An example of this issue is shown in Appendix C, using a Gaussian-mixture toy experiment.

Importance-corrected backup. Our solution is to keep the *unconditional* model as the target and use the conditional model only as a proposal, correcting via importance sampling (IS). Concretely, during expansion and rollout from $\mathbf{x}_t$ we sample $\mathbf{x}_{t-1}$ from the guided proposal $q_c(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \coloneqq p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, c)$ instead of $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$, and attach to the new node the *importance weight*

$$\eta(\mathbf{x}_{t-1}) \coloneqq \frac{p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)}{q_c(\mathbf{x}_{t-1} \mid \mathbf{x}_t)}, \tag{2}$$

the likelihood ratio of base to proposal. Models trained for CFG are convenient here because they provide both transitions. The soft value of Equation (1) is then an expectation under q_c weighted by η,

$$v_t(\mathbf{x}_t) = \frac{1}{\lambda} \log \mathbb{E}_{\mathbf{x}_{t-1} \sim q_c(\cdot \mid \mathbf{x}_t)}\big[\eta(\mathbf{x}_{t-1}) \exp\big(\lambda v_{t-1}(\mathbf{x}_{t-1})\big)\big], \tag{3}$$

so backups remain consistent with the unconditional base dynamics.

Importance correction in selection. Selection among existing children must be corrected for the same reason, since the children were drawn from q_c. We therefore select a child $\mathbf{z} \in \mathcal{C}(\mathbf{x}_t)$ with probability

$$\propto\ \eta(\mathbf{z})\ \exp\big(\lambda\, \hat{v}(\mathbf{z})\big), \tag{4}$$

which is self-normalized importance sampling of the aligned base transition $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \exp(\lambda v_{t-1}(\mathbf{x}_{t-1}))$. Without η, selection would be biased toward q_c and the tree would converge to a target that depends on the proposal.

3.2 Adaptation 2: Multi-Objective Diffusion Tree Sampling (MO-DTS)

Many materials problems involve *multiple* competing goals. Let $\mathbf{r}(\mathbf{x}) \coloneqq [r_1(\mathbf{x}), \ldots, r_m(\mathbf{x})]$ be a vector of m objectives to be maximized; a point $\mathbf{x}^\star$ *dominates* $\mathbf{x}$ if it is at least as good on all objectives and strictly better on one, and the set of non-dominated points is the *Pareto front*. Multi-objective optimization can be tackled by taking a *preference* over objectives $w \in \Delta^{m-1}$ that

encodes a trade-off, and a *scalarization*, which is a map $s : \mathbb{R}^m \times \Delta^{m-1} \to \mathbb{R}$ that turns a reward vector into a scalar according to w. The simplest choice is linear scalarization $s(\mathbf{a} \mid w) \coloneqq w^\top \mathbf{a}$, although other choices are possible (see Appendix E). For a fixed w this defines the scalar reward $r(\mathbf{x} \mid w) \coloneqq s(\mathbf{r}(\mathbf{x}) \mid w)$, and the multi-objective problem is reduced to a set of single-objective problems: $\max_{\mathbf{x}} r(\mathbf{x} \mid w)$.

Sampling from the Pareto front. Given objectives $\mathbf{r}(\mathbf{x}) \in \mathbb{R}^m$ and a scalarized $r(\mathbf{x} \mid w)$, we define the target of *sampling from the Pareto front* as first sampling a preference $w \sim p(w)$ where $p(w)$ is a user-chosen distribution over preferences (e.g., Dirichlet on the simplex), then sampling $\mathbf{x}_0$ from the corresponding reward-tilted base distribution:

$$\pi^\star(\mathbf{x}_0 \mid w) = p_\theta(\mathbf{x}_0) \exp(\lambda r(\mathbf{x}_0 \mid w))/Z(w), \qquad Z(w) = \mathbb{E}_{\mathbf{x}_0 \sim p_\theta}\left[\exp(\lambda r(\mathbf{x}_0 \mid w))\right].$$

Marginalizing out w yields a mixture of reward-tilted distributions,

$$\pi^\star(\mathbf{x}_0) = p_\theta(\mathbf{x}_0)\, \mathbb{E}_{w \sim p(w)}\left[\exp(\lambda r(\mathbf{x}_0 \mid w))/Z(w)\right], \tag{5}$$

which assigns high probability to candidates that are strong under *some* trade-off, while remaining plausible under the base model.

A single shared tree. When lifting a single-objective method to the multi-objective setting, treating each scalarized objective $r(\cdot \mid w)$ in isolation is highly inefficient: a key design requirement is to share computation across preferences. It is non-trivial to adapt arbitrary alignment methods (Best-of-N, SMC [Singhal et al., 2025]) under this requirement. DTS [Jain et al., 2025] can be extended rather directly to approximately sample from Equation (5) while sharing computation: maintain a separate value estimate per objective during backup, and use a *scalarized value* based on these estimates during selection.

Each terminal rollout produces a vector reward $\mathbf{r}(\mathbf{x}_0)$, and we maintain at each node a value vector $\mathbf{v}_t(\mathbf{x}_t) = [v_{t,1}(\mathbf{x}_t), \ldots, v_{t,m}(\mathbf{x}_t)]$, with each $v_{t,i}$ updated via the soft backup of Equation (1) applied to objective i. Each tree traversal first samples $w \sim p(w)$, then computes a scalarized value by applying the scalarization to the node's value vector, $v_t(\mathbf{x}_t|w) \coloneqq s(\mathbf{v}(\mathbf{x}_t) \mid w)$.

MO-DTS and conditioning. When available, we can use a conditional diffusion model where the condition is a function of the preference. That is, for each preference w, our diffusion model samples from $q_w(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \coloneqq p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, c(w))$ where $c(w)$ is a mapping from preferences to conditions that would help maximize $r(\cdot|w)$. In practice, this can mean conditioning the diffusion model on the features of samples in the training set that maximize $r(\cdot|w)$, as they represent reasonable targets.

At each pass through the tree, w can be sampled anew. The sampled w not only affects the choice of children while moving down the tree from the root, but also affects how new leaf nodes are generated. However, due to the importance correction of Section 3.1, the target distribution does not depend on the choice of $c(w)$, and values $\mathbf{v}_t(\mathbf{x}_t)$ do not depend on w, so the whole tree remains consistent.

Per-objective values as an upper-confidence approximation. Scalarizing at selection time is an approximation: the exact soft value of the scalarized reward, $v_t^\star(\mathbf{x}_t \mid w) \coloneqq \frac{1}{\lambda} \log \mathbb{E}[\exp(\lambda\, r(\mathbf{x}_0 \mid w)) \mid \mathbf{x}_t]$, would require a separate backup for every w, whereas MO-DTS backs up each objective once and uses $v_t(\mathbf{x}_t \mid w) = s(\mathbf{v}_t(\mathbf{x}_t) \mid w)$. In Appendix F we show that the surrogate is optimistic: it upper-bounds $v_t^\star$ for linear scalarizations at every λ, and coincides with $v_t^\star$ at $t = 0$. The gap is governed by how far the objectives are from co-varying under the denoising distribution, so selection is most optimistic about nodes whose objectives trade off; MO-DTS therefore provides a search heuristic for Equation (5) rather than an exact sampler for every w.

3.3 Coarse branching and parallelization

Following Jain et al. [2025], we restrict branching to a set of *branching times* $\mathcal{T} \subseteq \{1, \ldots, T+1\}$. A node $\mathbf{x}_t$ is *expandable* if $t \in \mathcal{T}$ and progressive widening permits, $|\mathcal{C}(\mathbf{x}_t)| < C\, N(\mathbf{x}_t)^\alpha$, where $\mathcal{C}$ is the children of a node, N is the number of visits so far, and C and α are hyperparameters. Nodes at other times are created only during rollouts and have a single child, so selection passes through them deterministically. In practice, we set the branching times to be regularly spaced every L timesteps. At a matched number of function evaluations (NFEs), DTS produces more samples due to trajectory

reuse. The search also admits a batched implementation: B traversals advance synchronously from the root node, and all rollout steps at the same t are batched into one call to the sampler, so each pass through the tree generates B leaf nodes. Throughput is thus at least as high as generating B independent trajectories as in best-of-N.

4 Application to Materials Discovery

We instantiate the framework on four crystal-generation problems chosen to demonstrate both technical contributions across two generative models. First, we demonstrate how inference-time steering with regular DTS improves models' ability to generate stable crystals. Next, symmetry-conditioned generation targeting rare space groups (Section 4.2) demonstrates the benefit of using conditional models and exercises the importance-corrected guided proposals of Section 3.1. A multi-objective experiment targeting the bandgap and dielectric constant tradeoff demonstrates the utility of building a single tree across different preferences in Section 4.3. We examine the effect of a mismatched proposal distribution and the use of importance-correction in Section 4.4, targeting high bulk modulus materials.

4.1 De novo stability search

A central challenge in de novo crystal generation is producing *thermodynamically stable* structures, quantified by the *energy above the convex hull* (E_{hull}): for a given composition, the convex hull is the lower envelope of formation energies across all known competing phases. E_{hull}=0 is stable; small positive E_{hull} (typically below 0.1 eV/atom) is *metastable*. Current generative models produce stable samples only a small fraction of the time, bottlenecking their usefulness [Betala et al., 2025]. E_{hull} depends on the composition through the (non-differentiable) hull construction, ruling out gradient-based guidance.

We define a single-objective reward based on E_{hull} as predicted by MatterSim [Yang et al., 2024], with the Alex-MP hull from Zeni et al. [2025]. This setup reduces to DTS with scalar reward and serves as a controlled assessment of tree-search-based steering vs. best-of-N sampling from the same generative model. We evaluate using the common summary metric of the rate of *Stable, Unique, Novel* (SUN) structures [Zeni et al., 2025], where uniqueness and novelty with respect to the training set are determined using Pymatgen's `StructureMatcher` [Ong et al., 2013].

Results with MatterGen. We compare three configurations at matched NFE budgets: standard sampling from MatterGen (best-of-N), DTS with null reward $r(\mathbf{x}_0) = 0$, and DTS with the reward $r(\mathbf{x}_0) = -E_{\text{hull}}(\mathbf{x}_0)$. The null-reward control isolates the effect of tree search itself (trajectory reuse) from the stability reward; we also vary the progressive widening parameter α, which controls tree width. Results averaged over three seeds are shown in Table 1. Tree search improves over best-of-N in absolute SUN count at the same NFE budget. The energy reward consistently lowers average E_{hull} and improves overall stability rate. We also report the number of unique chemical systems among the resulting samples to show that diversity is retained.

Table 1: Single-objective stability search with MatterGen at matched NFE budget, averaged over three seeds. "Null" is DTS with $r(\mathbf{x}_0) = 0$; α is the progressive widening parameter.

α	Reward	# Structures	# SUN	% Stable	SUN Unique Systems	Avg E_{hull}/atom
0.1	Null	5050 ± 29	1311 ± 39	66.5 ± 1.1	912.3 ± 33.7	0.091 ± 0.004
	$-E_{\text{hull}}$	5226 ± 42	1432 ± 73	75.7 ± 0.4	966 ± 12	0.073 ± 0.000
0.3	Null	3500 ± 0	1095 ± 55	65.3 ± 1.5	904.3 ± 29.4	0.1 ± 0.004
	$-E_{\text{hull}}$	3500 ± 0	1209 ± 31	70.2 ± 0.9	971.3 ± 29.0	0.085 ± 0.002
0.5	Null	3000 ± 0	974 ± 27	66.6 ± 1.4	894.7 ± 27.4	0.090 ± 0.003
	$-E_{\text{hull}}$	3000 ± 0	1087 ± 23	68.6 ± 1.3	979.7 ± 15.2	0.087 ± 0.002
MatterGen Baseline		2500	868 ± 11	68.4 ± 0.1	848.3 ± 10.9	0.086 ± 0.000

Results with DiffCSP and inference-time scaling. DiffCSP is roughly $20\times$ faster per sample than MatterGen, which lets us study how DTS scales with the NFE budget from 10^5 to 10^7 total NFEs (Figure 2). Stability rate increases with NFE budget under the E_{hull} reward, while the null-reward control and best-of-N baselines remain roughly flat – a clear and growing inference-time scaling

Table 2: DiffCSP results at matched 1M and 10M NFE budgets.

Method	Structures	Stability (%) ↑	Uniqueness (%) ↑	Novelty (%) ↑	SUN (%) ↑	Mean E_{hull} ↓
Matched 1M NFE budget						
Best-of-N	1,000	30.7 ± 0.9	99.6 ± 0.1	89.4 ± 0.3	21.0 ± 1.3	0.314 ± 0.031
SMC (2 resamples)	554	30.7 ± 0.5	84.9 ± 1.4	88.8 ± 1.0	16.3 ± 0.6	0.315 ± 0.005
SMC (4 resamples)	332	29.7 ± 3.2	87.3 ± 0.9	89.3 ± 4.5	16.7 ± 6.2	0.296 ± 0.020
SVDD-PM / TreeG-SC	725	34.2 ± 1.8	99.4 ± 0.2	88.3 ± 1.1	23.2 ± 2.2	0.724 ± 0.388
TreeG-SD	1,000	33.5 ± 4.8	99.4 ± 0.1	89.2 ± 1.2	23.2 ± 3.4	0.317 ± 0.034
DTS (Null Reward)	1,350	32.0 ± 1.5	99.3 ± 0.1	89.6 ± 0.8	22.5 ± 2.1	0.399 ± 0.149
DTS	1,350	$\mathbf{36.8 \pm 4.9}$	99.2 ± 0.1	87.5 ± 1.4	$\mathbf{25.1 \pm 3.6}$	$\mathbf{0.267 \pm 0.019}$
Matched 10M NFE budget						
Best-of-N	10,000	31.1 ± 0.3	96.9 ± 0.1	$\mathbf{89.2 \pm 0.2}$	20.3 ± 0.1	0.435 ± 0.035
SMC (2 resamples)	5,549	31.3 ± 0.1	84.5 ± 1.1	88.6 ± 1.4	16.8 ± 0.4	0.361 ± 0.013
SMC (4 resamples)	3,328	31.4 ± 2.0	84.1 ± 0.5	88.5 ± 1.3	16.4 ± 0.7	$\mathbf{0.320 \pm 0.028}$
SVDD-PM	7,246	33.3 ± 0.9	$\mathbf{97.4 \pm 0.1}$	88.2 ± 0.0	22.0 ± 0.8	0.574 ± 0.193
TreeG-SD	10,000	31.6 ± 0.5	96.9 ± 0.3	89.1 ± 0.3	20.8 ± 0.0	0.425 ± 0.033
DTS (Null Reward)	15,350	31.1 ± 0.8	95.5 ± 0.2	89.1 ± 0.5	19.7 ± 0.1	0.337 ± 0.036
DTS	15,450	$\mathbf{41.6 \pm 6.8}$	90.4 ± 3.2	84.5 ± 3.4	$\mathbf{24.1 \pm 2.2}$	0.331 ± 0.158

benefit. Uniqueness and novelty decline slightly as compute increases, reflecting concentration of rollouts on high-reward regions.

We also compare against three inference-time steering baselines: SVDD-PM [Li et al., 2024] (which is equivalent to TreeG-SC [Guo et al., 2026] with A=1), TreeG-SD [Guo et al., 2026] and rollout-based SMC [Singhal et al., 2025] with two and four resampling events. All methods use the same pretrained DiffCSP model, E_{hull} reward, and approximately matched 1M and 10M NFE budgets; further details are included in Appendix G.1. The results, shown in Table 2, demonstrate that DTS outperforms all baselines at generating stable, unique and novel crystals.

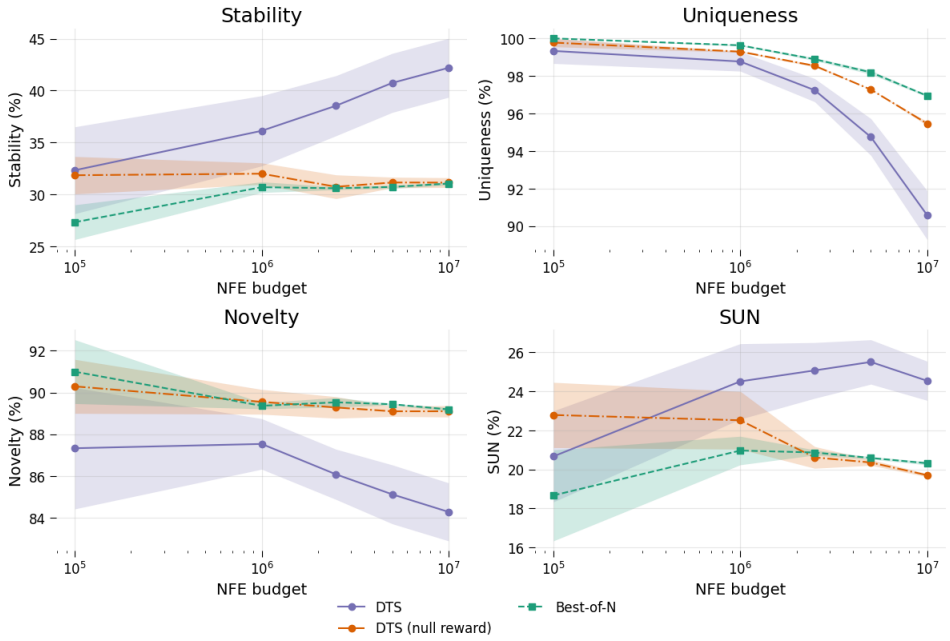


Figure 2: Inference-time scaling of DTS with DiffCSP across NFE budgets, compared with the control null-reward, and Best-of-N baselines. Standard deviations are computed across 3 seeds.

4.2 Symmetry-Conditioned Generation

We test whether DTS can amplify a property-conditional model on a rare, non-differentiable target: matching a specific space group. Both DTS and the best-of-N baseline use the same space-group-conditional MatterGen variant; the reward is the indicator $r(\mathbf{x}_0) = \mathbb{1}[\text{SG}(\mathbf{x}_0) = g^\star]$, with $\text{SG}(\cdot)$ computed using pymatgen [Ong et al., 2013]. We focus here on space group 204 ($Im\bar{3}$), which

appears in the training set with frequency $\sim$0.2%; results for additional rare target groups are in Appendix G.5.

Figure 3 shows cumulative SUN $\cap$ SG 204 counts versus NFE budget. DTS produces roughly an order of magnitude more on-target SUN structures than best-of-N, with the gap widening as compute increases. The unconditional MatterGen, with or without DTS, generates zero structures in SG 204 across the same budget. This contrast highlights the role of each component: the conditional proposal is *necessary* for DTS to access the targeted mode at all (the unconditional base places negligible mass there), but is not *sufficient* to produce on-target samples at useful rates. Search compounds the proposal by reusing partial trajectories that lead into the target region.

4.3 Multi-objective: Dielectric Materials

We consider a multi-objective setting from Riebesell et al. [2024]: simultaneously maximizing bandgap and dielectric constant. These properties are approximately inversely related [Penn, 1962, Robertson, 2005, Onishi and Fu, 2024], inducing a non-convex Pareto front. This task also demonstrates the conditional-model adaptation: MatterGen can be fine-tuned to condition jointly on (bandgap, dielectric constant), used here as a guided proposal inside DTS.

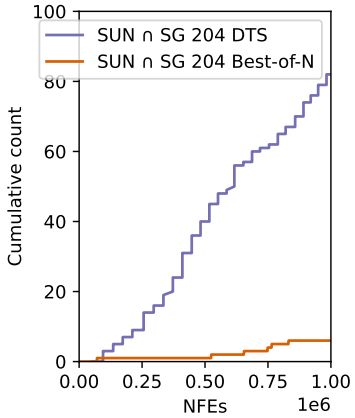


Figure 3: Cumulative count of SUN structures matching target space group 204 ($Im\bar{3}$) versus NFE budget. Both methods use the same space-group-conditional MatterGen variant.

Setup. Both properties are predicted with ALIGNN [Choudhary and DeCost, 2021]. We use a MatterGen base model fine-tuned on labels from this predictor, and run multi-objective DTS at 2.5M NFEs. As a baseline, we condition MatterGen on five target points evenly spaced along the dataset Pareto front and take the union of generated samples at matched NFE budget. We evaluate after filtering to SUN structures, reporting Pareto-front size, hypervolume (HV), R_2 [Hansen and Jaszkiewicz, 1998], and generational distance plus (GD+) [Ishibuchi et al., 2015] against the dataset Pareto front. We also include threshold counts: number of structures whose bandgap $\times$ dielectric constant exceeds fixed thresholds (30, 60, 120, 240) – these values correspond to linear scalarization in log space.

Results. Figure 4 plots SUN samples and Table 3 summarizes metrics. MO-DTS outperforms the baseline at matched NFE budget for most metrics.

Table 3: Summary metrics for the bandgap vs. dielectric constant experiment. $|PF|$ = Pareto front size. Entries are mean $\pm$ std across seeds. Threshold columns count structures where bandgap $\times$ dielectric $\geq$ threshold.

Method	N	$\lvert PF\rvert$	$N_{\geq 30}$	$N_{\geq 60}$	$N_{\geq 120}$	$N_{\geq 240}$	HV↑	GD+↓	R2↓
Training set	27,136	20	4,180	1,920	526	109	0.309	0.000	0.149
MatterGen (cond.)	**665 ± 0**	14 ± 1	262 ± 3	138 ± 6	28 ± 2	4 ± 0	0.118 ± 0.011	0.098 ± 0.024	**0.292 ± 0.002**
MO-DTS	645 ± 40	16 ± 1	**402 ± 54**	**234 ± 27**	**67 ± 12**	**17 ± 4**	**0.173 ± 0.057**	**0.054 ± 0.015**	0.338 ± 0.040

4.4 Density/bulk modulus: partially aligned reward

The previous experiments used either unconditional models or models fine-tuned for the target reward. Conditional DTS is more general, since importance sampling decouples the proposal from the target. To show this, we repurpose an existing MatterGen model conditioned on density as a proposal, and set the target to be generating crystals with a high bulk modulus (300 GPa). In the training set the two correlate (Pearson $r = 0.512$): crystals with density $\geq 15\,\mathrm{g/cm^3}$ are rare ($<2\%$) but about $20\times$ more likely than average to have bulk modulus $\geq 300\,\mathrm{GPa}$, so conditioning on a density of $15\,\mathrm{g/cm^3}$ is a useful but imperfect proxy. The mismatch is that training crystals near the bulk modulus target span a broad density range, so without correction the proposal should raise yield while distorting the density of the resulting crystals; IS should remove the distortion.

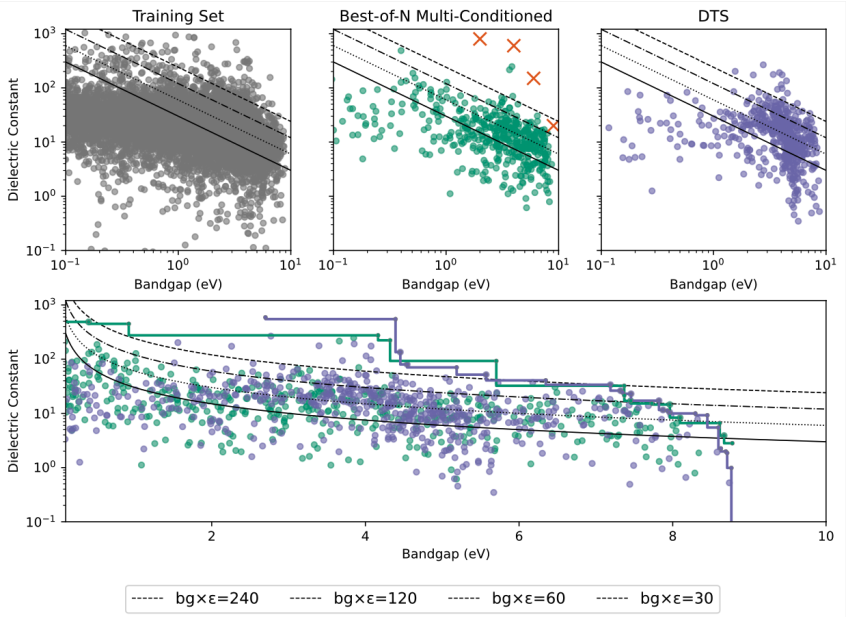


Figure 4: Dielectric constant vs. bandgap for SUN samples at 2.5M NFEs (∼2500 total samples per method). Diagonal lines indicate constant values of bandgap × dielectric, and X markers represent conditioning targets.

We compare four arms (Table 4): DTS with an unconditional model; the density-conditioned proposal alone (best-of-N); conditional DTS without IS; and conditional DTS with IS. All DTS arms use reward $r(\mathbf{x}) = -|K(\mathbf{x}) - 300|/400$ with K the bulk modulus predicted by ALIGNN [Choudhary and DeCost, 2021] in GPa, $\lambda = 20$, $\alpha = 0.3$, and guidance strength $\gamma = 1$ for the conditional proposal; each arm runs 10^6 NFEs over 3 seeds, with metrics computed on SUN crystals only. We report the number of SUN crystals within 30 GPa of the target, the mean absolute error to the target, and the density distortion: the Wasserstein distance W_1 between the density distribution of generated crystals within 30 GPa of the target and that of training crystals in the same range.

Table 4: Results for bulk modulus experiment, ablating conditioning and importance sampling correction (IS).

Model	DTS	IS	SUN crystals within 30 GPa ↑	Bulk MAE to 300 GPa ↓	Near-target density W_1 to training ↓
Unconditional	✓	✗	237	108.23	2.80
Density-Conditioned	✗	✗	61	131.02	4.72
Density-Conditioned	✓	✗	**302**	**62.82**	4.31
Density-Conditioned	✓	✓	260	70.33	**2.15**

The mismatched proposal is useful only together with search: on its own it yields four times fewer near-target SUN crystals than unconditional DTS, since it shifts samples toward dense crystals but does not select for bulk modulus. Conditional DTS without IS outperforms both, but the proposal creates biased samples: the resulting near-target crystals are much denser than training crystals in the same range. Importance correction removes this bias while keeping most of the gain in yield. An existing conditional model can therefore serve as a proposal for a new reward, with the output governed by the reward rather than by the proposal.

5 Related Work

Inference-time steering of diffusion models. Score-modifying approaches include classifier-based [Dhariwal and Nichol, 2021] and classifier-free [Ho and Salimans, 2022] guidance; both approximate

reward-tilted sampling but are biased and require differentiable rewards. Particle-based methods treat denoising as SMC [Singhal et al., 2025]. Methods such as SVDD [Li et al., 2024] and TreeG [Guo et al., 2026] perform a local search: they branch each denoising step into several candidates and keep the best by a value estimate. The closest line to us organizes intermediate states into a tree: Diffusion Tree Sampling [Jain et al., 2025] formalizes this for continuous diffusion as MCTS with soft-Bellman backups and is the basis for our method. PepTune [Tang et al., 2025] develops Monte Carlo Tree Guidance for discrete diffusion in multi-objective peptide design with a Pareto-dominance reward. Our approach operates on continuous diffusion with scalarization-based per-objective backups, and composes with conditional models via importance correction. A separate strand fine-tunes for specific objectives via conditional training [Zeni et al., 2025] or RL [Black et al., 2023, Chen et al., 2025]; our work leaves the model fixed and composes with such fine-tuned variants when available.

Multi-objective generation. Multi-objective generation has a long history in evolutionary algorithms and Bayesian optimization [Daulton et al., 2020]; multi-objective GFlowNets [Jain et al., 2023, Zhu et al., 2023] sample from a Pareto-tilted distribution by conditioning on preferences during training. Diffusion-based methods modify the score or velocity field during a single denoising pass (e.g. PROUD [Yao et al., 2024], ParetoFlow [Yuan et al., 2024], and Preference-Guided Diffusion [Annadani et al., 2025]), use diffusion as proposals [Kong et al., 2024], or wrap diffusion in outer BO loops [Gruver et al., 2023]. Our method differs by using tree-based search over pretrained continuous diffusion models, using per-objective values and backups, yielding the target of Equation (5) and admitting importance-corrected use of conditional models.

6 Discussion

We adapted Diffusion Tree Sampling to crystal generative models in two ways: importance-corrected guided proposals, which let existing property-conditional models steer the search without changing its target, and a multi-objective extension, in which per-objective value backups and a preference sampled per traversal let a single tree search over the Pareto front of competing properties. DTS alone improved stability and scaled with compute, beating best-of-N, but could not reach rare regions in the space of possible materials. This access can be provided by using conditional proposals, but at the cost of tying the output to whatever the proposal was tuned for. This dependence can be solved via importance correction, enabling cases where the proposal and reward are mismatched, or where we want a single tree to capture multiple different preferences.

The main caveat is reward fidelity. All reported rewards use learned surrogates, so search can amplify systematic predictor errors; DFT validation and synthesis remain essential next steps. Importance correction is also most reliable when guided and base models share a backbone; separately trained proposals may require stronger weight control.

More broadly, this method fits the foundation-model setting: a pretrained generative prior, adapted at inference time to objectives that it has not been specifically trained for. This setup benefits from inference-time search, where more compute is spent per candidate to produce higher-quality samples. Materials discovery is a natural fit because generation is cheap next to DFT validation and synthesis, but the same technique could also be applied to other AI for science settings, such as protein and small-molecule design.

Acknowledgment

This project is in part funded by CIFAR AI chairs, NSERC Discovery grant and IVADO. D.L.'s research is additionally supported by FRQNT. Computational resources are provided by Mila and the Digital Research Alliance of Canada.

References

Yashas Annadani, Syrine Belakaria, Stefano Ermon, Stefan Bauer, and Barbara E Engelhardt. Preference-guided diffusion for multi-objective offline optimization. *arXiv preprint arXiv:2503.17299*, 2025.

Marloes Arts, Victor Garcia Satorras, Chin-Wei Huang, Daniel Zugner, Marco Federici, Cecilia Clementi, Frank Noé, Robert Pinsler, and Rianne van den Berg. Two for one: Diffusion models and force fields for coarse-grained molecular dynamics. *Journal of Chemical Theory and Computation*, 19(18):6151–6159, 2023.

Siddharth Betala, Samuel P Gleason, Ali Ramlaoui, Andy Xu, Georgia Channing, Daniel Levy, Clémentine Fourrier, Nikita Kazeev, Chaitanya K Joshi, Sékou-Oumar Kaba, et al. Lemat-genbench: A unified evaluation framework for crystal generative models. *arXiv preprint arXiv:2512.04562*, 2025.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

V Joseph Bowman Jr. On the relationship of the tchebycheff norm and the efficient frontier of multiple-criteria objectives. In *Multiple Criteria Decision Making: Proceedings of a Conference Jouy-en-Josas, France May 21–23, 1975*, pages 76–86. Springer, 1976.

Junwu Chen, Jeff Guo, and Philippe Schwaller. Matinvent: Reinforcement learning for 3d crystal diffusion generation. In *AI for Accelerated Materials Design-ICLR 2025*, 2025.

Kamal Choudhary and Brian DeCost. Atomistic line graph neural network for improved materials property predictions. *npj Computational Materials*, 7(1):185, 2021.

Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective bayesian optimization. *Advances in neural information processing systems*, 33:9851–9864, 2020.

Kalyanmoy Deb, Karthik Sindhya, and Jussi Hakanen. Multi-objective optimization. In *Decision sciences*, pages 161–200. CRC Press, 2016.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Nate Gruver, Samuel Stanton, Nathan Frey, Tim GJ Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew G Wilson. Protein design with guided discrete diffusion. *Advances in neural information processing systems*, 36:12489–12517, 2023.

Yingqing Guo, Yukang Yang, Hui Yuan, and Mengdi Wang. Training-free guidance beyond differentiability: Scalable path steering with tree search in diffusion and flow models. *Advances in Neural Information Processing Systems*, 38:73343–73384, 2026.

Michael Pilegaard Hansen and Andrzej Jaszkiewicz. Evaluating the quality of approximations to the non-dominated set. 1998.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Hisao Ishibuchi, Hiroyuki Masuda, Yuki Tanigaki, and Yusuke Nojima. Modified distance calculation in generational distance and inverted generational distance. In *International conference on evolutionary multi-criterion optimization*, pages 110–125. Springer, 2015.

Moksh Jain, Sharath Chandra Raparthy, Alex Hernández-Garcıa, Jarrid Rector-Brooks, Yoshua Bengio, Santiago Miret, and Emmanuel Bengio. Multi-objective gflownets. In *International conference on machine learning*, pages 14631–14653. PMLR, 2023.

Vineet Jain, Kusha Sareen, Mohammad Pedramfar, and Siamak Ravanbakhsh. Diffusion tree sampling: Scalable inference-time alignment of diffusion models. *arXiv preprint arXiv:2506.20701*, 2025.

Rui Jiao, Wenbing Huang, Peijia Lin, Jiaqi Han, Pin Chen, Yutong Lu, and Yang Liu. Crystal structure prediction by joint equivariant diffusion. *Advances in Neural Information Processing Systems*, 36:17464–17497, 2023.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Lingkai Kong, Yuanqi Du, Wenhao Mu, Kirill Neklyudov, Valentin De Bortoli, Dongxia Wu, Haorui Wang, Aaron Ferber, Yi-An Ma, Carla P Gomes, et al. Diffusion models as constrained samplers for optimization with unknown constraints. *arXiv preprint arXiv:2402.18012*, 2024.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, et al. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. *arXiv preprint arXiv:2408.08252*, 2024.

Zhenqin Li and Harold A Scheraga. Monte carlo-minimization approach to the multiple-minima problem in protein folding. *Proceedings of the National Academy of Sciences*, 84(19):6611–6615, 1987.

Zhuoyuan Li, Siyu Liu, Beilin Ye, David J Srolovitz, and Tongqi Wen. Active learning for conditional inverse design with crystal generation and foundation atomic models. *arXiv preprint arXiv:2502.16984*, 2025.

Kaisa Miettinen. *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media, 1999.

Artem R Oganov and Colin W Glass. Crystal structure prediction using ab initio evolutionary techniques: Principles and applications. *The Journal of chemical physics*, 124(24), 2006.

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.

Yugo Onishi and Liang Fu. Universal relation between energy gap and dielectric constant. *Physical Review B*, 110(15):155107, 2024.

David R Penn. Wave-number-dependent dielectric function of semiconductors. *Physical review*, 128 (5):2093, 1962.

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

Janosh Riebesell, T Surta, Rhys Goodall, Michael Gaultois, and Alpha A Lee. Pushing the pareto front of band gap and permittivity: Ml-guided search for dielectric materials. *arXiv preprint arXiv:2401.05848*, 2024.

John Robertson. High dielectric constant gate oxides for metal oxide si transistors. *Reports on progress in Physics*, 69(2):327, 2005.

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. *arXiv preprint arXiv:2501.06848*, 2025.

Sophia Tang, Yinuo Zhang, and Pranam Chatterjee. Peptune: De novo generation of therapeutic peptides with multi-objective-guided discrete diffusion. *ArXiv*, pages arXiv–2412, 2025.

Zhiwei Tang, Jiangweizhi Peng, Jiasheng Tang, Mingyi Hong, Fan Wang, and Tsung-Hui Chang. Inference-time alignment of diffusion models with direct noise optimization. *arXiv preprint arXiv:2405.18881*, 2024.

Masatoshi Uehara, Yulai Zhao, Chenyu Wang, Xiner Li, Aviv Regev, Sergey Levine, and Tommaso Biancalani. Inference-time alignment in diffusion models with reward-guided generation: Tutorial and review. *arXiv preprint arXiv:2501.09685*, 2025.

David J Wales and Jonathan PK Doye. Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms. *The Journal of Physical Chemistry A*, 101(28):5111–5116, 1997.

David J Wales and Harold A Scheraga. Global optimization of clusters, crystals, and biomolecules. *Science*, 285(5432):1368–1372, 1999.

Han Yang, Chenxi Hu, Yichi Zhou, Xixian Liu, Yu Shi, Jielan Li, Guanzhi Li, Zekun Chen, Shuizhou Chen, Claudio Zeni, et al. Mattersim: A deep learning atomistic model across elements, temperatures and pressures. *arXiv preprint arXiv:2405.04967*, 2024.

Yinghua Yao, Yuangang Pan, Jing Li, Ivor Tsang, and Xin Yao. Proud: Pareto-guided diffusion model for multi-objective generation. *Machine Learning*, 113(9):6511–6538, 2024.

Po-Hung Yeh, Kuang-Huei Lee, and Jun-Cheng Chen. Training-free diffusion model alignment with sampling demons. *arXiv preprint arXiv:2410.05760*, 2024.

Ye Yuan, Can Chen, Christopher Pal, and Xue Liu. Paretoflow: Guided flows in multi-objective optimization. *arXiv preprint arXiv:2412.03718*, 2024.

Sheheryar Zaidi, Michael Schaarschmidt, James Martens, Hyunjik Kim, Yee Whye Teh, Alvaro Sanchez-Gonzalez, Peter Battaglia, Razvan Pascanu, and Jonathan Godwin. Pre-training via denoising for molecular property prediction. *arXiv preprint arXiv:2206.00133*, 2022.

Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Zilong Wang, Aliaksandra Shysheya, Jonathan Crabbé, Shoko Ueda, et al. A generative model for inorganic materials design. *Nature*, 639(8055):624–632, 2025.

Yiheng Zhu, Jialu Wu, Chaowen Hu, Jiahuan Yan, Tingjun Hou, Jian Wu, et al. Sample-efficient multi-objective molecular optimization with gflownets. *Advances in Neural Information Processing Systems*, 36:79667–79684, 2023.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state three contributions (importance-corrected use of conditional foundation models inside DTS, multi-objective DTS, and application to crystal generation), each of which is developed in Sections 3.1, 3.2, and the Application section respectively, and supported by the experiments in Sections 4.1–4.3.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The Discussion section notes that gains depend on the foundation model's prior, that ML rewards are approximations that may be exploited, and that experimental validation of generated materials is left to future work. Variance and degeneracy concerns of importance-weighted estimators are discussed in Appendix **??**, and the regimes in which the per-objective surrogate value bound holds (and where it can fail) are characterized in Appendix F.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The single theoretical result, Proposition F.1, states its assumptions (linear vs. concave monotone scalarization, low-temperature or sharp-posterior regime) explicitly and is proved in full in Appendix F, including a remark on regimes where the bound can fail.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The full algorithm is given in Appendix A; predictor–corrector handling in Appendix D; the foundation models (DiffCSP, MatterGen and its conditional variants), reward predictors (MatterSim, ALIGNN, CHGNet), evaluation hull (Alex-MP), NFE budgets, progressive-widening sweeps, batch sizes, seed counts, baselines, and the SMC configuration are described in Section 4.1–4.3 and Appendix G.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper includes a code release. All foundation models (DiffCSP, MatterGen) and reward predictors (MatterSim, ALIGNN) used are publicly released by their original authors, and the paper specifies hyperparameters and budgets sufficient to reimplement the procedure on top of them.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: No model training is performed in this work; experiments are inference-time only, on top of pretrained foundation models. The relevant search hyperparameters (progressive-widening α and C, inverse temperature λ, batch size, NFE budgets, number of seeds, novelty reference set) and how they were selected (sweeps at a fixed NFE budget, then evaluation across budgets) are reported in Section 4.1 and Appendix G.1; the SMC baseline configuration is also specified.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All main experiments report mean $\pm$ standard deviation across three seeds (Tables 1, 3, 9, and Figure 2).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix G.2 reports approximate wall-clock times on a single A100 (DiffCSP, 10M NFEs) and a single RTX8000 (MatterGen, 1M NFEs) for each method/reward combination, and main-text tables report NFE budgets that proxy compute.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses publicly released pretrained models, public materials databases, and computational property predictors; it involves no human subjects, no scraped personal data, and no privacy-sensitive content.

   Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The introduction motivates the work by the value of focusing inference compute on high-quality candidates for downstream DFT validation and synthesis. The Discussion notes that ML rewards are approximations and may be exploited and that experimental validation is required before any practical claims; the work concerns proposing candidate inorganic crystals at the design stage and has no foreseeable direct path to malicious use beyond the dual-use nature of materials discovery in general.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper releases no new pretrained models, datasets, or generative assets. The proposed method is an inference-time search procedure layered on existing publicly released foundation models and produces candidate inorganic crystal structures, which are not in a high-misuse-risk category.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All foundation models, property predictors, and datasets used (DiffCSP, MatterGen, MatterSim, ALIGNN, pymatgen, MP-20, Alex-MP) are cited at point of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new datasets, model weights, or other assets. The contribution is an algorithm; pseudocode is provided in Appendix A.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human subjects.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not part of the core methodology. The foundation models in this work are diffusion models for crystal structures, not language models.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

A Algorithm

Algorithm 1 Diffusion Tree Sampling with guided proposals and multiple objectives. Unmarked lines are DTS [Jain et al., 2025]; lines tagged [IS] add the importance-corrected guided proposals of Section 3.1, and lines tagged [MO] add multi-objective selection and backup of Section 3.2. Batched execution is described in Section 3.3.

Input: base model p_θ; inverse temperature λ; NFE budget N_{NFE} ;
Input: widening parameters C,α, branching timesteps $\mathcal{T}$
Input: guided proposal q_c (default $q_c = p_\theta$) [IS]
Input: objectives $\mathbf{r} = (r_1, \dots, r_m)$; scalarization s; preferences $p(w)$; condition map $w \mapsto c_w$ (default $m = 1$, $s = \mathrm{id}$, $c_w \equiv c$) [MO]
Initialize dummy root $\mathbf{x}_{T+1}$ with $p(\mathbf{x}_T \mid \mathbf{x}_{T+1}) = p_T$; $\hat{\mathbf{v}}(\mathbf{x}_{T+1}) \leftarrow \mathbf{0}$, $N(\mathbf{x}_{T+1}) \leftarrow 1$, $\mathcal{C}(\mathbf{x}_{T+1}) \leftarrow \emptyset$
nfe $\leftarrow 0$
while nfe $< N_{\mathrm{NFE}}$ **do**
  // Initialize path
  $P \leftarrow \{\mathbf{x}_{T+1}\}$; $t \leftarrow T + 1$
  $w \sim p(w)$; $c \leftarrow c_w$ [MO]
  // Selection: descend until reaching a terminal or an expandable node
  while $t > 0$ **do**
    if $t \in \mathcal{T}$ **and** $|\mathcal{C}(\mathbf{x}_t)| < C\, N(\mathbf{x}_t)^\alpha$ **then**
      break
    end if
    $\mathbf{x}_{t-1} \leftarrow \mathbf{z} \in \mathcal{C}(\mathbf{x}_t)$ with probability $\dfrac{\eta(\mathbf{z})\, \exp\big(\lambda\, s(\hat{\mathbf{v}}(\mathbf{z}) \mid w)\big)}{\sum_{\mathbf{z}' \in \mathcal{C}(\mathbf{x}_t)} \eta(\mathbf{z}')\, \exp\big(\lambda\, s(\hat{\mathbf{v}}(\mathbf{z}') \mid w)\big)}$ [IS] [MO]
    $P \leftarrow P \cup \{\mathbf{x}_{t-1}\}$; $t \leftarrow t - 1$
  end while
  // Expansion and rollout: grow a new path to $t = 0$, adding every node to the tree
  while $t > 0$ **do**
    $\mathbf{x}_{t-1} \sim q_c(\cdot \mid \mathbf{x}_t)$; nfe $\leftarrow$ nfe $+ 1$ [IS]
    $\hat{\mathbf{v}}(\mathbf{x}_{t-1}) \leftarrow \mathbf{0} \in \mathbb{R}^m$; $N(\mathbf{x}_{t-1}) \leftarrow 1$; $\mathcal{C}(\mathbf{x}_t) \leftarrow \mathcal{C}(\mathbf{x}_t) \cup \{\mathbf{x}_{t-1}\}$ [MO]
    $\eta(\mathbf{x}_{t-1}) \leftarrow p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \,/\, q_c(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ [IS]
    $P \leftarrow P \cup \{\mathbf{x}_{t-1}\}$; $t \leftarrow t - 1$
  end while
  // Terminal evaluation
  $\hat{\mathbf{v}}(\mathbf{x}_0) \leftarrow \mathbf{r}(\mathbf{x}_0)$ [MO]
  // Backup: per-objective self-normalized soft Bellman update
  for each $\mathbf{x}_t \in P$ with $t \geq 1$, from $\mathbf{x}_1$ up to $\mathbf{x}_{T+1}$ **do**
    for $i = 1, \dots, m$: $\hat{v}_i(\mathbf{x}_t) \leftarrow \dfrac{1}{\lambda} \log \dfrac{\sum_{\mathbf{z} \in \mathcal{C}(\mathbf{x}_t)} \eta(\mathbf{z})\, e^{\lambda \hat{v}_i(\mathbf{z})}}{\sum_{\mathbf{z} \in \mathcal{C}(\mathbf{x}_t)} \eta(\mathbf{z})}$ [IS] [MO]
    $N(\mathbf{x}_t) \leftarrow N(\mathbf{x}_t) + 1$
  end for
end while
return terminal nodes $\{\mathbf{x}_0\}$ of the tree

B Connection to Basin Hopping and Monte Carlo Minimization

DTS bears a structural resemblance to a family of classical global optimization methods used in computational chemistry, most notably Monte Carlo Minimization [Li and Scheraga, 1987] and its successor, basin hopping [Wales and Doye, 1997, Wales and Scheraga, 1999]. These algorithms iterate a perturb–relax–accept cycle: a candidate configuration is randomly displaced, mapped to the bottom of the resulting basin via a deterministic local minimizer, and accepted or rejected by a Metropolis criterion at temperature T on the minimized energies.

Different components of DTS have natural counterparts in this framework. Tree expansion of a node x_t plays the role of perturbation, with the noise level t controlling an effective step size; the unrolled

reverse process from t to 0 plays the role of local minimization, an interpretation supported by the established equivalence between denoising score matching and learning a Boltzmann force field over equilibrium structures [Zaidi et al., 2022, Arts et al., 2023]; the soft Bellman backup with inverse temperature λ plays the role of Metropolis acceptance, with the greedy variant of DTS recovering monotonic basin hopping.

Several features distinguish our method from these classical methods. The search tree retains and shares value estimates across all visited nodes rather than maintaining a single chain. Rather than just minimizing energy via structural relaxation, using a generative model enables swapping of atom types in a structure. Lastly, the equivalence between score matching and force fields does not apply to property-conditional models, which steer generation towards specific properties rather than just energy minima.

C Illustrative experiments

C.1 Gaussian-mixture toy illustration of the double-tilt problem

Figure 5 illustrates the double-tilt problem and the importance correction of Section 3.1 on a Gaussian mixture. The unconditional distribution is an equal-weight four-component mixture; with linear reward $r(x) = x$ the analytic target $\pi(x) \propto p(\mathbf{x}) \exp(\lambda f(x))$ concentrates on the higher-x modes. A conditional proposal $p(\mathbf{x} \mid c)$ biased toward positive x aligns better with π, but naively running DTS with this proposal double-tilts and distorts the distribution. Multiplying the reward weights by the importance ratio recovers π to within a few percent at every mode.

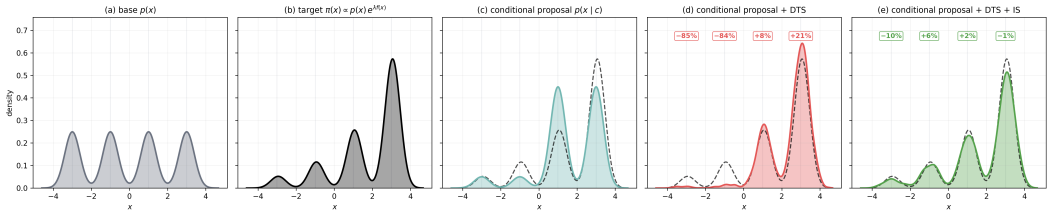


Figure 5: (a) The unconditional distribution $p(\mathbf{x})$ is an equal-weight four-component Gaussian mixture; (b) with linear reward $f(x) = x$, the analytic target $\pi(x) \propto p(\mathbf{x}) \exp(\lambda f(x))$ concentrates mass on higher-$\mathbf{x}$ modes; (c) the conditional proposal $p(\mathbf{x}|c)$ is biased toward positive $\mathbf{x}$ and aligns better with the target (dashed black); (d) naively running DTS with $p(\mathbf{x}|c)$ as the proposal double-tilts the distribution: the two left modes collapse by roughly 85% and the rightmost mode overshoots by 21%; (e) multiplying the reward weights by the importance ratio $p(\mathbf{x})/p(\mathbf{x}|c)$ recovers π to within a few percent at every mode.

C.2 Illustration on a Gaussian mixture in multi-objective setting

Figure 6 illustrated how our method handles multiple conflicting objectives by scalarizing different preferences over the objectives in a single shared tree. The unconditional distribution is a two-dimensional Gaussian $p(x, y) = \mathcal{N}((0, 0); I)$. The two objectives $f_1(x, y) = x$ and $f_2(x, y) = y$ with corresponding weights w_1 and w_2 tilt the distribution to give the target $\pi(x, y) \propto p(x, y) \exp(w_1 x + w_2 y)$. Different preferences $\mathbf{w} = (w_1, w_2)$ define different target distributions, and our aim is to produce samples from the Pareto front that covers all possible preferences. For illustrative purposes, we fix three specific preferences.

The unconditional model, even with DTS, fails to produce samples from the target distribution since the proposal model does not have high mass in the target region. We use a conditional model that is conditioned on $(w_1 x + w_2 y)$ to produce samples closer to the target region and use it as a guided proposal within DTS. However, naively replacing the proposal results in *double-tilting*, so the samples lie outside the target distribution. We fix this by using an importance sampling correction (as described in Section 3.1) that accounts for the conditional model.

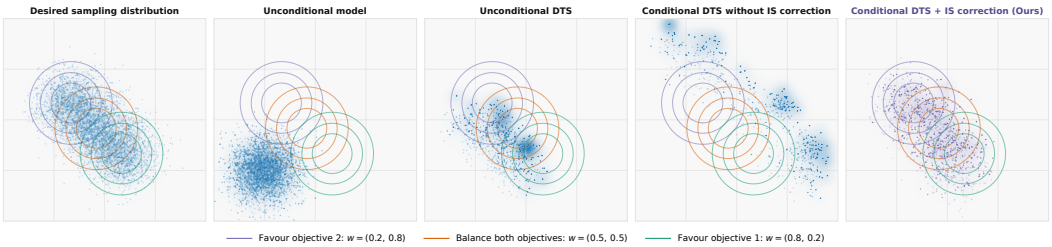


Figure 6: (a) The target distribution is a mixture of different preferences over the tilted distribution $\pi(x, y) \propto p(x, y)\ \exp(w_1 x + w_2 y)$; (b) the unconditional distribution $p(x, y)$ is a two-dimensional standard Gaussian; (c) using DTS with the unconditional proposal fails to sample accurately from the target distribution due to low density of the base distribution in this region; (d) naively running DTS with a conditional model $p(x, y|c)$ as the proposal double-tilts the distribution; (e) multiplying the reward weights by the importance ratio $p(x, y)/p(x, y|c)$ produces samples that align better with the target distribution.

D Predictor–corrector and other multi-stage samplers

The discussion in Section 3.1 assumes a single-step reverse transition. In practice, modern diffusion samplers often implement a *multi-stage* update between t and $t-1$, for improved stability and sample quality. A common pattern is *predictor–corrector* (PC) sampling, where a predictor step performs a discretized reverse-SDE/ODE update, and a corrector step applies one or more refinement moves (e.g., Langevin/MCMC steps) that approximately preserve the desired marginal at the new noise level. More broadly, modern samplers may use higher-order solvers, adaptive step sizes, or alternating deterministic/stochastic updates; our method remains valid as long as we can write down (or evaluate up to a constant) the probability of the composite transition used by the base sampler.

To incorporate PC sampling into guided proposals and importance correction, we treat the overall $t \to t-1$ move as a composition through an intermediate state $\tilde{\mathbf{x}}_{t-1}$:

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \int p_{\text{pred}}(\tilde{\mathbf{x}}_{t-1} \mid \mathbf{x}_t)\, p_{\text{corr}}(\mathbf{x}_{t-1} \mid \tilde{\mathbf{x}}_{t-1})\, d\tilde{\mathbf{x}}_{t-1},$$

with an analogous guided/conditional composite transition where $q_{\text{pred},w}(\cdot \mid \mathbf{x}_t) := p_{\text{pred}}(\cdot \mid \mathbf{x}_t, c(w))$ and similarly for $q_{\text{corr},w}$. In the tree expansion/rollout we explicitly sample $\tilde{\mathbf{x}}_{t-1}$ and then $\mathbf{x}_{t-1}$, and use a factorized importance ratio consistent with this sampling path:

$$\frac{p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)}{q_w(\mathbf{x}_{t-1} \mid \mathbf{x}_t)} \approx \frac{p_{\text{pred}}(\tilde{\mathbf{x}}_{t-1} \mid \mathbf{x}_t)}{q_{\text{pred},w}(\tilde{\mathbf{x}}_{t-1} \mid \mathbf{x}_t)} \times \frac{p_{\text{corr}}(\mathbf{x}_{t-1} \mid \tilde{\mathbf{x}}_{t-1})}{q_{\text{corr},w}(\mathbf{x}_{t-1} \mid \tilde{\mathbf{x}}_{t-1})}. \tag{6}$$

This replaces the single-step ratio in Equation (3). Although the right-hand side is not the marginal density ratio for $\mathbf{x}_{t-1}$ alone, it is the exact path-space importance weight for the joint augmented transition $(\tilde{\mathbf{x}}_{t-1}, \mathbf{x}_{t-1})$ actually sampled by the algorithm; the $\approx$ symbol reflects that the marginalization over $\tilde{\mathbf{x}}_{t-1}$ is performed implicitly by sampling. In particular, if the corrector is shared between conditional and unconditional samplers (i.e., $q_{\text{corr},w} = p_{\text{corr}}$), the corrector ratio cancels and only the predictor ratio is needed. When both predictor and corrector differ (e.g., guided score used in both stages), Equation (6) yields the appropriate weight for maintaining consistency with the unconditional-base target while still leveraging guided proposals during the full sampling procedure.

E Scalarizations

As explained in Section 3.2, a standard way to operationalize multi-objective optimization is *scalarization*: decompose it into a family of single-objective problems of the form $\max_{\mathbf{x}} r(\mathbf{x} \mid w)$, where $w \in \Delta^{m-1} := \{w \in \mathbb{R}^m_{\geq 0} : \sum_i w_i = 1\}$ encodes preferences and $r(\mathbf{x} \mid w) := s(\mathbf{r}(\mathbf{x}) \mid w)$ for scalarization s. The simplest choice for s is the weighted sum, $s(\mathbf{a} \mid w) = \sum_{i=1}^m w_i\, a_i$. An important alternative is (weighted) *Tchebycheff* scalarization [Bowman Jr, 1976], which measures the worst (weighted) deviation from an *ideal* vector $\mathbf{z} \in \mathbb{R}^m$:

$$s(\mathbf{a} \mid w, \mathbf{z}) \;=\; -\max_{i \in \{1,\ldots,m\}} w_i\, |a_i - z_i|, \tag{7}$$

i.e., maximizing $s(\cdot \mid w, \boldsymbol{z})$ is equivalent to minimizing the worst weighted distance to the ideal. This encourages balanced progress across objectives and can recover Pareto-optimal points that may be missed by linear scalarization on non-convex fronts. Both linear and Tchebycheff scalarizations can be unified via a generalized p-norm form,

$$s(\mathbf{a} \mid w, \boldsymbol{z}) \;=\; -\|w \odot |\mathbf{a} - \mathbf{z}|\|_p, \tag{8}$$

where $\odot$ is elementwise multiplication. Linear and Tchebycheff scalarizations are recovered for $p = 1$ and $p = \infty$ respectively. More generally, other weight-parameterized scalarizations (e.g., augmented Tchebycheff, PBI, achievement scalarizing functions) can be viewed as different ways to sweep the Pareto front. Our method is *indifferent* to the particular scalarization: any mapping $w \mapsto s(\cdot \mid w)$ is compatible as long as we can systematically choose w from a specified set/distribution and evaluate the resulting scalar objective.

F Properties of the per-objective value surrogate

Scalarizing only at selection time enables computation sharing across all preferences. The exact soft value of the scalarized reward, $v_t^\star(\mathbf{x}_t \mid w) := \frac{1}{\lambda} \log \mathbb{E}[\exp(\lambda\, r(\mathbf{x}_0 \mid w)) \mid \mathbf{x}_t]$, would instead require a separate backup tree for every w. The per-objective surrogate of $v_t(\mathbf{x}_t|w)$ relates to it as follows.

Proposition F.1. *Let $v_t^\star(\mathbf{x}_t \mid w)$ be the exact scalarized soft value, and let*

$$v_{t,i}(\mathbf{x}_t) \;:=\; \tfrac{1}{\lambda} \log \mathbb{E}_{p_\theta(\mathbf{x}_0|\mathbf{x}_t)}\big[\exp(\lambda\, r_i(\mathbf{x}_0))\big], \tag{9}$$

be the per-objective values used by MO-DTS, with $v_t(\mathbf{x}_t|w) \coloneqq s(\mathbf{v}_t(\mathbf{x}_t)|w)$ its scalarization. Assume the rewards r_i are bounded. Then:

1. *For* linear *scalarization, $v_t^\star(\mathbf{x}_t \mid w) \le v_t(\mathbf{x}_t \mid w)$ for every $\lambda > 0$.*
2. *For any continuous scalarization that is non-decreasing in each objective, $v_t^\star$ is upper bounded by v_t in the low-temperature limit:*

$$\lim_{\lambda\to\infty} v_t^\star(\mathbf{x}_t \mid w) \;\le\; \lim_{\lambda\to\infty} v_t(\mathbf{x}_t \mid w). \tag{10}$$

3. *For any continuous scalarization, the approximation becomes exact at $t = 0$:*

$$v_0(\mathbf{x}_0 \mid w) = v_0^\star(\mathbf{x}_0 \mid w), \tag{11}$$

and, whenever the conditional uncertainty in $\mathbf{r}(\mathbf{x}_0)$ given $\mathbf{x}_t$ vanishes as $t \to 0$, for every fixed $\lambda > 0$,

$$|v_t(\mathbf{x}_t \mid w) - v_t^\star(\mathbf{x}_t \mid w)| \to 0. \tag{12}$$

Proof. Fix $\mathbf{x}_t$ and let all expectations and essential suprema be under $p_\theta(\mathbf{x}_0 \mid \mathbf{x}_t)$. Write $R_i := r_i(\mathbf{x}_0)$ and $\mathbf{r} := (R_1, \dots, R_m)$.

(i) Linear scalarization, all λ. Assume $s(\mathbf{a} \mid w) = \sum_i w_i a_i$ with $w_i \ge 0$ and $\sum_i w_i = 1$, and set $X_i := \exp(\lambda R_i)$. Then

$$\mathbb{E}[\exp(\lambda s(\mathbf{r} \mid w))] \;=\; \mathbb{E}\Big[\textstyle\prod_i X_i^{w_i}\Big]. \tag{13}$$

Hölder's inequality, with conjugate exponents $1/w_i$ for the positive weights, gives

$$\mathbb{E}\Big[\textstyle\prod_i X_i^{w_i}\Big] \;\le\; \textstyle\prod_i \mathbb{E}[X_i]^{w_i}. \tag{14}$$

Taking $\frac{1}{\lambda}\log$ on both sides yields

$$v_t^\star(\mathbf{x}_t \mid w) \;\le\; \sum_i w_i\, \tfrac{1}{\lambda} \log \mathbb{E}[X_i] \;=\; \sum_i w_i\, v_{t,i}(\mathbf{x}_t) \;=\; v_t(\mathbf{x}_t \mid w). \tag{15}$$

Equality in Equation (14) holds iff the active tilted variables are almost surely proportional, equivalently $R_i - R_j$ is almost surely constant for all pairs with $w_i, w_j > 0$.

A second-order Taylor expansion of the cumulant generating function gives, for bounded rewards,

$$v_{t,i}(\mathbf{x}_t) = \mathbb{E}[r_i \mid \mathbf{x}_t] + \tfrac{\lambda}{2}\operatorname{Var}(r_i \mid \mathbf{x}_t) + O(\lambda^2), \tag{16}$$

$$v_t^\star(\mathbf{x}_t \mid w) = \mathbb{E}[r(\mathbf{x}_0 \mid w) \mid \mathbf{x}_t] + \tfrac{\lambda}{2}\operatorname{Var}(r(\mathbf{x}_0 \mid w) \mid \mathbf{x}_t) + O(\lambda^2). \tag{17}$$

Under linear scalarization, $\mathbb{E}[r(\mathbf{x}_0 \mid w) \mid \mathbf{x}_t] = \sum_i w_i \mathbb{E}[r_i \mid \mathbf{x}_t]$, so the zeroth-order terms agree and

$$v_t(\mathbf{x}_t \mid w) - v_t^\star(\mathbf{x}_t \mid w) \;=\; \tfrac{\lambda}{2}\Big(\sum_i w_i \operatorname{Var}(r_i \mid \mathbf{x}_t) - \operatorname{Var}(\textstyle\sum_i w_i r_i \mid \mathbf{x}_t)\Big) + O(\lambda^2), \tag{18}$$

whose first-order coefficient is non-negative by convexity of variance. The exact non-negativity for all λ is the Hölder bound above.

(ii) Monotone scalarizations, low temperature. We use the standard log-sum-exp limit: if a random variable Y is bounded above, then

$$\lim_{\lambda\to\infty} \frac{1}{\lambda}\log \mathbb{E}[\exp(\lambda Y)] \;=\; \operatorname{ess\,sup} Y. \tag{19}$$

Indeed, the upper bound follows from $\mathbb{E}e^{\lambda Y} \le e^{\lambda \operatorname{ess\,sup} Y}$; for any $\epsilon > 0$, the event $Y \ge \operatorname{ess\,sup} Y - \epsilon$ has positive probability, giving the matching lower bound after dividing by λ and taking $\lambda \to \infty$.

Applying Equation (19) to $Y = s(\mathbf{r} \mid w)$ gives

$$\lim_{\lambda\to\infty} v_t^\star(\mathbf{x}_t \mid w) \;=\; \operatorname{ess\,sup} s(\mathbf{r} \mid w). \tag{20}$$

Applying it componentwise gives $v_{t,i}(\mathbf{x}_t) \to \operatorname{ess\,sup} R_i$. By continuity of $s(\cdot \mid w)$,

$$\lim_{\lambda\to\infty} v_t(\mathbf{x}_t \mid w) \;=\; s\big((\operatorname{ess\,sup} R_1, \ldots, \operatorname{ess\,sup} R_m) \mid w\big). \tag{21}$$

Since $R_i \le \operatorname{ess\,sup} R_i$ almost surely for every i, coordinatewise monotonicity implies $s(\mathbf{r} \mid w) \le s\big((\operatorname{ess\,sup} R_1, \ldots, \operatorname{ess\,sup} R_m) \mid w\big)$ almost surely. Therefore

$$\operatorname{ess\,sup} s(\mathbf{r} \mid w) \;\le\; s\big((\operatorname{ess\,sup} R_1, \ldots, \operatorname{ess\,sup} R_m) \mid w\big), \tag{22}$$

which proves the claimed low-temperature optimistic bound.

(iii) Low-noise limit. At $t = 0$, the conditional law $p_\theta(\mathbf{x}_0' \mid \mathbf{x}_0)$ is a point mass at $\mathbf{x}_0$, so

$$v_{0,i}(\mathbf{x}_0) = r_i(\mathbf{x}_0), \qquad v_0(\mathbf{x}_0 \mid w) = v_0^\star(\mathbf{x}_0 \mid w) = s(\mathbf{r}(\mathbf{x}_0) \mid w). \tag{23}$$

For the limiting statement, consider any sequence of low-noise conditional laws for which $\mathbf{r} \to \bar{\mathbf{r}}$ in probability and the rewards remain bounded. By dominated convergence,

$$\mathbb{E}[\exp(\lambda R_i)] \to \exp(\lambda \bar{r}_i), \qquad \mathbb{E}[\exp(\lambda s(\mathbf{r} \mid w))] \to \exp(\lambda s(\bar{\mathbf{r}} \mid w)), \tag{24}$$

for every fixed $\lambda > 0$. Thus $v_{t,i}(\mathbf{x}_t) \to \bar{r}_i$ and $v_t^\star(\mathbf{x}_t \mid w) \to s(\bar{\mathbf{r}} \mid w)$. Continuity of $s(\cdot \mid w)$ then gives $v_t(\mathbf{x}_t \mid w) = s(\mathbf{v}_t(\mathbf{x}_t) \mid w) \to s(\bar{\mathbf{r}} \mid w)$, so $|v_t(\mathbf{x}_t \mid w) - v_t^\star(\mathbf{x}_t \mid w)| \to 0$. □

Remark on nonlinear scalarizations. The low-temperature statement above is intentionally asymptotic. For general nonlinear scalarizations, $v_t^\star(\mathbf{x}_t \mid w) \le v_t(\mathbf{x}_t \mid w)$ need not hold at every finite λ. The monotonicity condition is also essential: negative-distance Tchebycheff scalarization, $s(\mathbf{a} \mid w, z) = -\max_i w_i|a_i - z_i|$, is monotone only on domains where each objective lies on the same side of the ideal point z_i and improvement means moving toward z_i. If feasible rewards can overshoot the ideal, the componentwise essential supremum may move away from the ideal, and the optimistic bound can fail. In applications that use such scalarizations, the proposition applies either after restricting to the monotone region or after redefining the backed-up coordinates as monotone improvement scores relative to the target.

Table 5: Approximate wall-clock times on a single A100 for the 10M NFE budget for DiffCSP, with a batch size of 50.

Method	Reward	Time
DTS	-E_{hull}	$\approx$ 5.9 h
DTS	0	$\approx$ 5.2 h
Best-of-N baseline	0	$\approx$ 1.5 h
SMC	-E_{hull}	$\approx$ 2.2 h

Table 6: Approximate wall-clock times on a single RTX8000 for the 1M NFE budget for MatterGen, with a batch size of 50.

Method	Reward	Time
DTS	-E_{hull}	$\approx$ 3.2 h
DTS	0	$\approx$ 3.0 h
Best-of-N baseline	0	$\approx$ 2.7 h

G Additional Experimental Details and Results

G.1 DiffCSP

For the DiffCSP experiment, we sweep progressive-widening parameters α, C, and inverse temperature λ at a 1M-NFE budget, selecting the configuration that maximizes the stability metric. The chosen configuration — $\alpha = 0.3$, $C = 0.5$ and $\lambda = 10$ — is then evaluated at all reported NFE budgets. The batch size is set to 50. Novelty is computed against DiffCSP's MP-20 training set.

We compare against SMC [Singhal et al., 2025], SVDD-PM [Li et al., 2024], and TreeG-SD [Guo et al., 2026], which are each training-free and gradient-free steering methods for diffusion models. SVDD-PM branches each particle into 10 next-state candidates and scores their Tweedie $\hat{x}_0$ estimates. TreeG-SD samples 10 perturbed clean targets and moves toward the highest-reward candidate. In our DiffCSP adaptation, TreeG-SD perturbs fractional coordinates and lattice variables while holding atom types fixed, since Gaussian perturbations of discrete type logits would produce arbitrary compositions. Because Tweedie estimates at high noise levels generally do not represent physically meaningful crystals, we activate branching after certain noise levels: $t \leq 900$ for SVDD-PM and $t \leq 600$ for TreeG-SD, where t decreases from 1000 (most noise) to 0 (no noise). SMC instead uses full terminal rollouts to calculate its resampling potentials. SMC is run with 400 particles over 1000 denoising steps, resampling every 20 steps for $t < 990$. At each resample point, the rewards are estimated by a full rollout to $t = 0$, resulting in $\approx$ 10M NFEs for rollout-based reward estimation. All reported NFE budgets fall within 2 % of Best-of-N's nominal 10M-NFE budget.

G.2 Wall-clock time comparisons

In Table 5, we compare the wall-clock times for generating samples for all methods at $\approx$ 10M NFEs. For DiffCSP, the overhead is mostly from reward computation since the denoising itself is very fast. In Table 6, we compare the wall-clock times for 1M NFEs using MatterGen. MatterGen is much slower, so the reward calculation time makes less of a difference.

G.3 Matched-output count results

Due to trajectory reuse, DTS produces more candidate structures than regular best-of-N sampling. As such, we present results for the stability experiment of Section 4.1 at a fixed number of output samples, rather than presenting results over all samples created using the same NFEs.

MatterGen. All methods are evaluated at $N = 2500$ terminal candidates, matching the baseline. Two selection rules are shown: first-N by NFE (stopping the search early) and best-N by predicted E_{hull}.

Table 7: MatterGen results with matched output counts ($N = 2500$), using first-N and best-N selection.

α	Method	Selection	# SUN	Stable (%)	SUN unique systems	Avg. E_{hull}/atom
—	MatterGen baseline	All 2500	868.0	68.4	**848.3**	0.0858
0.1	DTS (Null-Reward)	First-N	733.0 ± 26.8	66.9 ± 1.3	544.0 ± 25.2	0.0904 ± 0.0038
0.3	DTS (Null-Reward)	First-N	789.3 ± 32.7	65.7 ± 1.5	673.7 ± 17.6	0.0911 ± 0.0039
0.5	DTS (Null-Reward)	First-N	813.3 ± 24.0	66.7 ± 1.4	749.7 ± 23.4	0.0901 ± 0.0035
0.1	DTS + E_{hull} reward	First-N	734.3 ± 52.2	$\mathbf{72.7 \pm 0.7}$	533.0 ± 19.1	$\mathbf{0.0792 \pm 0.0007}$
0.3	DTS + E_{hull} reward	First-N	866.0 ± 35.5	70.6 ± 0.9	714.7 ± 27.8	0.0843 ± 0.0021
0.5	DTS + E_{hull} reward	First-N	$\mathbf{921.0 \pm 22.5}$	69.3 ± 1.3	833.3 ± 14.9	0.0861 ± 0.0028
0.1	DTS (Null-Reward)	Best-N	859.0 ± 30.4	$\mathbf{100.0 \pm 0.0}$	629.3 ± 33.9	0.0319 ± 0.0006
0.3	DTS (Null-Reward)	Best-N	1095.3 ± 55.4	91.4 ± 2.2	904.3 ± 29.4	0.0500 ± 0.0007
0.5	DTS (Null-Reward)	Best-N	973.7 ± 26.9	80.0 ± 1.7	894.7 ± 27.4	0.0599 ± 0.0020
0.1	DTS + E_{hull} reward	Best-N	710.3 ± 25.6	$\mathbf{100.0 \pm 0.0}$	506.3 ± 13.1	$\mathbf{0.0185 \pm 0.0023}$
0.3	DTS + E_{hull} reward	Best-N	$\mathbf{1207.0 \pm 29.8}$	98.3 ± 1.3	969.7 ± 27.9	0.0454 ± 0.0015
0.5	DTS + E_{hull} reward	Best-N	1087.3 ± 22.8	82.4 ± 1.5	$\mathbf{979.7 \pm 15.2}$	0.0581 ± 0.0024

Table 8: Matched-output comparisons with 1,000 candidates at the 1M-NFE budget and 10,000 candidates at the 10M-NFE budget.

Selection	Method	Stability (%) ↑	SUN (%) ↑	Mean E_{hull} ↓
Matched 1,000 output candidates				
First 1,000	Best-of-N	30.7 ± 0.9	21.0 ± 1.3	0.314 ± 0.031
	DTS (Null-Reward)	31.3 ± 0.3	22.2 ± 1.3	0.349 ± 0.062
	DTS	$\mathbf{35.2 \pm 5.6}$	$\mathbf{24.2 \pm 4.3}$	$\mathbf{0.278 \pm 0.027}$
	TreeG-SD	33.5 ± 4.8	23.2 ± 3.4	0.317 ± 0.034
Lowest E_{hull} 1,000	Best-of-N	30.7 ± 0.9	21.0 ± 1.3	0.314 ± 0.031
	DTS (Null-Reward)	43.2 ± 2.0	30.4 ± 2.8	0.145 ± 0.002
	DTS	$\mathbf{49.7 \pm 6.6}$	$\mathbf{33.9 \pm 4.8}$	$\mathbf{0.123 \pm 0.018}$
	TreeG-SD	33.5 ± 4.8	23.2 ± 3.4	0.317 ± 0.034
Formula-diverse 1,000	Best-of-N	30.7 ± 0.9	21.0 ± 1.3	0.314 ± 0.031
	DTS (Null-Reward)	41.9 ± 1.8	29.9 ± 2.8	0.151 ± 0.002
	DTS	$\mathbf{48.3 \pm 6.4}$	$\mathbf{33.6 \pm 4.7}$	$\mathbf{0.128 \pm 0.018}$
	TreeG-SD	33.5 ± 4.8	23.2 ± 3.4	0.317 ± 0.034
Matched 10,000 output candidates				
First 10,000	Best-of-N	31.1 ± 0.3	20.3 ± 0.1	0.435 ± 0.035
	DTS (Null-Reward)	30.3 ± 0.6	19.6 ± 0.9	0.338 ± 0.020
	DTS	$\mathbf{40.5 \pm 6.6}$	$\mathbf{24.4 \pm 2.1}$	$\mathbf{0.317 \pm 0.125}$
	TreeG-SD	31.6 ± 0.5	20.8 ± 0.0	0.425 ± 0.033
Lowest E_{hull} 10,000	Best-of-N	31.1 ± 0.3	20.3 ± 0.1	0.435 ± 0.035
	DTS (Null-Reward)	47.8 ± 1.3	30.3 ± 0.2	0.121 ± 0.003
	DTS	$\mathbf{64.2 \pm 10.4}$	$\mathbf{37.3 \pm 3.5}$	$\mathbf{0.080 \pm 0.022}$
	TreeG-SD	31.6 ± 0.5	20.8 ± 0.0	0.425 ± 0.033
Formula-diverse 10,000	Best-of-N	31.1 ± 0.3	20.3 ± 0.1	0.435 ± 0.035
	DTS (Null-Reward)	40.5 ± 0.9	29.0 ± 0.3	0.149 ± 0.002
	DTS	$\mathbf{50.0 \pm 5.5}$	$\mathbf{35.7 \pm 3.3}$	$\mathbf{0.120 \pm 0.014}$
	TreeG-SD	31.6 ± 0.5	20.8 ± 0.0	0.425 ± 0.033

These results show that given the same NFE but selecting the highest-reward samples, DTS is able to generate far more SUN candidates. Alternatively, for the same target number of SUN crystals, DTS is able to generate using fewer NFEs. However, using a smaller value of α leads to more trajectory reuse and less diversity, which can result in fewer SUN crystals.

DiffCSP. We evaluate a matched output count of 1,000 and 10,000 candidates at the 1M- and 10M-NFE budgets, respectively. We use three protocols to select these N candidates: (i) the first N structures generated, which involves no post-hoc energy selection; (ii) the N structures with the lowest E_{hull}; and (iii) a formula-diverse selection that first maximizes distinct chemical-formula coverage and then ranks candidates within each formula by E_{hull}.

SVDD-PM and SMC are omitted only from these matched-output comparisons because they produce fewer than 1,000 and 10,000 terminal structures at the corresponding NFE budgets. The first-N

Table 9: Summary metrics for all evaluated methods on the dielectric/bandgap multi-objective task. $|PF|$ = Pareto front size. For methods with multiple seeds, entries are mean $\pm$ std across seeds. Threshold columns count structures where bandgap $\times$ dielectric $\geq$ threshold.

Method	N	$\lvert PF\rvert$	$N_{\geq 30}$	$N_{\geq 60}$	$N_{\geq 120}$	$N_{\geq 240}$	HV↑	GD+↓	IGD+↓	R2↓
Training set	27,136	20	4,180	1,920	526	109	0.309	0.000	0.000	0.149
MatterGen (cond.)	665 ± 0	14 ± 1	262 ± 3	138 ± 6	28 ± 2	4 ± 0	0.118 ± 0.011	0.098 ± 0.024	0.143 ± 0.005	0.292 ± 0.002
DTS-C α=0.5	645 ± 40	16 ± 1	402 ± 54	234 ± 27	67 ± 12	17 ± 4	0.173 ± 0.057	0.054 ± 0.015	0.180 ± 0.045	0.338 ± 0.040
DTS-C α=0.3	674 ± 44	15 ± 3	403 ± 31	231 ± 23	68 ± 6	19 ± 1	0.236 ± 0.052	0.054 ± 0.012	0.120 ± 0.043	0.287 ± 0.037
DTS-C α=0.1	804 ± 46	11 ± 1	455 ± 35	242 ± 49	59 ± 21	13 ± 5	0.155 ± 0.032	0.030 ± 0.025	0.214 ± 0.028	0.375 ± 0.024
DTS-U α=0.1	1,433	12	198	60	9	3	0.061	0.180	0.249	0.369
DTS-U α=0.3	1,172	16	187	57	6	0	0.031	0.211	0.194	0.325
DTS-U α=0.5	1,057	23	116	37	2	0	0.024	0.266	0.201	0.330

protocol shows that DTS's improvement is not solely due to post-hoc selection. The other protocols demonstrate the practical benefit of trajectory reuse: at the same inference budget, DTS produces additional candidates from which a fixed-size downstream batch can be selected, while the formula-diverse protocol shows that the improvement persists after controlling for composition concentration.

G.4 Dielectric Constant and Bandgap

More results for the multi-objective dielectric constant vs bandgap experiment are included in Table 9. We compare across different values of progressive widening parameter α. The conditional version of MatterGen (DTS-C) is fine-tuned over 200 epochs on the MP-20 dataset, where each crystal is labeled with dielectric constants and bandgaps predicted by ALIGNN [Choudhary and DeCost, 2021] and using a conditioning dropout of 20%. We use Tchebycheff scalarization throughout this experiment. The conditional model is conditioned on the Dielectric constant and bandgaps of samples from the Pareto front of the training set, determined using Tchebycheff scalarization. Preferences are sampled from a Dirichlet distribution with $\alpha = 1$. We also compare against a variant that uses an unconditional MatterGen model as its base (DTS-U).

G.5 Symmetry-Conditioned Generation

We ran the symmetry-conditioned experiment of Section 4.2 with four target space groups, chosen to span a range of rarity in the MatterGen training distribution. All runs use the space-group-conditional MatterGen variant for both DTS and the best-of-N baseline; the reward is the indicator $r(\mathbf{x}_0) = \mathbb{1}[\mathrm{SG}(\mathbf{x}_0) = g^\star]$, for target group $g^\star$.

Target space groups. We selected four space groups that appear rarely in the training set. Their frequencies span from $\sim 10^{-2}$ to $\sim 10^{-5}$, allowing us to see how proposal-distribution coverage interacts with search.

Results. Figure 7 shows cumulative SUN $\cap$ target-SG counts versus NFE budget for each target. Three of the four targets (SG 40, 146, 204) follow the pattern of Section 4.2: DTS produces substantially more on-target SUN structures than best-of-N, with the gap widening as the budget grows. The exception is SG 138, the rarest target at $\sim 10^{-5}$ % of the training set: neither method produces any SUN structures matching this space group within the 500K-NFE budget. All runs use $\alpha = 0.5$, $C = 1.0$, $\lambda = 100$. The NFE budget is 500K per method per target. Results are from a single seed per (method, target) pair.

H Importance Sampling Details

H.1 Importance Sampling Ablations

In addition to the ablation study in Section 4.4, we ran ablations on importance sampling (IS) correction for the Space Group and Dielectric/Bandgap experiments. We note that in our method, the purpose of the correction is not to improve reward or yield. A conditional model can provide a more efficient proposal, but it also changes the proposal distribution. The likelihood-ratio correction is intended to reduce this proposal-induced bias relative to the reward-tilted unconditional target.

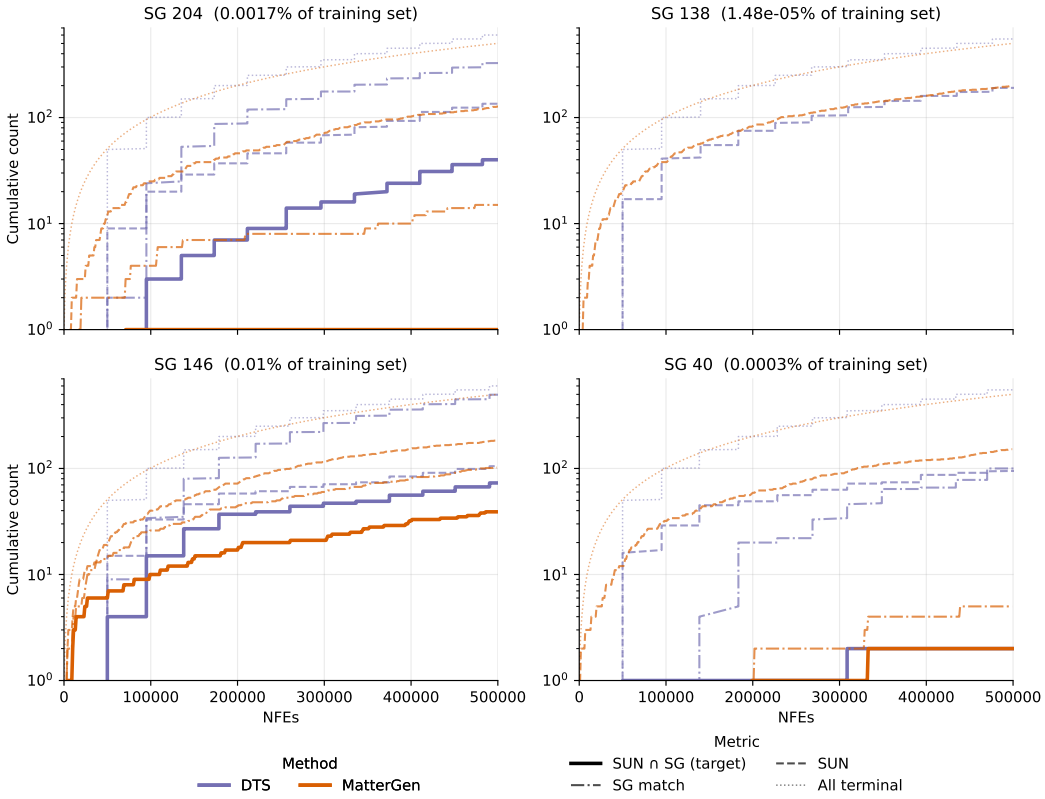


Figure 7: Full view symmetry-conditioned experiment. Color encodes method (DTS vs. MatterGen best-of-N); linestyle encodes metric.

Target space group. We ablate the correction when generating crystals in space group 204 using a space-group-conditioned MatterGen proposal. We evaluate over 3 seeds and 10^6 NFEs.

Table 10: IS ablation for space group 204 targeting (3 seeds, 1M NFEs).

Method	# SG 204 and SUN
DTS with IS	88.0 ± 16.4
DTS without IS	75.0 ± 24.7

Dielectric constant and bandgap. We next ablate IS on the anti-correlated dielectric/bandgap multi-objective task. All runs use $\lambda = 100$ and progressive widening $\alpha = 0.5$. Note that this evaluation uses only 10^6 NFEs, and is not directly comparable to the main results of Section 4.3, which used 2.5×10^6 NFEs.

We compare DTS with importance sampling correction, DTS without IS, and unconditional MatterGen best-of-N.

Table 11: IS ablation on the dielectric/bandgap multi-objective task (1M NFEs). $N_{\geq k}$ denotes the number of structures satisfying threshold k.

Method	SUN Structures	HV ↑	GD+ ↓	R2 ↓	$N_{\geq 30}$	$N_{\geq 60}$	$N_{\geq 120}$	$N_{\geq 240}$
Best-of-N	264 ± 0	0.072 ± 0.035	0.183 ± 0.060	0.306 ± 0.008	103 ± 1	57 ± 3	13 ± 2	1 ± 1
DTS, no IS	325 ± 34	0.084 ± 0.021	0.149 ± 0.022	0.371 ± 0.024	154 ± 16	78 ± 8	20 ± 4	3 ± 1
DTS, IS	286 ± 12	0.068 ± 0.024	0.152 ± 0.037	0.352 ± 0.013	146 ± 7	77 ± 5	20 ± 3	2 ± 0

Removing the IS correction leaves most Pareto metrics within one seed standard deviation of the corrected run, so there is no strong degradation from dropping importance-corrected backup/selection in this setting.

H.2 Importance Sampling Statistics

We report edge log-weight distributions and normalized effective sample size, ESS/n, for the space-group (Table 12), dielectric/bandgap (Table 13), and density/bulk modulus experiments (Table 14). Each tree edge spans 100 diffusion steps between consecutive branching points; its log-weight is the sum of per-step log-ratios over that interval, and the quantiles below are computed over all edges in each interval. At each branching node,

$$\frac{\mathrm{ESS}}{n} = \frac{\left(\sum_{i=1}^{n} \eta_i\right)^2}{n \sum_{i=1}^{n} \eta_i^2} \tag{25}$$

across its n children: a value of 1 indicates equal weights, while a value of approximately $1/n$ indicates concentration on a single child.

H.2.1 Space group.

Table 12: Importance sampling statistics for the space group 204 experiment.

Timesteps	# Branching nodes	Mean # children	Median ESS/n	Log weight 5%	Median log weight	Log weight 95%
1000–901	–	100.00	0.348	-0.03	-0.0001	0.03
900–801	169	5.29	0.763	-0.05	-0.0000	0.05
800–701	304	5.30	0.688	-0.08	-0.0001	0.07
700–601	477	3.36	0.501	-0.40	-0.0011	0.36
600–501	188	2.88	0.500	-15.20	-1.40	1.50
500–401	21	2.76	0.500	-19.79	-2.38	0.71
400–301	1	4.00	0.458	-1.65	-0.07	0.58

For most of the trajectory ($t = 1000$–701), conditional and unconditional transitions remain closely aligned: the median log-weight is approximately 0, the central 90% lies roughly within $[-0.08, 0.07]$, and the median ESS/n is 0.69–0.76 at levels with $\sim$5 children. Concentration peaks between $t = 600$ and 401, where the log-weight tail widens and the median ESS/n drops to 0.50. This effect is localized rather than trajectory-wide; the $t = 500$–401 and 400–301 bins contain only 21 and 1 branching nodes, respectively, so those estimates are noisier.

H.2.2 Dielectric constant and bandgap.

Table 13: Importance sampling statistics for the dielectric/bandgap experiment.

Timesteps	# Branching nodes	Mean # children	Median ESS/n	Log weight 5%	Median log weight	Log weight 95%
1000–901	3	73.00	0.508	-0.002	-0.000	0.002
900–801	210	6.29	0.709	-0.002	-0.000	0.002
800–701	481	3.59	0.795	-0.003	-0.000	0.003
700–601	284	3.33	0.791	-0.090	-0.000	0.087
600–501	130	3.81	0.519	-0.681	-0.021	0.482
500–401	76	3.78	0.547	-0.804	-0.028	0.495
400–301	46	3.86	0.657	-0.457	-0.007	0.319
300–201	31	3.33	0.623	-0.370	-0.005	0.273
200–101	15	2.98	0.717	-0.416	-0.006	0.299
100–001	5	2.00	0.746	-0.508	-0.007	0.348

For dielectric/bandgap, the median ESS/n is 0.71–0.80 between $t = 900$ and 601, then dips to 0.52–0.55 between $t = 600$ and 401, before recovering to 0.62–0.75.

Weights remain usable because DTS corrects locally between consecutive branching nodes rather than accumulating a single importance ratio over the full 1,000-step trajectory. Each decision uses only the ratio from the preceding 100-step segment.

Overall, the weights are neither uniformly benign nor fully degenerate: most branching levels retain moderate ESS, and both tasks show a localized mid-trajectory interval with broader log-weight tails and fewer effective children.

H.2.3 Density/bulk modulus.

Table 14: Importance sampling statistics for the density/bulk modulus experiment.

Timesteps	# Branching nodes	Median ESS/n	Log weight 5%	Median log weight	Log weight 95%
1000–900	48	0.740	-0.0145	-0.0003	0.0133
900–800	122	0.816	-0.0145	-0.0001	0.0138
800–700	244	0.826	-0.0174	-0.0001	0.0167
700–600	314	0.658	-0.0777	-0.0002	0.0742
600–500	161	0.674	-0.4631	-0.0093	0.3683
500–400	21	0.803	-0.4804	-0.0100	0.3521
400–300	2	0.729	-0.1806	-0.0016	0.1508
300–200	1	0.995	-0.1350	-0.0013	0.1199
200–100	0	–	-0.1571	-0.0013	0.1384
100–0	0	–	-0.2253	-0.0032	0.1930