# ESS-Flow: Training-free guidance as Bayesian inference in source space

**Adhithyan Kalaivanan**[1] **Zheng Zhao**[1] **Jens Sjölund**[2] **Fredrik Lindsten**[1]
[1]Linköping University, Sweden [2]Uppsala University, Sweden
{adhithyan.kalaivanan,zheng.zhao,fredrik.lindsten}@liu.se
jens.sjolund@it.uu.se

## Abstract

Guiding pretrained flow-based generative models for conditional generation or to produce samples with desired target properties enables solving diverse tasks without retraining on paired data. We present ESS-Flow, a gradient-free method that leverages the Gaussian prior of the source distribution in flow-based models to perform Bayesian inference directly in the source space using elliptical slice sampling. ESS-Flow only requires forward passes through the generative model and observation process, making it applicable even when gradients are unavailable, such as with non-differentiable simulation-based observations or quantization in the generation process. We demonstrate its effectiveness on scientific applications including generating materials with target properties, protein backbone structures with improved designability, and synthesizable, drug-like molecules. The associated code is available at https://github.com/Adhithyan8/essflow.

## 1 Introduction

In generative modeling, we are given data samples and aim to construct a sampler that approximates the data distribution. Diffusion models [22, 47] and continuous normalizing flows [1, 32, 33] achieve this by transporting samples from a simple source distribution to the data distribution. Beyond unconditional sampling, we often need to generate samples with specific properties or to match observed data in the context of inverse problems. If sufficient paired data are available, we can train specialized conditional generative models [17, 36, 44] for each task.

Training-free conditional generation methods offer a flexible alternative by reusing pretrained models. Instead of paired training data, these methods typically use a likelihood (or more generally, a potential) function to measure how well a sample matches the desired target. Guidance-based methods modify the generative process using the score of the likelihood [47], though this often requires approximations. Optimization-based methods minimize the negative log-likelihood with gradient descent, using the generative model as an explicit regularizer through noising-denoising procedures [30, 35] or an implicit regularizer that constrains the gradient flow to the data manifold [5]. Alternatively, sampling-based approaches formulate controlled generation as Bayesian inference with the pretrained model as the prior. However, existing methods require likelihood gradients [12, 24, 25, 55, 59] or are restricted to linear Gaussian observations [8, 27].

These methods all rely on differentiating through the observation process, and fail when such gradients are unavailable. This limits their applicability in scientific domains where the observation or reward function is often computed by non-differentiable external simulators or involves discrete operations that are not amenable to automatic differentiation [e.g. 2, 3]. Gradient-free alternatives for sampling in the data space could be considered, but these require designing proposal distributions over a complex, high-dimensional data manifold and careful tuning of step sizes.

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

Our key insight is that many pretrained diffusion and flow-based models can be viewed as continuous normalizing flows with Gaussian source distributions, enabling Bayesian inference in the source space instead of the complex data space. The Gaussian prior allows us to use elliptical slice sampling [ESS, 37], a Markov chain Monte Carlo (MCMC) method that generates proposals along ellipses defined by the current state and a draw from the prior. ESS requires no hyperparameter tuning and provides adaptive step sizes through a shrinkage mechanism. This results in ESS-Flow, a gradient-free controlled generation method that approximates the target distribution in the source space of flow-based models, as illustrated in Figure 1.

ESS-Flow can be used with any generative prior, including those trained with Riemannian flow matching on manifold-valued data, as long as the source distribution is Gaussian and the transport map is deterministic. Furthermore, unlike guidance-based methods, ESS-Flow preserves the pretrained vector field. As discussed by Wang et al. [54], this has the benefit that if the prior vector field is optimized for fast generation, for example when trained with minibatch-OT coupling [42, 49], this property is retained.

There exists a plethora of methods [9, 12, 26, 41, 46] designed for image restoration problems, such as inpainting, denoising, and superresolution. Our method addresses a complementary class of problems with important use-cases in scientific applications. There are at least two key differences between these application domains.

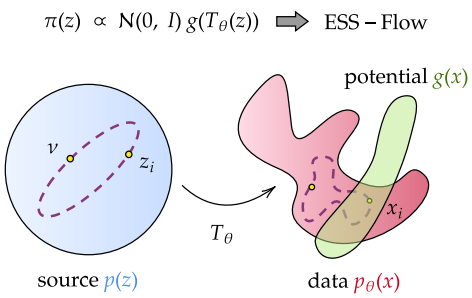



Figure 1: Illustration of ESS-Flow. We sample the target distribution $\pi(z)$ in the source space of flow-based models, which typically has a Gaussian prior $p(z)$. This allows gradient-free sampling with elliptical slice sampling. At stationarity, the transformed ellipse passes through regions of high potential in the data space by construction.

First, many image restoration problems have differentiable and highly informative likelihoods. For instance, in inpainting, the exact equality constraints for observed pixels concentrate the posterior on a low-dimensional manifold in the data space. In such cases, leveraging the available likelihood gradients to steer samples toward this manifold is highly beneficial. In scientific applications, however, we often condition on weakly informative (highly noisy) measurements or target specific properties where the design space remains flexible and non-degenerate. This makes the sampling more amenable to gradient-free methods. Furthermore, non-differentiable likelihoods are more common in scientific applications, ruling out gradient-based methods. We demonstrate these problem types in Section 5, confirming the effectiveness of ESS-Flow in such cases.

Second, for applications in computer vision, perceptual realism in the reconstruction is often the primary objective. This is in stark contrast with many scientific applications where we need sufficient diversity for exploratory analysis and/or reliable uncertainty quantification for hypothesis testing. In such applications, capitalizing on the statistical consistency of well-established Monte Carlo methods, as ESS-Flow does, ensures statistical rigor in the analysis.

We summarize our main contributions as:

- We present ESS-Flow, a training-free and asymptotically exact sampling method for controlled generation that works with non-differentiable observations and discrete operations.
- We demonstrate ESS-Flow on generating materials with target properties, achieving lower mean absolute errors and higher sample quality compared to gradient-based methods. We additionally apply ESS-Flow to protein backbone design and molecule generation with non-differentiable rewards, improving designability and drug-likeness, respectively.

## 2 Preliminaries

**Flow-based generative models.** Consider a generative model that transports samples $x_0$ from a simple source distribution at $t = 0$ to samples $x_1$ from the data distribution at $t = 1$ through a learned vector field $u_t^\theta(x)$ by solving the ordinary differential equation (ODE):

$$x_1 = T_\theta(x_0) \coloneqq x_0 + \int_0^1 u_t^\theta(x_t)\, \mathrm{d}t. \tag{1}$$

Both diffusion models with the probability flow ODE for generation [47] and models trained with the conditional flow matching objective [1, 32] can be viewed as instances of this model class. We collectively refer to them as flow-based generative models with a transport map $T_\theta$ and consider the case where the source distribution is Gaussian, i.e., the model defines a continuous normalizing flow. To simplify notation, we drop the time subscript $t$ and denote source samples $x_0$ by $z$ and data samples $x_1$ by $x$, i.e., the generative model prior is defined by $x = T_\theta(z)$ with $z \sim \mathcal{N}(0, I)$.

**Controlled generation as Bayesian inference.** In controlled generation, we seek samples from a target distribution $\pi(x) \propto g(x)p_\theta(x)$, where the normalization constant is assumed finite so that $\pi(x)$ is a probability distribution. Here, $p_\theta(x)$ is the prior (unconditional) generative model and $g(x)$ is a nonnegative potential function guiding samples toward desired properties. This formulation yields the posterior $p(x|y)$ when $g(x)$ is a likelihood $p(y|x)$ conditioned on observation $y$, or a tilted prior when $g(x)$ is a general reward function. In our setting, the prior data distribution is given by the pretrained flow-based model of the form in equation (1).

## 3 Related Work

Many methods have been proposed for controlled generation with diffusion and flow-based models [see surveys in, e.g., 13, 14, 61]. We focus on those that are training-free and applicable to ODE-based generative models of the form in equation (1). This excludes directly training conditional generative models with paired data [17, 44] and RL-based methods to finetune diffusion models [51, 60].

Methods such as DPS [12], OT-ODE [41], and FlowDPS [28] incorporate a guidance term into the transport map, obtaining target samples in a single pass through the modified generative process. This involves the score of the likelihood, which is often approximated, and the sequential process cannot correct for errors that occur early in the generation. Methods like PnP-Flow [35], DAPS [59], DDSMC [27] and FK-Steering [45] address these limitations by alternating between the noise space and the data space with progressively annealed noise. PnP-Flow optimizes in the data space and regularizes through noising-denoising steps between gradient updates. DAPS generalizes this framework for posterior sampling, and DDSMC extends DAPS using sequential Monte Carlo for linear Gaussian potentials. These methods do not require differentiating the transport map and perform approximate posterior updates in the data space at each iteration of the generative procedure. However, they require access to the noising process used during training.

Source space methods like D-Flow [5] perform gradient-based optimization, relying on implicit regularization that constrains gradient flow to the data manifold. Purohit et al. [43] extend D-Flow to posterior sampling using Langevin Monte Carlo. Wang et al. [54] use Hamiltonian Monte Carlo in the source space, which was explored earlier by Graham and Storkey [20] for variational autoencoders. A key limitation of these methods is their reliance on gradients, requiring backpropagation through the ODE solver. Outsourced Diffusion Sampling [52], which operates in the source space, is gradient-free and can be viewed as variational inference, where the sampler is amortized through additional training. In concurrent work, Wang et al. [53] present gradient-free MCMC in the source space by combining preconditioned Crank–Nicolson (pCN) and parallel tempering over annealed rewards. Elliptical slice sampling, used in our method, can be seen as a tuning-free, adaptive step-size variant of pCN.

## 4 Method

Suppose that we have a generative model prior $p_\theta(x)$ that maps samples from a Gaussian source distribution $z \sim \mathcal{N}(0, I)$ to the data space with the transport map $x = T_\theta(z)$. Given a potential function $g(x)$, such as the likelihood $p(y|x)$ for an observation $y$ or a general reward function, we seek samples from the target $\pi(x)$:

$$\pi(x) \propto g(x)\, p_\theta(x) = g(x)\, \frac{p(z)}{|\det(J_{T_\theta}(z))|}, \tag{2}$$

where $z = T_\theta^{-1}(x)$ and $J_{T_\theta}(z)$ denotes the Jacobian of the transport map evaluated at $z$.

Sampling the target distribution $\pi(x)$ directly in the data space is challenging as exact density evaluation requires computing the Jacobian determinant, which is expensive for continuous normalizing flows. Moreover, gradient-free MCMC methods would require designing effective proposals on the

complex, high-dimensional data manifold. Instead, we can work in the source space, where the simple Gaussian prior enables efficient gradient-free sampling.

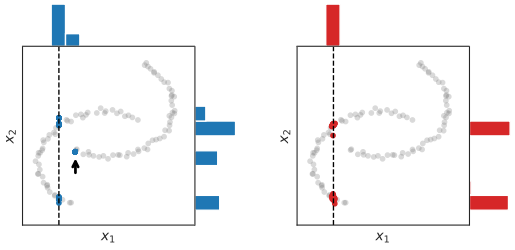
### 4.1 ESS-Flow: gradient-free source space sampling

With a change of variables, we can express the target distribution $\pi(z)$ in the source space:

$$\pi(z) = \pi(x)\,|\det(J_{T_\theta}(z))| \propto g(x)\,\frac{p(z)}{|\det(J_{T_\theta}(z))|}\,|\det(J_{T_\theta}(z))| = g(T_\theta(z))\,p(z). \tag{3}$$

The Jacobian terms cancel when both the prior and the posterior are expressed in the source space, enabling straightforward point-wise density evaluation. We are interested in settings where gradients are unavailable, for instance, when the generative process involves quantization, or when evaluating the potential requires non-differentiable simulators. This rules out gradient-based source space sampling methods such as Langevin Monte Carlo [43] or Hamiltonian Monte Carlo [20, 54], which require computing $\nabla_z g(T_\theta(z)) = J_{T_\theta}(z)^\mathsf{T} \nabla_x g(x)$. Moreover, even when approximate gradients can be computed, gradient-based methods in the source space can struggle when the data distribution has multiple disconnected modes.

To illustrate this, consider D-Flow applied to a toy problem, where the prior $p_\theta(x)$ over $x = (x_1, x_2) \in \mathbb{R}^2$ consists of two interleaving half-circles, and the potential $g(x)$ favors samples whose first coordinate $x_1$ is close to a target value $c$. D-Flow maximizes $g(T_\theta(z))$ w.r.t. $z$ using gradient-based optimization while relying on implicit regularization. A small gradient step $\delta_z \propto J_{T_\theta}(z)^\mathsf{T} \nabla_x g(x)$ in the source space corresponds to the step $\delta_x = J_{T_\theta}(z) J_{T_\theta}(z)^\mathsf{T} \nabla_x g(x)$ in the data space. This projects gradients along the data manifold, resulting in a manifold-constrained gradient flow [5]. As a result, the source space gradient can vanish even when $\nabla_x g(x)$ is non-zero, and samples following the gradient flow can be trapped in local optima. As seen in Figure 2, even with a large learning rate that deviates significantly from a gradient flow, many D-Flow samples remain within the disconnected manifold component where they are initialized. This failure mode is evident in our material generation experiments in Section 5.1.




(a) D-Flow samples (b) ESS-Flow samples

Figure 2: Conditional generation on a 2D toy problem, targeting samples whose first coordinate $x_1$ is close to a value $c$ indicated by the dotted line. Prior samples are shown in gray with marginals along the border. Some D-Flow samples, which follow the source-space gradient, become trapped in disconnected manifold components (indicated by the arrow), while ESS-Flow avoids this problem.

We propose to use elliptical slice sampling [ESS, 37] to sample from the target distribution in the source space. This leverages the fact that the prior $p(z)$ is a multivariate Gaussian, while all the complexity of the transport map $T_\theta(z)$ is absorbed into the potential according to equation (3). This is precisely the setup where ESS excels. Our resulting approach, ESS-Flow, requires only point-wise evaluations of the generative model and involves no gradient or Jacobian computation. Being an MCMC method, ESS-Flow is flexible with initialization, e.g., by sampling $z_0$ from the Gaussian prior, and then proceeding iteratively. As illustrated in Figure 1, a proposal is drawn from an ellipse in the source space defined by the current state $z_i$ and a random sample $\nu$ from the Gaussian prior. Assuming that the transport map $T_\theta$ is continuous in $z$, it nonlinearly maps the ellipse to a continuous, connected curve in the data space that passes through the current state $x_i$. The potential function $g(T_\theta(z))$ evaluated in the data space determines whether the proposal will be accepted. Unlike Metropolis–Hastings, where a rejected proposal requires drawing a new ellipse, ESS adaptively shrinks the angular bracket around the current ellipse when a proposal is rejected. This excludes regions containing the rejected proposal while maintaining support near the current state, from which a new proposal is then drawn. This mechanism provides an adaptive step size that enables larger jumps without tuning. As discussed by Murray et al. [37], this guarantees that the procedure will terminate by accepting a proposal in finite time when the pullback potential function $g \circ T_\theta$ is continuous in a neighborhood around $z_i$. Searching along the ellipse in the source space corresponds to gradient-free exploration of the target distribution in the data space, and the resulting Markov chain in the source space $\{z_i\}$ yields the corresponding Markov chain $\{x_i\}$ when passed through the

transport map. ESS-Flow has no hyperparameters to tune, and one MCMC step is summarized in Algorithm 1.

ESS is known to perform well in high dimensions, with well-defined extensions to infinite-dimensional Hilbert spaces [21]. Murray et al. [37] discuss the convergence of ESS by showing that the induced Markov kernel leaves the target distribution invariant. A more detailed convergence and reversibility analysis is provided by Natarovskii et al. [38] and Hasenpflug et al. [21]. For completeness, we state one of the main results here, adapted to our setting of flow-based models.

**Proposition 4.1** (Geometric convergence of ESS-Flow, Theorem 2.2 by Natarovskii et al. [38])**.** *Suppose that the pullback potential $z \mapsto g \circ T_\theta(z)$ is bounded away from 0 and $\infty$ on compact sets and has regular tail behavior (details in Natarovskii et al. [38], Assumption 2.1), then the ESS Markov chain, which we denote $\nu(\cdot, z)$, converges geometrically fast to the target measure $\pi$:*

$$\|\nu^n(\cdot, z) - \pi\|_{\mathrm{TV}} \leq c\,(1 + \|z\|_2)\,\beta^n, \quad \forall z \in \mathbb{R}^d,$$

*where $\|\cdot\|_{\mathrm{TV}}$ stands for the total variation distance, $\nu^n$ is the $n$-th iteration of the chain, and $c > 0$ and $0 < \beta < 1$ are constants.*

While the result does not characterize how $c$ and $\beta$ depend on dimensionality, Natarovskii et al. [38] empirically find the statistical efficiency of ESS, in terms of the effective sample size, to be dimension-independent. We also present numerical evaluations for the scaling of ESS-Flow with dimension in Appendix A on a GMM prior with a tractable exact posterior.

**Algorithm 1** ESS-Flow: one MCMC iteration

**Require:** Transport map $T_\theta$, potential $g$, current state and potential $(z, g(T_\theta(z)))$

```
ν ~ N(0, I)                                  ▷ Sample the variable which defines the ellipse
u ~ U(0, 1)                                  ▷ Sample the acceptance threshold for this step
θ ~ U(0, 2π), [θ_l, θ_u] ← [θ − 2π, θ]       ▷ Initialize the angle and the bracket
while true do
    z' = z cos θ + ν sin θ                   ▷ Define the proposal along the ellipse
    if log g(T_θ(z')) > log g(T_θ(z)) + log u then
        return (z', g(T_θ(z')))              ▷ If the condition is met, accept the proposal
    else
        if θ < 0 then θ_l ← θ else θ_u ← θ    ▷ Else shrink the bracket to exclude the proposal
        Resample θ ~ U(θ_l, θ_u)             ▷ Sample new angle from the shrunk bracket
```

### 4.2 Multi-fidelity sampling with ESS-Flow

Flow-based generative models are defined in continuous time according to equation (1), but in practice the transport map is solved numerically with finite discretization $\Delta$ rather than exactly. This means that we have access to a class of approximate priors $\{p_\theta^\Delta(x) : \Delta \in (0, 1)\}$ corresponding to different discretization levels, where smaller $\Delta$ gives rise to more accurate models.

Our proposed MCMC-based sampler could be generalized to take advantage of such a multi-fidelity setup. The high-level idea is to rely on coarser, and thus computationally cheaper, evaluations of the transport map for most of the sampling, while ensuring that the final samples nevertheless target a prespecified high-fidelity model. This could be accomplished by delayed acceptance ESS [6], or by methods resembling parallel tempering [18] or simulated tempering [34] where the annealing temperature is replaced by the discretization level. A simpler approach, which we explore here as a proof of concept, is post-correction of the generated samples using importance weighting.

Let $T_\theta^\Delta(z)$ denote the transport map simulated with coarse discretization $\Delta$ and $T_\theta^\delta(z)$ with fine discretization $\delta \ll \Delta$. The target distribution for the high-fidelity model from equation (3) can be rewritten as:

$$\pi^\delta(z) \propto g(T_\theta^\delta(z))\,p(z) = \frac{g(T_\theta^\delta(z))}{g(T_\theta^\Delta(z))} g(T_\theta^\Delta(z))\,p(z) \propto w\,\pi^\Delta(z) \tag{4}$$

We sample from ESS-Flow targeting $\pi^\Delta(z)$ using the coarse transport map $T_\theta^\Delta$ and reweight samples with self-normalized importance weights $w_i \propto g(T_\theta^\delta(z_i))/g(T_\theta^\Delta(z_i))$. The expensive high-fidelity

Table 1: Target properties and values for the material generation experiments.

| Property $P_c$ | Target $y$ | Atoms $n$ | Potential function $g(c)$ | Scale $\sigma_y$ |
|---|---|---|---|---|
| Space group | $P6_3/mmc$ | 8 | $\mathbf{1}[P_c = y]$ | – |
| Bulk modulus | 300 GPa | 4 | $\exp(-(P_c - y)^2/2\sigma_y^2)$ | 10 GPa |
| Shear modulus | 200 GPa | 4 | $\exp(-(P_c - y)^2/2\sigma_y^2)$ | 10 GPa |
| Band gap | 10 eV | 12 | $\exp(-(P_c - y)^2/2\sigma_y^2)$ | 0.1 eV |
| Energy above hull | – | 8 | $\exp(-P_c/\sigma_y)$ | 0.01 eV |

transport map is evaluated only on the final MCMC samples, not during ESS itself, significantly reducing computational cost while maintaining the accuracy of the target distribution. We evaluate this approach in Section 5.1.1 and discuss its effectiveness and limitations.

# 5 Experiments

We evaluate ESS-Flow on generating materials with target properties, designing protein backbone structures, and drug-like molecules. For material generation, we consider differentiable surrogate potentials to compare against state-of-the-art gradient-based methods including D-Flow [5], PnP-Flow [35], and DAPS [59]. For designing protein structures and molecules, the rewards are computed with non-differentiable external programs, which rules out gradient-based methods, and we compare against unconditional sampling from the prior.

## 5.1 Material generation

We use FlowMM [36], a flow-based generative model trained with Riemannian flow matching on the MP-20 dataset [23], as the prior over materials. A crystalline material $c$ with unit lattice consisting of $n$ atoms is represented by the tuple $(\boldsymbol{a}, \boldsymbol{f}, \boldsymbol{l}, \boldsymbol{\beta})$, where $\boldsymbol{a} \in \{-1, 1\}^{n\times 7}$ are the binarized atomic numbers, $\boldsymbol{f} \in [0, 1]^{n\times 3}$ are the fractional coordinates of each atom, $\boldsymbol{l} \in \mathbb{R}^3_+$ are the lattice cell lengths, and $\boldsymbol{\beta} \in [60^\circ, 120^\circ]^3$ are the lattice angles.

While $\boldsymbol{f}, \boldsymbol{l}, \boldsymbol{\beta}$ are continuous variables, the 7-bit binarized atomic numbers $\boldsymbol{a}$ are discrete. FlowMM handles this by embedding atomic types in a continuous space and produces a soft encoding $\tilde{\boldsymbol{a}} \in \mathbb{R}^{n\times 7}$, which is then rounded to obtain the final discrete $\boldsymbol{a}$. This rounding step makes the generative process non-differentiable with respect to the source. We also convert the uniform and log-normal source distributions of $\boldsymbol{f}, \boldsymbol{l}, \boldsymbol{\beta}$ into a standard Gaussian via a change of variables.

To demonstrate ESS-Flow's effectiveness with non-differentiable potentials, we first target materials with a specific space-group symmetry. The potential function for this task is a binary indicator computed using an external program [48], making gradient-based methods inapplicable. To enable comparison with gradient-based baselines, we additionally consider materials with target property values, where the properties are predicted using differentiable ALIGNN [10] models trained on the JARVIS-DFT [11] dataset rather than simulation-based procedures. As target properties, we consider bulk modulus, shear modulus, and band gap, choosing target values above the 99th percentile of the prior property distribution. Additionally, to evaluate the reward-tilted scenario, we guide generation toward stable materials (low predicted energy above hull) without targeting a specific value. Since FlowMM requires specifying the number of atoms per unit lattice, we choose values based on the distribution of atoms for materials with large property values. The target property values, number of atomic sites per lattice, and potential function details are summarized in Table 1.

We apply dimension-wise learning rates for all methods, as gradient magnitudes vary across $(\boldsymbol{a}, \boldsymbol{f}, \boldsymbol{l}, \boldsymbol{\beta})$. Gradient-based methods require approximations to handle the discrete atomic types. ALIGNN expects integer atomic numbers $k_i$ for each atom, which it uses to create atom embeddings $\boldsymbol{h}_i$ by selecting the $k_i$-th row of its embedding matrix $\mathbf{E}$. For D-Flow and PnP-Flow, we maintain differentiability by using the continuous soft encoding $\tilde{\boldsymbol{a}}_i$ to derive continuous values $\tilde{k}_i$, and create soft atom embeddings $\boldsymbol{h}_i$:

$$z_{ij} = (j - \tilde{k}_i)^2, \quad \boldsymbol{v}_i = \mathrm{softmax}(-\boldsymbol{z}_i/\tau), \quad \boldsymbol{h}_i = \mathbf{E}^\mathsf{T}\boldsymbol{v}_i \tag{5}$$

where we set $\tau = 0.1$. For DAPS, we avoid this approximation by applying Langevin Monte Carlo only to the continuous variables $(\boldsymbol{f}, \boldsymbol{l}, \boldsymbol{\beta})$ and sampling $\boldsymbol{a}$ with Metropolis–Hastings using proposals

Table 2: Mean and standard deviation of absolute errors between the sample properties and targets, and of energy above hull values.

| Method | Bulk modulus | Shear modulus | Band gap | Energy above hull |
|---|---|---|---|---|
| D-Flow | $205.88 \pm 60.79$ | $165.93 \pm 27.72$ | $9.24 \pm 1.18$ | $1.92 \pm 1.38$ |
| PnP-Flow | $49.93 \pm 32.33$ | $75.48 \pm 33.45$ | $5.63 \pm 1.03$ | $-0.02 \pm 0.06$ |
| DAPS | $39.14 \pm 26.47$ | $84.33 \pm 37.10$ | $3.90 \pm 1.67$ | $-0.06 \pm 0.05$ |
| ESS-Flow | $\mathbf{8.99 \pm 6.69}$ | $\mathbf{10.53 \pm 9.21}$ | $\mathbf{1.85 \pm 1.66}$ | $\mathbf{-0.19 \pm 0.07}$ |

Table 3: S.U.N.T. rates (%) for the generated samples across different material generation tasks.

| Method | Bulk modulus | Shear modulus | Band gap | Energy above hull | Space group |
|---|---|---|---|---|---|
| Unconditional | 0.0 | 0.0 | 0.0 | 1.4 | 1.3 |
| D-Flow | 0.0 | 0.0 | 0.0 | 1.8 | – |
| PnP-Flow | 3.8 | 0.9 | 0.1 | 34.5 | – |
| DAPS | 9.4 | 0.5 | 0.0 | 17.6 | – |
| ESS-Flow | **13.7** | **4.7** | **16.0** | **37.6** | **25.5** |

that flip one bit of the binarized atomic numbers. The noising and denoising steps are adapted from the original diffusion formulation to match FlowMM. ESS-Flow requires no such approximations and handles discrete atomic types naturally, as it only makes point-wise evaluations of the generative model and potential function. We provide a detailed analysis on the runtime costs of the different methods in Appendix B.2.

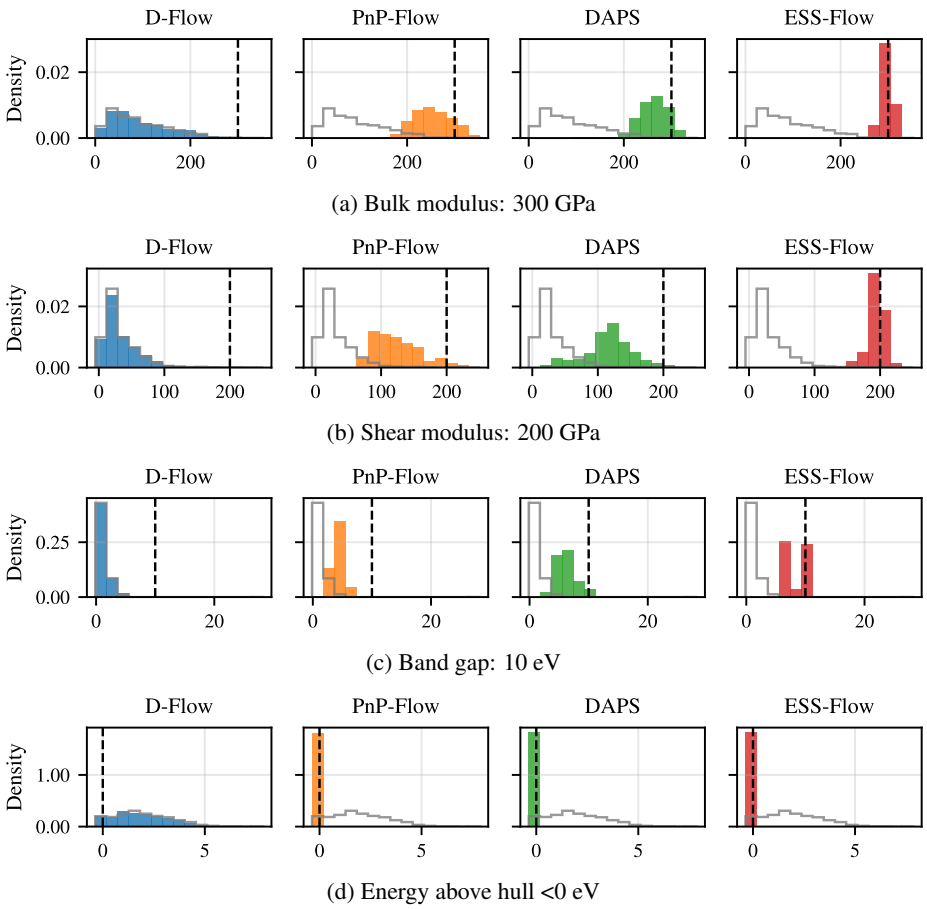
**Results.** ESS-Flow successfully generates 92.3% of samples with the target $P6_3/mmc$ space group, compared to only 2.5% from unconditional sampling. For the property-targeting tasks, we measure performance using absolute errors between sample properties and targets, and for stability, we report the predicted energy above hull, as shown in Table 2. As anticipated from the toy example in Figure 2, D-Flow fails to explore atomic compositions far from initialization even with the continuous approximation for $\boldsymbol{a}$, with performance nearly identical to unconditional sampling from the prior. PnP-Flow, which optimizes in data space with noising-denoising steps, is less constrained by disconnected modes. By avoiding gradient-based exploration of $\boldsymbol{a}$, DAPS and ESS-Flow perform considerably better. ESS-Flow achieves the lowest errors across all tasks, outperforming the baselines significantly. The sample property distributions illustrated in Figure 3 show that ESS-Flow better recovers the sharp target distributions.

Beyond accuracy in matching target properties, we assess the quality of generated samples by computing the S.U.N.T. rate (stability, uniqueness, novelty, and threshold), adapted from prior works developing generative models for materials such as FlowMM and MatterGen [58]. Briefly, the S.U.N.T. rate measures the following four properties: Stability (S.) as the proportion of structures that are thermodynamically stable after relaxation, Uniqueness (U.) as the proportion of generated structures that are distinct from each other, Novelty (N.) as the proportion that do not appear in the training set, and Threshold (T.) as the proportion meeting the desired target constraint. The combined S.U.N.T. rate is the proportion of generated samples that satisfy all four criteria simultaneously. While the S.U.N. rate is an established quality metric for unconditional material generation, we introduce the threshold rate (T.) to adapt it to the controlled generation setting. A detailed description of each metric and the full relaxation pipeline can be found in Appendix B.1. The S.U.N.T. rates computed over 1000 generated samples are reported in Table 3.

ESS-Flow successfully targets the desired properties and achieves the highest S.U.N.T. rates across all tasks. For comparison, the S.U.N. rate for unconditional generation with FlowMM is nearly 26%. Despite the challenging controlled generation problem, where we target extreme values for the desired properties, the resulting S.U.N.T. rates indicate that ESS-Flow generates materials that are viable candidates for further exploration.

### 5.1.1 Evaluation of multi-fidelity sampling

We perform a preliminary evaluation of the multi-fidelity sampling approach proposed in Section 4.2. ESS-Flow samples are obtained by evaluating the potential with a coarse discretization $\Delta = 1/50$,




(a) Bulk modulus: 300 GPa

(b) Shear modulus: 200 GPa

(c) Band gap: 10 eV

(d) Energy above hull <0 eV

Figure 3: Sample property distributions for different methods. Property values are along the x-axis, the prior property distribution is shown in gray, and target values are indicated by dotted lines. ESS-Flow samples concentrate near target values across all tasks.

then reweighted using importance weights computed with a much finer discretization $\delta = 1/1000$, as described in equation (4). The weighted samples have reasonable effective sample sizes of 65.3% and 33.9% for the bulk modulus and shear modulus tasks, respectively. However, for the band gap and stability tasks, which have sharper, concentrated target distributions, the effective sample sizes drop to 0.1% and 1.0%, respectively. This is a limitation of the simple importance reweighting approach, where the weights $w \propto g(T^{\delta}(z))/g(T^{\Delta}(z))$ can become extremely large when the coarse discretization yields samples with lower potential values. Further analysis comparing ESS-Flow samples and the importance weighted samples is provided in Appendix B.3.

### 5.2 Protein backbone design

We consider the 60M-parameter variant of Proteina [19], a flow-based generative model, as the prior over protein backbones with 100 residues. We target structures with higher designability, which is measured as follows [19, 57]. For a generated backbone, $N$ candidate amino acid sequences are predicted with the inverse folding model ProteinMPNN [15] at temperature 0.1, and each sequence is refolded with ESMFold [31]. The self-consistency RMSD (scRMSD) is defined as the minimum RMSD between the refolded structures and the generated backbone after alignment. The backbone is considered designable if its scRMSD is below 2Å, and designability is the percentage of designable backbones among the generated samples.

We sample the tilted prior with the non-differentiable potential $g(x) = \exp(-w \cdot \text{scRMSD}(x))$, computed with $N = 1$, for different reward strengths $w > 0$. All samples are generated with the

Table 4: Designability and diversity of generated protein backbone structures.

| Method | scRMSD (Å) ↓ | Designability (%) ↑ | TM-score ↓ |
|---|---|---|---|
| Unconditional | 6.74 | 9.6 | 0.361 |
| ESS-Flow ($w = 0.5$) | 2.49 | 53.1 | 0.368 |
| ESS-Flow ($w = 1.0$) | 1.92 | 74.0 | 0.367 |
| ESS-Flow ($w = 2.0$) | 1.53 | 86.4 | 0.380 |

Table 5: Metrics for the generated molecules. Conn., Uniq., and Qual. are in percentage, and QED and SA scores are reported with mean and standard deviation.

| Method | Conn. ↑ | Div. ↑ | Uniq. ↑ | QED ↑ | SA ↓ | Qual. ↑ |
|---|---|---|---|---|---|---|
| Unconditional (DDPM) | 70.0 | 0.843 | 100.0 | $0.413 \pm 0.183$ | $6.954 \pm 0.934$ | 0.0 |
| Unconditional (DDIM) | 44.3 | 0.848 | 100.0 | $0.351 \pm 0.177$ | $6.922 \pm 0.945$ | 0.1 |
| ESS-Flow ($w = 5$) | 99.8 | 0.849 | 91.6 | $0.435 \pm 0.189$ | $6.689 \pm 1.116$ | 0.3 |
| ESS-Flow ($w = 10$) | 99.1 | 0.848 | 80.4 | $0.520 \pm 0.190$ | $6.377 \pm 1.336$ | 0.4 |
| ESS-Flow ($w = 20$) | 97.9 | 0.826 | 39.7 | $0.635 \pm 0.159$ | $4.760 \pm 2.252$ | 0.9 |

probability flow ODE using 50 steps, and ESS-Flow is run with 16 parallel MCMC chains for 264 steps, discarding the first 200 steps as burn-in. Over the resulting 1024 samples, we report the mean scRMSD and designability, evaluated with $N = 8$. We also measure diversity as the average pairwise TM-score [19] among the designable samples, where lower values indicate higher diversity.

**Results.** As shown in Table 4, ESS-Flow lowers the mean scRMSD and substantially improves designability over unconditional sampling, with larger gains at higher reward strengths. Sample diversity is largely preserved, with a slight increase in the pairwise TM-score at the highest reward strength as the tilted prior becomes more concentrated.

### 5.3 Molecule generation

We use GeoLDM [56], a latent diffusion model trained on the GEOM-Drugs dataset [4], as the prior over 28-atom 3D molecules. With the DDIM sampler using 50 steps, the composition of the latent transport and the decoder defines a deterministic map from the Gaussian source to molecules, making GeoLDM suitable for ESS-Flow. We target synthesizable, drug-like molecules using the quantitative estimate of drug-likeness, QED $\in [0, 1]$, where higher values indicate higher drug-likeness, and the synthetic accessibility score, SA $\in [1, 10]$, where lower values indicate ease of synthesis, both computed with the RDKit library. Unconditional samples from GeoLDM often consist of disconnected fragments, and those that remain connected have low QED and high SA scores. For a reward strength $w > 0$, we sample the tilted prior with the non-differentiable potential $g(x) = \exp(-w)$ if the molecule $x$ is disconnected, and $g(x) = \exp(w \cdot (\text{QED}(x) + \text{SA}_{\text{norm}}(x))/2)$ otherwise, where $\text{SA}_{\text{norm}} = (10 - \text{SA})/9 \in [0, 1]$.

Following Lee et al. [29], we report the percentage of connected molecules (Conn.), and filter out disconnected ones before computing diversity (Div.), the percentage of unique molecules (Uniq.), and the QED and SA scores over 1024 generated samples. Uniqueness is the percentage of distinct canonical SMILES, and diversity is the average pairwise Tanimoto distance between the Morgan fingerprints of all connected molecules. We report quality (Qual.) as the percentage of samples that are connected and unique, with QED $\geq 0.6$ and SA $\leq 4.0$. We compare against unconditional sampling with both the original DDPM sampler with 1000 steps and the deterministic DDIM sampler. ESS-Flow is run with the same configuration as in the protein backbone design experiment.

**Results.** ESS-Flow almost exclusively produces connected molecules as shown in Table 5, compared to unconditional generation with both DDPM and DDIM samplers. The QED and SA scores steadily improve with increasing reward strength, and at $w = 20$, the fraction of unique molecules drops as the tilted prior becomes more concentrated. However, the overall quality, which accounts for uniqueness and the QED and SA thresholds, continues to increase with reward strength, indicating that ESS-Flow reliably steers generation towards the desired properties.

## 6 Limitations and broader impacts

As a method for sampling the tilted prior, the performance of ESS-Flow is limited by how well the generative model approximates the true data distribution under moderate ODE discretization. Recent developments in flow maps [7] that maintain sample quality with few step evaluations will further enable the method. Additionally, ESS-Flow is more expensive in terms of the number of function evaluations than the compared baselines, which should be expected of a zeroth-order sampling method designed to solve a broader class of problems where gradient-based methods fail. However, we have not optimized the method for computational efficiency in this work. ESS admits tunable variants that reduce the number of shrinkage steps per iteration, by restricting the initial angular bracket to a length smaller than $2\pi$, or by biasing proposals toward the current state instead of sampling uniformly along the ellipse. Such problem-specific tuning moves the method closer to pCN, at the expense of the tuning-free nature of ESS-Flow. Lastly, warm starting the MCMC chains and leveraging parallel chains to speed up mixing [39] would further reduce the computational cost.

Controlled generation methods can accelerate the discovery of therapeutic drugs and novel materials by improving the screening process. However, they carry dual-use risks, as they could be repurposed to design harmful substances [50]. While generative models have wide-ranging societal impacts, ESS-Flow does not introduce new risks beyond those of the pretrained models it builds upon.

## 7 Conclusion

We present ESS-Flow, a training-free and gradient-free sampling method for controlled generation with pretrained flow-based generative models. The method is particularly valuable for problems where gradients are unavailable, such as when the generative process involves quantization or when evaluating the potential requires non-differentiable simulations, settings that are common in scientific applications. We propose and evaluate a multi-fidelity extension using importance reweighting as a proof of concept. Future work could explore more sophisticated multi-fidelity approaches, such as tempering-based methods, to better balance computational efficiency with sampling accuracy.

## Acknowledgments and Disclosure of Funding

This work was financially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation, the Swedish Research Council (project no: 2024-05011), and the Excellence Center at Linköping–Lund in Information Technology (ELLIIT). Computational resources were provided on the Berzelius system funded by the Knut and Alice Wallenberg foundation and operated by NAISS.

## References

[1] M. S. Albergo, N. M. Boffi, and E. Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

[2] R. F. Alford, A. Leaver-Fay, J. R. Jeliazkov, M. J. O'Meara, F. P. DiMaio, H. Park, M. V. Shapovalov, P. D. Renfrew, V. K. Mulligan, K. Kappel, et al. The Rosetta all-atom energy function for macromolecular modeling and design. *Journal of chemical theory and computation*, 13(6):3031–3048, 2017.

[3] A. Alhossary, S. D. Handoko, Y. Mu, and C.-K. Kwoh. Fast, accurate, and reliable molecular docking with QuickVina 2. *Bioinformatics*, 31(13):2214–2216, 2015.

[4] S. Axelrod and R. Gomez-Bombarelli. GEOM, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific data*, 9(1):185, 2022.

[5] H. Ben-Hamu, O. Puny, I. Gat, B. Karrer, U. Singer, and Y. Lipman. D-Flow: Differentiating through flows for controlled generation. In *International Conference on Machine Learning*, pages 3462–3483. PMLR, 2024.

[6] K. Bitterlich, D. Rudolf, and B. Sprungk. Delayed acceptance elliptical slice sampling. *PAMM*, 25(1):e70005, 2025.

[7] N. M. Boffi, M. S. Albergo, and E. Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation. In *Advances in Neural Information Processing Systems*, 2025.

[8] G. V. Cardoso, Y. J. El Idrissi, S. Le Corff, and E. Moulines. Monte Carlo guided diffusion for Bayesian linear inverse problems. In *ICLR International Conference on Learning Representations*, 2024.

[9] J. Choi, S. Kim, Y. Jeong, Y. Gwon, and S. Yoon. ILVR: Conditioning method for denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14367–14376, 2021.

[10] K. Choudhary and B. DeCost. Atomistic line graph neural network for improved materials property predictions. *npj Computational Materials*, 7(1):185, 2021.

[11] K. Choudhary, K. F. Garrity, A. C. Reid, B. DeCost, A. J. Biacchi, A. R. Hight Walker, Z. Trautt, J. Hattrick-Simpers, A. G. Kusne, A. Centrone, et al. The joint automated repository for various integrated simulations (JARVIS) for data-driven materials design. *npj computational materials*, 6(1):173, 2020.

[12] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023.

[13] H. Chung, J. Kim, and J. C. Ye. Diffusion models for inverse problems. *arXiv preprint arXiv:2508.01975*, 2025.

[14] G. Daras, H. Chung, C.-H. Lai, Y. Mitsufuji, J. C. Ye, P. Milanfar, A. G. Dimakis, and M. Delbracio. A survey on diffusion models for inverse problems. *arXiv preprint arXiv:2410.00083*, 2024.

[15] J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. Wicky, A. Courbet, R. J. d. Haas, N. Bethel, et al. Robust deep learning–based protein sequence design using ProteinMPNN. *Science*, 378(6615):49–56, 2022.

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

[17] P. Dhariwal and A. Nichol. Diffusion models beat GANs on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[18] D. J. Earl and M. W. Deem. Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7(23):3910–3916, 2005.

[19] T. Geffner, K. Didi, Z. Zhang, D. Reidenbach, Z. Cao, J. Yim, M. Geiger, C. Dallago, E. Kucukbenli, A. Vahdat, and K. Kreis. Proteina: Scaling flow-based protein structure generative models. In *International Conference on Learning Representations*, 2025.

[20] M. Graham and A. Storkey. Asymptotically exact inference in differentiable generative models. In *Artificial Intelligence and Statistics*, pages 499–508. PMLR, 2017.

[21] M. Hasenpflug, V. Telezhnikov, and D. Rudolf. Reversibility of elliptical slice sampling revisited. *Bernoulli*, 31(2):1377–1401, 2025.

[22] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[23] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, et al. Commentary: The Materials Project: A materials genome approach to accelerating materials innovation. *APL materials*, 1(1), 2013.

[24] Y. Janati, B. Moufad, A. Durmus, E. Moulines, and J. Olsson. Divide-and-conquer posterior sampling for denoising diffusion priors. *Advances in Neural Information Processing Systems*, 37:97408–97444, 2024.

[25] Y. Janati, B. Moufad, M. Abou El Qassime, A. O. Durmus, E. Moulines, and J. Olsson. A mixture-based framework for guiding diffusion models. In *Forty-second International Conference on Machine Learning*, 2025.

[26] B. Kawar, M. Elad, S. Ermon, and J. Song. Denoising diffusion restoration models. *Advances in neural information processing systems*, 35:23593–23606, 2022.

[27] F. E. Kelvinius, Z. Zhao, and F. Lindsten. Solving linear-Gaussian Bayesian inverse problems with decoupled diffusion sequential Monte Carlo. In *International Conference on Machine Learning*, 2025.

[28] J. Kim, B. S. Kim, and J. C. Ye. FlowDPS: Flow-driven posterior sampling for inverse problems. *arXiv preprint arXiv:2503.08136*, 2025.

[29] S. Lee, K. Kreis, S. P. Veccham, M. Liu, D. Reidenbach, Y. Peng, S. G. Paliwal, W. Nie, and A. Vahdat. GenMol: A drug discovery generalist with discrete diffusion. In *International Conference on Machine Learning*, 2025.

[30] A. Levy, E. R. Chan, S. Fridovich-Keil, F. Poitevin, E. D. Zhong, and G. Wetzstein. Solving inverse problems in protein space using diffusion-based priors. *arXiv preprint arXiv:2406.04239*, 2024.

[31] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[32] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

[33] X. Liu, C. Gong, et al. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

[34] E. Marinari and G. Parisi. Simulated tempering: a new Monte Carlo scheme. *Europhysics letters*, 19(6):451, 1992.

[35] S. T. Martin, A. Gagneux, P. Hagemann, and G. Steidl. PnP-Flow: Plug-and-play image restoration with flow matching. In *International Conference on Learning Representations*, 2025.

[36] B. K. Miller, R. T. Chen, A. Sriram, and B. M. Wood. FlowMM: Generating materials with Riemannian flow matching. In *International Conference on Machine Learning*, pages 35664–35686. PMLR, 2024.

[37] I. Murray, R. Adams, and D. MacKay. Elliptical slice sampling. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 541–548. JMLR Workshop and Conference Proceedings, 2010.

[38] V. Natarovskii, D. Rudolf, and B. Sprungk. Geometric convergence of elliptical slice sampling. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 7969–7978. PMLR, 2021.

[39] R. Nishihara, I. Murray, and R. P. Adams. Parallel MCMC with generalized elliptical slice sampling. *The Journal of Machine Learning Research*, 15(1):2087–2112, 2014.

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

[41] A. Pokle, M. J. Muckley, R. T. Chen, and B. Karrer. Training-free linear image inverses via flows. *Transactions on Machine Learning Research*, 2024.

[42] A.-A. Pooladian, H. Ben-Hamu, C. Domingo-Enrich, B. Amos, Y. Lipman, and R. T. Chen. Multisample flow matching: Straightening flows with minibatch couplings. In *International Conference on Machine Learning*, pages 28100–28127. PMLR, 2023.

[43] V. Purohit, M. Repasky, J. Lu, Q. Qiu, Y. Xie, and X. Cheng. Consistency posterior sampling for diverse image synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28327–28336, 2025.

[44] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[45] R. Singhal, Z. Horvitz, R. Teehan, M. Ren, Z. Yu, K. McKeown, and R. Ranganath. A general framework for inference-time scaling and steering of diffusion models. In *International Conference on Machine Learning*, 2025.

[46] J. Song, A. Vahdat, M. Mardani, and J. Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023.

[47] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[48] A. Togo, K. Shinohara, and I. Tanaka. Spglib: a software library for crystal symmetry search. *Science and Technology of Advanced Materials: Methods*, 4(1):2384822, 2024.

[49] A. Tong, K. Fatras, N. Malkin, G. Huguet, Y. Zhang, J. Rector-Brooks, G. Wolf, and Y. Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, pages 1–34, 2024.

[50] F. Urbina, F. Lentzos, C. Invernizzi, and S. Ekins. Dual use of artificial-intelligence-powered drug discovery. *Nature machine intelligence*, 4(3):189–191, 2022.

[51] S. Venkatraman, M. Jain, L. Scimeca, M. Kim, M. Sendera, M. Hasan, L. Rowe, S. Mittal, P. Lemos, E. Bengio, et al. Amortizing intractable inference in diffusion models for vision, language, and control. *Advances in neural information processing systems*, 37:76080–76114, 2024.

[52] S. Venkatraman, M. Hasan, M. Kim, L. Scimeca, M. Sendera, Y. Bengio, G. Berseth, and N. Malkin. Outsourced diffusion sampling: Efficient posterior inference in latent spaces of generative models. In *International Conference on Machine Learning*, pages 61212–61239. PMLR, 2025.

[53] S.-H. Wang, J. Keller, T. Transue, D. B. Brown, T. Strohmer, and B. Wang. Test-time guidance for flow-based generative models via parallel tempering on source distributions. In *Forty-third International Conference on Machine Learning*, 2026.

[54] Z. Wang, A. Harting, M. Barreau, M. M. Zavlanos, and K. H. Johansson. Source-guided flow matching. *arXiv preprint arXiv:2508.14807*, 2025.

[55] L. Wu, B. Trippe, C. Naesseth, D. Blei, and J. P. Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. *Advances in Neural Information Processing Systems*, 36:31372–31403, 2023.

[56] M. Xu, A. S. Powers, R. O. Dror, S. Ermon, and J. Leskovec. Geometric latent diffusion models for 3D molecule generation. In *International Conference on Machine Learning*, pages 38592–38610, 2023.

[57] J. Yim, B. L. Trippe, V. De Bortoli, E. Mathieu, A. Doucet, R. Barzilay, and T. Jaakkola. SE(3) diffusion model with application to protein backbone generation. In *International Conference on Machine Learning*, 2023.

[58] C. Zeni, R. Pinsler, D. Zügner, A. Fowler, M. Horton, X. Fu, Z. Wang, A. Shysheya, J. Crabbé, S. Ueda, et al. A generative model for inorganic materials design. *Nature*, 639(8055):624–632, 2025.

[59] B. Zhang, W. Chu, J. Berner, C. Meng, A. Anandkumar, and Y. Song. Improving diffusion inverse problem solving with decoupled noise annealing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20895–20905, 2025.

[60] Y. Zhao, M. Uehara, G. Scalia, S. Kung, T. Biancalani, S. Levine, and E. Hajiramezanali. Adding conditional control to diffusion models with reinforcement learning. In *International Conference on Learning Representations*, 2025.

[61] Z. Zhao, Z. Luo, J. Sjölund, and T. B. Schön. Conditional sampling within generative diffusion models. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 383(2299):20240329, 2025.

## 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: We claim our main contribution as introducing a training-free method for controlled generation in flow-based models that does not require differentiability of the generative or observation processes. We present the method in detail in Section 4 and provide experimental validation in Section 5.

   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: We discuss the limitations of our work in Section 6.

   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: [N/A]

   Justification: While we restate the result of Natarovskii et al. [38] adapted to our setting for completeness, we do not claim any new theoretical contribution. The consistency and convergence properties of our method rely on existing theoretical results for elliptical slice sampling.

   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 experiments and evaluation metrics are described in Section 5, with implementation details for all methods provided in the Appendix.

   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: [No]

   Justification: The code is not provided at the time of submission, but will be made publicly available upon publication.

   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: A general description of the experiments is provided in Section 5, with additional hyperparameter details included in the Appendix.

   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: We report the mean and standard deviation of the metrics across generated samples in the results.

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: We compare and discuss the computational costs and peak GPU memory usage for all the methods in the Appendix.

   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: We have reviewed the guidelines and confirm that our work adheres to the NeurIPS Code of Ethics.

   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: We discuss the broader impacts of the work in Section 6.

    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: We do not release any new model or dataset in this work.

    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 existing datasets, models, and code used in this work are appropriately cited, and are used in accordance with their respective licenses and terms 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: No new datasets or models are released with this work. The implementation code will be made public upon publication.

    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: This work does not involve crowdsourcing experiments or research with 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 crowdsourcing or research with 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: The core method development does not involve LLMs.

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.

Table 6: Sliced Wasserstein distance (mean $\pm$ standard error) and average number of likelihood evaluations per MCMC step (mean $\pm$ standard deviation) across 10 random seeds.

| | SWD | | | $n_{\text{ellipse}}$ | | |
|---|---|---|---|---|---|---|
| $d_y$ | $d_x = 8$ | $d_x = 80$ | $d_x = 800$ | $d_x = 8$ | $d_x = 80$ | $d_x = 800$ |
| 1 | $0.037 \pm 0.010$ | $0.044 \pm 0.019$ | $0.064 \pm 0.025$ | $2.8 \pm 1.3$ | $2.8 \pm 1.1$ | $2.7 \pm 0.5$ |
| 2 | $0.034 \pm 0.014$ | $0.034 \pm 0.009$ | $0.037 \pm 0.016$ | $3.2 \pm 0.7$ | $2.8 \pm 1.2$ | $3.2 \pm 1.6$ |
| 4 | $0.028 \pm 0.008$ | $0.031 \pm 0.014$ | $0.038 \pm 0.024$ | $3.5 \pm 1.2$ | $3.0 \pm 1.4$ | $4.1 \pm 2.4$ |

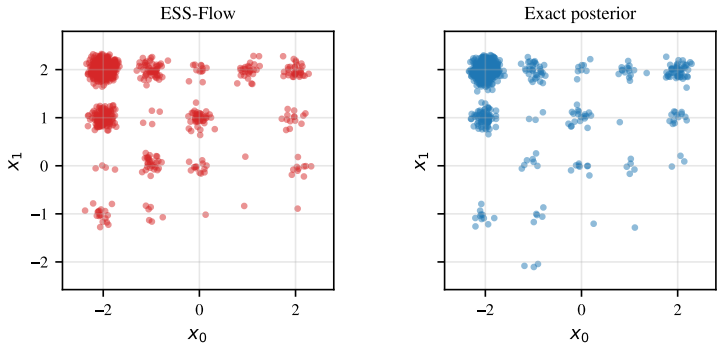



Figure 4: First two dimensions of the posterior approximated with ESS-Flow compared to the exact posterior for the GMM example with $d_x = 800$ and $d_y = 1$.

## A Gaussian mixture model

We numerically evaluate how ESS-Flow scales with dimension on simulated data. The prior distribution is a Gaussian mixture model with 25 components, with means $\mu_{i,j} = (i, j, \ldots, i, j) \in \mathbb{R}^{d_x}$ for $(i, j) \in \{-2, -1, 0, 1, 2\}^2$ and covariances $(1/8)^2 I_{d_x}$. The unnormalized weights of the mixture components are independently drawn from a $\chi^2$ distribution. We consider a linear-Gaussian observation model $y = Ax + \sigma_y \epsilon$ with $\epsilon \sim \mathcal{N}(0, I_{d_y})$, so the posterior is also a Gaussian mixture that can be computed exactly. The setup is similar to that of Cardoso et al. [8], except we use rescaled means and covariances while maintaining the relative separation between mixture components.

To avoid model approximation errors from learning a transport map, we use the analytical expression derived in Albergo et al. [1] for the linear interpolant $x_t = (1 - t)x_0 + tx_1$ from a standard normal source to the Gaussian mixture. For ESS-Flow, we run 1000 parallel MCMC chains for 501 steps each and collect the last sample from each chain. The ODE is numerically integrated using the midpoint method with a step size of 0.05. We report the sliced Wasserstein distance (SWD) between ESS-Flow samples and exact posterior samples in Table 6. The average number of likelihood evaluations per MCMC step, denoted by $n_{\text{ellipse}}$, measures sampling efficiency within ESS.

As dimension increases, ESS-Flow's sampling efficiency remains stable and the SWD remains low, indicating good approximation. For qualitative evaluation, we visualize the first two dimensions of the posterior for one random seed with $d_x = 800$ and $d_y = 1$ in Figure 4.

## B Additional details for material generation

The hyperparameters for all methods used in the material generation experiments are provided in Table 7.

### B.1 Quality metrics

We describe how the S.U.N.T. rate is computed for a set of $M$ generated materials $\{c_1, \ldots, c_M\}$.

Table 7: Hyperparameters for the material generation tasks. Gradient-based methods use dimension-specific learning rates for $(\boldsymbol{a}, \boldsymbol{f}, \boldsymbol{l}, \boldsymbol{\beta})$.

| Method / Parameter | Bulk modulus | Shear modulus | Band gap | Energy above hull |
|---|---|---|---|---|
| **D-Flow** | | | | |
| ODE integration steps $N$ | 50 | 50 | 50 | 50 |
| Optimizer | SGD | SGD | SGD | SGD |
| Optimization steps $N_{opt}$ | 100 | 100 | 100 | 100 |
| Learning rate $(\boldsymbol{a}, \boldsymbol{l}, \boldsymbol{\beta})$ | $10^{-6}$ | $10^{-5}$ | $10^{-4}$ | $10^{-4}$ |
| Learning rate $(\boldsymbol{f})$ | $10^{-8}$ | $10^{-7}$ | $10^{-6}$ | $10^{-6}$ |
| **PnP-Flow** | | | | |
| Optimizer | SGD | SGD | SGD | SGD |
| Optimization steps $N_{opt}$ | 100 | 100 | 100 | 100 |
| Learning rate decay | linear | linear | linear | linear |
| Min. learning rate | 0 | 0 | 0 | 0 |
| Max. learning rate $(\boldsymbol{a})$ | $10^{-4}$ | $10^{-3}$ | $10^{-1}$ | $10^{-2}$ |
| Max. learning rate $(\boldsymbol{f})$ | $10^{-6}$ | $10^{-5}$ | $10^{-3}$ | $10^{-4}$ |
| Max. learning rate $(\boldsymbol{l}, \boldsymbol{\beta})$ | $10^{-5}$ | $10^{-4}$ | $10^{-2}$ | $10^{-3}$ |
| ODE integration steps $N$ | $100 - n$ | $100 - n$ | $100 - n$ | $100 - n$ |
| **DAPS** | | | | |
| ODE integration steps $N$ | 5 | 5 | 5 | 5 |
| Optimization steps $N_{opt}$ | 100 | 100 | 100 | 100 |
| Sampling steps $N_s$ | 20 | 20 | 20 | 20 |
| Learning rate decay | linear | linear | linear | linear |
| Min. learning rate | $10^{-2}\alpha$ | $10^{-2}\alpha$ | $10^{-2}\alpha$ | $10^{-2}\alpha$ |
| Max. learning rate $(\alpha_{\boldsymbol{f}})$ | $10^{-3}$ | $10^{-3}$ | $10^{-5}$ | $10^{-4}$ |
| Max. learning rate $(\alpha_{\boldsymbol{l}}, \alpha_{\boldsymbol{\beta}})$ | $10^{-2}$ | $10^{-2}$ | $10^{-4}$ | $10^{-3}$ |
| $p(x_0\|x_t)$ scale $(r_{\boldsymbol{a}})$ | $3(1-t)$ | $3(1-t)$ | $3(1-t)$ | $3(1-t)$ |
| $p(x_0\|x_t)$ scale $(r_{\boldsymbol{f}}, r_{\boldsymbol{l}}, r_{\boldsymbol{\beta}})$ | $1-t$ | $1-t$ | $1-t$ | $1-t$ |
| **ESS-Flow** | | | | |
| ODE integration steps $N$ | 50 | 50 | 50 | 50 |
| MCMC steps | 201 | 201 | 501 | 201 |
| Burn-in | 200 | 200 | 500 | 200 |

**Validity and relaxation.** A generated material $c_m$ is valid if its lattice volume is greater than 0.1 Å$^3$, its atomic numbers are within the valid range, and its charge is balanced.

$$v_m = \mathbf{1}[\text{vol}(c_m) > 0.1] \cdot \mathbf{1}[a_m \in \mathcal{A}_{\text{valid}}] \cdot \mathbf{1}[\text{charge}(c_m) = 0], \tag{6}$$

where $\mathcal{A}_{\text{valid}}$ denotes the set of valid atomic numbers. We discard invalid materials and relax the remaining structures using CHGNet [16], following FlowMM, without further relaxations using density functional theory. Let $\tilde{c}_m$ denote the relaxed structure.

**Stability.** The stability indicator for a relaxed structure $\tilde{c}_m$ is given by

$$s_m = \mathbf{1}[E_{\text{hull}}(\tilde{c}_m) < 0.1 \text{ eV/atom}], \tag{7}$$

where $E_{\text{hull}}$ is the energy above hull predicted by CHGNet. Note that this differs from the ALIGNN predictor used to guide generation of metastable materials, and can thus produce different results. The stability rate (S.) is $\sum_m v_m s_m / M$.

**Uniqueness and novelty.** Two relaxed structures $\tilde{c}_m$ and $\tilde{c}_{m'}$ are considered identical if they are matched by pymatgen's `StructureMatcher` [40] with default settings. The uniqueness and novelty indicators are given by

$$u_m = \mathbf{1}[\tilde{c}_m \text{ is not matched by any } \tilde{c}_{m'}, m' \neq m], \tag{8}$$

$$n_m = \mathbf{1}[\tilde{c}_m \text{ is not matched by any structure in the training set}]. \tag{9}$$

The uniqueness (U.) and novelty (N.) rates are $\sum_m v_m u_m / M$ and $\sum_m v_m n_m / M$, respectively.

Table 8: Quality metrics (%) for generated samples across different material generation tasks.

| Task | Method | Valid | S. | U.N. | T. | S.U.N. | S.U.N.T. |
|---|---|---|---|---|---|---|---|
| Space group | Unconditional | 73.0 | 26.5 | 72.4 | 2.3 | 26.0 | 1.3 |
| | ESS-Flow | 87.1 | 50.0 | 54.6 | 81.9 | 26.7 | **25.5** |
| Bulk modulus | Unconditional | 77.9 | 32.4 | 73.2 | 0.4 | 28.3 | 0.0 |
| | D-Flow | 75.2 | 31.0 | 70.2 | 0.4 | 26.3 | 0.0 |
| | PnP-Flow | 94.8 | 46.2 | 71.4 | 17.8 | 29.0 | 3.8 |
| | DAPS | 95.4 | 57.6 | 80.8 | 19.8 | 47.0 | 9.4 |
| | ESS-Flow | 98.3 | 56.4 | 46.1 | 79.6 | 19.3 | **13.7** |
| Shear modulus | Unconditional | 76.6 | 31.4 | 71.9 | 0.0 | 27.3 | 0.0 |
| | D-Flow | 78.6 | 32.2 | 73.2 | 0.0 | 27.5 | 0.0 |
| | PnP-Flow | 92.7 | 43.2 | 68.3 | 5.4 | 25.7 | 0.9 |
| | DAPS | 86.2 | 47.2 | 74.6 | 1.8 | 39.1 | 0.5 |
| | ESS-Flow | 98.2 | 50.3 | 30.5 | 59.9 | 10.0 | **4.7** |
| Band gap | Unconditional | 68.0 | 14.5 | 68.0 | 0.0 | 14.5 | 0.0 |
| | D-Flow | 69.8 | 16.2 | 69.7 | 0.0 | 16.1 | 0.0 |
| | PnP-Flow | 44.0 | 1.6 | 43.8 | 0.2 | 1.6 | 0.1 |
| | DAPS | 7.1 | 0.4 | 7.1 | 0.0 | 0.4 | 0.0 |
| | ESS-Flow | 48.8 | 16.9 | 48.0 | 23.2 | 16.5 | **16.0** |
| Energy above hull | Unconditional | 69.9 | 25.1 | 69.0 | 2.3 | 24.2 | 1.4 |
| | D-Flow | 69.7 | 25.2 | 68.8 | 2.7 | 24.3 | 1.8 |
| | PnP-Flow | 79.8 | 52.6 | 73.9 | 57.4 | 47.2 | 34.5 |
| | DAPS | 47.0 | 19.9 | 46.8 | 42.0 | 19.7 | 17.6 |
| | ESS-Flow | 73.3 | 44.5 | 64.9 | 73.3 | 37.6 | **37.6** |

Table 9: Number of function evaluations per generated sample for the baseline methods.

| Method | $\text{NFE}_{T_\theta}$ | $\text{NFE}_g$ | Gradients |
|---|---|---|---|
| D-Flow | $5 \times 10^3$ | $1 \times 10^2$ | $T_\theta$ and $g$ |
| PnP-Flow | $5 \times 10^3$ | $1 \times 10^2$ | $g$ only |
| DAPS | $5 \times 10^2$ | $2 \times 10^3$ | $g$ only |

**Threshold.** The threshold rate measures the proportion of generated materials meeting the desired target constraint for the specific task. The threshold indicator $t_m$ is task specific: for bulk modulus, shear modulus, and band gap, it requires the predicted property to be within 5% of the target value; for energy above hull, it requires the predicted value to be less than 0 eV; and for space group, it requires an exact match with the target space group $P6_3/mmc$. The threshold rate (T.) is $\sum_m v_m t_m / M$.

**Combined S.U.N.T. rate.** The S.U.N.T. rate is the proportion of generated materials satisfying all four criteria.

$$\text{S.U.N.T.} = \frac{1}{M} \sum_{m=1}^{M} v_m \cdot s_m \cdot u_m \cdot n_m \cdot t_m. \tag{10}$$

A breakdown of these metrics for the generated samples is given in Table 8.

### B.2 Runtime cost analysis

We report the number of function evaluations (NFE) of both the generative model $T_\theta$ and the potential function $g$ for all the methods in Tables 9 and 10. For ESS-Flow, the NFE varies based on the number of evaluations made along the ellipse per MCMC step ($n_{\text{ellipse}}$), where we report the mean values for each task, averaged over 1000 independent MCMC chains. Unlike ESS-Flow, gradient-based methods additionally require backward passes through the generative model, the potential function, or both, further increasing the computational cost, which is not reflected in this metric.

The NFE for ESS-Flow is an order of magnitude higher than the baseline methods, but this should be expected of a zeroth-order sampling method designed to solve a broader class of problems where

Table 10: Number of function evaluations per generated sample for ESS-Flow on each task.

| Task | $n_{\text{ellipse}}$ | $\text{NFE}_{T_\theta}$ | $\text{NFE}_g$ | Gradients |
|---|---|---|---|---|
| Bulk modulus | 6.7 | $7 \times 10^4$ | $1 \times 10^3$ | None |
| Shear modulus | 7.1 | $7 \times 10^4$ | $1 \times 10^3$ | None |
| Band gap | 17.4 | $4 \times 10^5$ | $9 \times 10^3$ | None |
| Energy above hull | 12.0 | $1 \times 10^5$ | $2 \times 10^3$ | None |
| Space group | 4.8 | $5 \times 10^4$ | $1 \times 10^3$ | None |

Table 11: Peak GPU memory usage (GB) for the different methods on material generation tasks.

| Method | Bulk modulus | Shear modulus | Band gap | Energy above hull | Space group |
|---|---|---|---|---|---|
| D-Flow | 6.91 | 6.91 | 62.22 | 27.90 | – |
| PnP-Flow | 0.86 | 0.86 | 2.28 | 1.57 | – |
| DAPS | 0.86 | 0.86 | 2.28 | 1.57 | – |
| ESS-Flow | 0.29 | 0.29 | 0.60 | 0.45 | 0.18 |

gradient-based methods fail. We have also not optimized the method for computational efficiency in this work. In the material generation experiments, we run parallel MCMC chains and treat only the final step as the generated sample, discarding all preceding steps as burn-in. Instead, retaining multiple thinned samples from each chain would lower the NFE per sample.

The overall computational cost can be further reduced by running ESS-Flow in parallel with large batch sizes, enabled by its low memory requirements. Table 11 shows the peak GPU memory usage for the different methods with a batch size of 50. ESS-Flow requires up to two orders of magnitude less memory than D-Flow.

### B.3 Additional evaluation of multi-fidelity sampling

We evaluate the distribution of material properties and quality of importance-reweighted samples compared to standard ESS-Flow samples. We resample 1000 source samples according to their self-normalized importance weights and obtain the corresponding material structures by integrating the ODE with fine discretization $\delta = 1/1000$. The resulting property distributions are shown in Figure 5, and the accuracy metrics are reported in Table 12.

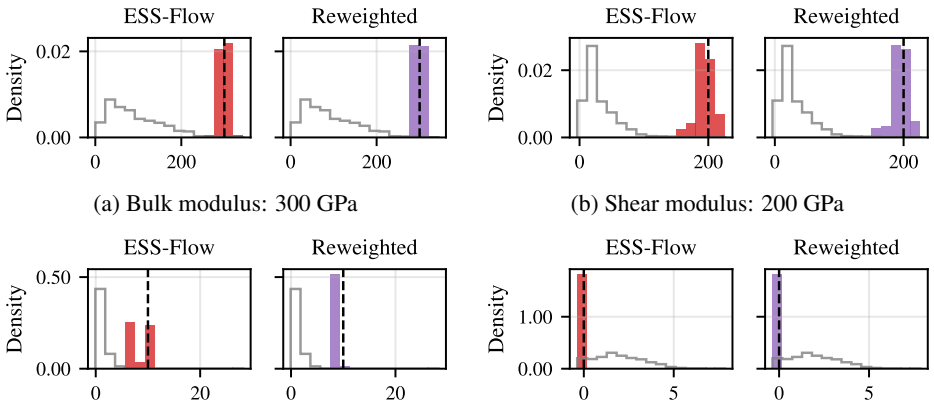
While the property distributions appear to better match the target, the quality of the reweighted samples as measured by the S.U.N.T. rate in Table 13 is considerably worse, though this metric may not capture all aspects of sample quality. The uniqueness rate drops due to resampling from the discrete weighted distribution. While simple importance reweighting can improve accuracy for moderately concentrated targets, more sophisticated multi-fidelity approaches are needed for sharply concentrated ones, which we leave for future work.

Table 12: Mean and standard deviation of absolute errors between the sample properties and targets, and of energy above hull values.

| Method | Bulk modulus | Shear modulus | Band gap | Energy above hull |
|---|---|---|---|---|
| ESS-Flow | $8.99 \pm 6.69$ | $10.53 \pm 9.21$ | $1.85 \pm 1.66$ | $-0.19 \pm 0.07$ |
| Reweighted | $9.36 \pm 6.70$ | $10.00 \pm 9.35$ | $0.85 \pm 0.47$ | $-0.22 \pm 0.08$ |

Table 13: S.U.N.T. rates (%) for standard ESS-Flow and importance-reweighted samples across tasks.

| Method | Bulk modulus | Shear modulus | Band gap | Energy above hull |
|---|---|---|---|---|
| ESS-Flow | 13.7 | 4.7 | 16.0 | 37.6 |
| Reweighted | 9.5 | 3.3 | 0.9 | 12.3 |




(a) Bulk modulus: 300 GPa

(b) Shear modulus: 200 GPa

(c) Band gap: 10 eV

(d) Energy above hull <0 eV

Figure 5: Comparison of sample property distributions between standard ESS-Flow and importance reweighted ESS-Flow. Property values are on the x-axis, the prior property distribution is shown in gray, and target values are indicated by dotted lines.