INTERACTING PARTICLE GUIDANCE FOR SAMPLING REWARD-TILTED GENERATIVE PRIORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Inference-time steering adapts pretrained diffusion and flow-based models to new tasks, e.g., to generate samples from a conditional distribution or samples with desired properties, without retraining. This can be formalized as sampling from a reward-tilted generative prior. As exact sampling from this distribution is intractable, guidance-based methods rely on approximations producing biased samples, and sequential Monte Carlo (SMC) methods correct for this bias using importance weights. However, while exact in the large particle limit, SMC suffers from weight degeneracy and particle collapse in practice. We propose interacting particle guidance (IPG), which replaces reweighting with transport. The particles interact through an additional drift, derived from the Feynman–Kac PDE to cancel the reweighting term, and remain unweighted. Choosing the drift in a reproducing kernel Hilbert space yields a closed-form solution that is cheap to compute, with negligible overhead compared to SMC. We demonstrate the method on Gaussian mixtures with known posteriors, and on high-dimensional image inpainting and protein structure inference tasks.

1 INTRODUCTION

Generative models based on dynamic measure transport, such as diffusion (Ho et al., 2020; Song et al., 2021) and flow-based models (Lipman et al., 2023; Liu et al., 2023; Albergo et al., 2025), are the state of the art approaches in a wide range of domains. Often, beyond unconditionally sampling from the data distribution, we may want to generate samples with specific properties or conditionally on some observed data. A convenient way to do so is to steer the generation process at test time, which allows reusing pretrained models without task-specific retraining. For a given reward function or observation likelihood, this can be formalized as sampling from an exponentially reward-tilted generative prior.

For differentiable reward functions, a heuristic approach is to include the reward gradient as a guidance term during generation (Chung et al., 2023; Boys et al., 2024). However, the modified process does not offer any guarantees of sampling from the reward-tilted prior. A principled approach is to correct for this using sequential Monte Carlo (SMC) methods (Naesseth et al., 2019), in which particles with self-normalized importance weights provide a weighted empirical distribution that converges to the target distribution in the large particle limit. In practice, however, when the prior is given by a generative model, the number of particles is limited by GPU memory, leading to pronounced weight degeneracy and particle collapse (Wu et al., 2023; Singhal et al., 2025; Skreta et al., 2025; Ekström Kelvinius et al., 2025). As a result, in practical implementations of SMC-based methods with generative priors and a moderate number of particles, reweighting and resampling are often interpreted as local best-of-N search strategies (Millard et al., 2026, Appendix A.6).

To mitigate the weight degeneracy, we analyze the dynamics of the intermediate target densities in the continuous time limit, given by the Feynman–Kac PDE (Skreta et al., 2025). By introducing an additional transport term that cancels the reweighting term, we aim to derive unweighted particle updates. Similar approaches that minimize the reweighting term have been explored for sampling unnormalized densities, where the added transport is parameterized by a neural network and learned over simulated trajectories (Albergo & Vanden-Eijnden, 2025), or at each intermediate step using the current particles (Arbel et al., 2021). To preserve the training-free nature of inference-time steering, the drift must be parameterized such that it admits a closed-form solution. Prior work has studied

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

linear parameterizations with predefined bases (Ren et al., 2026), but as these do not fully cancel the reweighting term, they still require periodic resampling to maintain a high effective sample size. Building on Stein transport (Nüsken, 2024), we show that choosing the drift in a reproducing kernel Hilbert space (RKHS) is surprisingly effective and enables sampling with an unweighted interacting particle system. The drift is available in closed form, and is cheap to compute for the moderate number of particles used with generative priors.

Contributions. (*i*) We propose interacting particle guidance (IPG), which augments guidance-based methods with an interacting drift estimated to cancel the reweighting term in the Feynman–Kac PDE, enabling sampling from a reward-tilted generative prior with an *unweighted* interacting particle system. The drift admits a closed-form solution, with negligible overhead to compute. (*ii*) We empirically show that IPG avoids particle collapse and outperforms prior methods, on Gaussian mixtures with known posteriors, as well as on high-dimensional image inpainting and protein structure inference with flow-based generative priors.

2 Preliminaries

2.1 Diffusion and flows

Consider a generative model that transports samples $\{X_0^i\}_{i=1}^N$ from a simple base distribution q_0 at time $t = 0$ to the data distribution q_1 at $t = 1$ along a learned vector field $v_t(x)$, by solving the ordinary differential equation (ODE)

$$\mathrm{d}X_t^i = v_t(X_t^i)\,\mathrm{d}t, \quad X_0^i \sim q_0. \tag{1}$$

Both flow-based models (Lipman et al., 2023; Liu et al., 2023; Albergo et al., 2025) and diffusion models using the probability flow ODE (Song et al., 2021) can be viewed as instances of this model class. The intermediate marginal densities q_t satisfy the continuity equation,

$$\partial_t q_t(x) = -\nabla \cdot (v_t(x) q_t(x)). \tag{2}$$

In the next section, we derive an analogous PDE for reward-tilted intermediate densities.

2.2 Reward tilting

Generating samples with specific properties or conditionally on observed data can be formulated as sampling from a reward-tilted prior $p_1(x) \propto q_1(x)\exp(R(x))$ for a reward function $R(x)$. We can do this by evolving particles along an interpolating path of densities $p_t(x) \propto q_t(x)\exp(r(x,t))$, with $r(x,0) = 0$ and $r(x,1) = R(x)$, so that $p_0 = q_0$ is the base distribution. We show in Appendix A that p_t satisfies

$$\partial_t p_t(x) = -\nabla \cdot (v_t(x) p_t(x)) + p_t(x)(g_t(x) - \mathbb{E}_{p_t}[g_t]), \tag{3}$$

where $g_t(x) = \partial_t r(x,t) + \langle v_t(x), \nabla r(x,t)\rangle$.

Equation (3) takes the form of a Feynman–Kac PDE (Skreta et al., 2025), and can be simulated with a weighted particle system $\{(X_t^i, w_t^i)\}_{i=1}^N$ whose weighted empirical distribution approximates p_t,

$$\mathrm{d}X_t^i = v_t(X_t^i)\,\mathrm{d}t, \quad \mathrm{d}\omega_t^i = g_t(X_t^i)\,\mathrm{d}t, \tag{4}$$

where the log-weights ω_t^i are initialized at zero, and $w_t^i = \exp(\omega_t^i)/\sum_k \exp(\omega_t^k)$ are the self-normalized importance weights. However, this approach is equivalent to importance sampling with the prior q_1 as the proposal, and performs poorly for small values of N.

Adding drift-diffusion. A popular strategy to let the reward influence the particle updates is to add and subtract $\sigma_t \Delta p_t(x)$, for $\sigma_t \in \mathbb{R}_{>0}$, in equation (3), and write

$$-\sigma_t \Delta p_t(x) = -\nabla \cdot ((\sigma_t \nabla \log p_t(x)) p_t(x)). \tag{5}$$

The resulting modified PDE for p_t is given by

$$\partial_t p_t(x) = -\nabla \cdot ((v_t(x) + \sigma_t \nabla \log p_t(x)) p_t(x)) + \sigma_t \Delta p_t(x) + p_t(x)(g_t(x) - \mathbb{E}_{p_t}[g_t]), \tag{6}$$

which, as before, can be simulated using the weighted stochastic differential equations (SDEs),

$$\mathrm{d}X_t^i = (v_t(X_t^i) + \sigma_t \nabla \log p_t(X_t^i))\,\mathrm{d}t + \sqrt{2\sigma_t}\,\mathrm{d}W_t, \quad \mathrm{d}\omega_t^i = g_t(X_t^i)\,\mathrm{d}t. \tag{7}$$

Here, $\nabla \log p_t(x) = \nabla \log q_t(x) + \nabla r(x,t)$, where the score $\nabla \log q_t(x)$ is directly estimated in diffusion models, or computed from v_t in flow-based models with a Gaussian base distribution (Albergo et al., 2025). For the choice $r(x,t) = \beta_t R(x)$, equation (7) recovers the reward-tilted SDE of Skreta et al. (2025), where the weighted SDEs are simulated using SMC, resampling the particles whenever the weights become too skewed. However, in high dimensions and with the few particles affordable with generative priors, the weights degenerate and resampling leads to particle collapse (Naesseth et al., 2019).

3 Method

The Langevin terms in equation (7) influence the particle updates, but leave the weights unchanged. Our methodology is based on compensating for the weights by an additional drift, as detailed below.

Adding drift-reweighting. For any vector field u_t, we add and subtract $\nabla \cdot (u_t(x)p_t(x))$ in equation (3), and write

$$\nabla \cdot (u_t(x)p_t(x)) = p_t(x)(S_{p_t}u_t(x)), \tag{8}$$

where $S_{p_t}u_t = \nabla \cdot u_t + \langle u_t, \nabla \log p_t \rangle$ is the Stein operator. The modified PDE for p_t is

$$\partial_t p_t(x) = -\nabla \cdot ((v_t(x) + u_t(x))p_t(x)) + p_t(x)(S_{p_t}u_t(x) + g_t(x) - \mathbb{E}_{p_t}[g_t]). \tag{9}$$

As $\mathbb{E}_{p_t}[S_{p_t}u_t(x)] = 0$ by Stein's identity, p_t can be approximated with the weighted particle system

$$\mathrm{d}X_t^i = (v_t(X_t^i) + u_t(X_t^i))\,\mathrm{d}t, \quad \mathrm{d}\omega_t^i = (S_{p_t}u_t(X_t^i) + g_t(X_t^i))\,\mathrm{d}t. \tag{10}$$

As before, the Langevin terms can be added without changing the weight updates, since they leave p_t invariant.

We note that the drift u_t can be chosen to reduce the variance of the particle weights, and in particular, if u_t solves the PDE

$$S_{p_t}u_t(x) + g_t(x) - \mathbb{E}_{p_t}[g_t] = 0, \tag{11}$$

then the log-weights remain constant and p_t is approximated by *unweighted* particles. We refer to such u_t as a corrective drift.

3.1 Interacting particle guidance

As solving the PDE (11) exactly is intractable, we minimize the squared residual locally at each time. Following Stein transport (Nüsken, 2024), choosing $u_t \in \mathcal{H}_k^d$, where $\mathcal{H}_k$ is the reproducing kernel Hilbert space (RKHS) for a kernel k and $\mathcal{H}_k^d$ is its d-fold Cartesian product, and adding Tikhonov regularization with $\lambda > 0$, yields the objective

$$\mathcal{L}(u_t) = \mathbb{E}_{p_t}[(S_{p_t}u_t(x) + g_t(x) - \mathbb{E}_{p_t}[g_t])^2] + \lambda \|u_t\|^2_{\mathcal{H}_k^d}. \tag{12}$$

Approximating $\mathbb{E}_{p_t}[\cdot]$ with the empirical distribution given by the particles $\{X_t^i\}_{i=1}^N$, the unique minimizer is

$$u_t(x) = \frac{1}{N}\sum_{j=1}^N \phi_j [k(x, X_t^j)\nabla \log p_t(X_t^j) + \nabla_{X_t^j} k(x, X_t^j)], \quad \left(\frac{1}{N}\xi + \lambda I_N\right)\phi = -\mathrm{g}_t, \tag{13}$$

where the coefficients $\{\phi_j\}_{j=1}^N$ are obtained by solving the $N \times N$ linear system. Here, $\xi \in \mathbb{R}^{N\times N}$ is the Gram matrix of the Stein kernel, with $[\xi]_{ij} = S_{p_t}^{X_t^i} S_{p_t}^{X_t^j} k(X_t^i, X_t^j)$ and the superscript denoting the variable the Stein operator acts on, and $[\mathrm{g}_t]_i = g_t(X_t^i) - \frac{1}{N}\sum_{j=1}^N g_t(X_t^j)$.

As the self-normalized importance weights are invariant to adding a constant to the log-weights of all particles, the weight updates reduce to

$$\mathrm{d}\omega_t^i = (S_{p_t}u_t(X_t^i) + g_t(X_t^i) - \frac{1}{N}\sum_{j=1}^N g_t(X_t^j))\,\mathrm{d}t = \left(\frac{1}{N}\xi\phi + \mathrm{g}_t\right)_i \mathrm{d}t = -\lambda\phi_i\,\mathrm{d}t. \tag{14}$$

However, as in Stein transport, the weight updates can be ignored in practice by choosing a small λ, yielding an unweighted interacting particle system that we refer to as interacting particle guidance (IPG), described in Algorithm 1.

Algorithm 1 Interacting particle guidance (IPG and IPG-CV)

Require: v_t and $\nabla \log q_t(x)$ from the generative prior, reward schedule $r(x,t)$, kernel k, regularization λ, noise schedule σ_t, time steps $\{t_k\}_{k=0}^K$, number of particles N
Initialize $X_0^i \sim q_0$ for $i = 1, \dots, N$
for $t = t_0, \dots, t_{K-1}$ **do**
  $\nabla \log p_t(X_t^i) \leftarrow \nabla \log q_t(X_t^i) + \nabla r(X_t^i, t)$,
  Compute $u_t(X_t^i)$ using equations (13) for IPG, or (15) for IPG-CV,
  $X_{t+\Delta t}^i \leftarrow X_t^i + (v_t(X_t^i) + \sigma_t \nabla \log p_t(X_t^i) + u_t(X_t^i))\Delta t + \sqrt{2\sigma_t \Delta t}\, \epsilon_t^i, \quad \epsilon_t^i \sim \mathcal{N}(0, I)$
return $\{X_1^i\}_{i=1}^N$

Using a control variate. Alternatively, $\mu_t = \mathbb{E}_{p_t}[g_t]$ in objective (12) can be treated as a free parameter to be jointly minimized. With the empirical approximation for the outer expectation, the optimal $\mu_t = \frac{1}{N}\sum_{i=1}^N (S_{p_t} u_t(X_t^i) + g_t(X_t^i))$ resembles using a Stein control variate (Oates et al., 2017) to estimate $\mathbb{E}_{p_t}[g_t]$. The unique minimizer is now given by (see Appendix B),

$$u_t(x) = \frac{1}{N}\sum_{j=1}^N (\Pi\phi)_j [k(x, X_t^j)\nabla \log p_t(X_t^j) + \nabla_{X_t^j} k(x, X_t^j)], \qquad \left(\frac{1}{N}\Pi\xi\Pi + \lambda I_N\right)\phi = -\mathrm{g}_t, \tag{15}$$

where $\Pi = I_N - \frac{1}{N}\mathbf{1}\mathbf{1}^\top$ is the centering matrix. The weight updates once again reduce to,

$$\mathrm{d}\omega_t^i = (S_{p_t} u_t(X_t^i) + g_t(X_t^i) - \mu_t)\,\mathrm{d}t = \left(\frac{1}{N}\Pi\xi\Pi\phi + \mathrm{g}_t\right)_i \mathrm{d}t = -\lambda\phi_i\,\mathrm{d}t, \tag{16}$$

and can be ignored by choosing a small λ. We refer to the resulting method as IPG-CV in Algorithm 1, emphasizing the use of a Stein control variate.

Connection to Stein transport. The objective (12) has the same form as in Stein transport (Nüsken, 2024), where particles are transported deterministically from a tractable prior π_0 to the posterior along $\pi_t \propto \pi_0 \exp(th)$, where h is the log-likelihood, by the corrective drift alone. In our notation, this corresponds to the case with $v_t = 0$, $\sigma_t = 0$, $q_t = \pi_0$ and $r(x,t) = th(x)$. In IPG, the path is induced by a generative model whose pointwise density is unavailable, particles are transported by the generative drift and guidance, and the corrective drift only accounts for the remaining reweighting term.

4 Connections to prior work

We give an overview of inference-time steering methods in Section 4.1, and relate our framework to prior work by discussing alternative approaches to the corrective drifts and p_t invariant correctors in Sections 4.2 and 4.3, respectively.

4.1 Guidance and Monte Carlo approximations

Guidance methods. Sampling from a reward-tilted generative prior can be achieved either by finetuning the model (Venkatraman et al., 2024; Domingo i Enrich et al., 2025; Potaptchik et al., 2026a), or by modifying the generation process at test time. Training-free guidance methods (Chung et al., 2023; Song et al., 2023; Pokle et al., 2024; Kim et al., 2025) adapt the generative ODE/SDE by adding the reward gradient, typically evaluated at the denoised estimate, but do not sample from the target distribution due to the approximations involved. Nguyen et al. (2026) consider correcting the distribution of denoised estimates using Stein variational gradient descent (SVGD), but this relies on the prior score evaluated at clean data where it is unreliable.

Monte Carlo methods. SMC-based methods correct for the bias in guidance through weighted particles and resampling (Wu et al., 2023; Dou & Song, 2024; Singhal et al., 2025; Skreta et al., 2025; Ekström Kelvinius et al., 2025). Alternatively, when flow maps trained for one-step sampling of the final state given an intermediate noisy state are available, a consistent Monte Carlo estimate of

the guidance drift itself can be obtained (Potaptchik et al., 2026b; Holderrieth et al., 2026; Pan et al., 2026). Markov chain Monte Carlo (MCMC) methods that sample the pullback of the reward-tilted prior onto the base also target the correct distribution, and are asymptotically exact in the large time limit (Kalaivanan et al., 2025; Wang et al., 2026).

4.2 Parametric approximations of the corrective drift

Neural networks. In the related task of sampling unnormalized probability densities, weight degeneracy is mitigated by learning additional transport, where the corrective drift is parameterized by a neural network (Arbel et al., 2021; Vargas et al., 2024; Albergo & Vanden-Eijnden, 2025). As a specific example, NETS (Albergo & Vanden-Eijnden, 2025) parameterizes both the drift $u(x,t)$ and the log-partition function $F(t) = \log Z_t = \log Z_0 + \int_0^t \mathbb{E}_{p_s}[g_s]\,\mathrm{d}s$ with a network, trained over simulated trajectories to minimize

$$\mathcal{L}(u,F) = \int_0^1 \mathbb{E}_{p_t}[(S_{p_t}u(t,x) + g_t(x) - \partial_t F(t))^2]\,\mathrm{d}t. \tag{17}$$

For fixed t, this coincides with the IPG-CV objective up to regularization, where μ_t denotes $\partial_t F(t)$.

Predefined bases. DriftLite (Ren et al., 2026) adapts NETS for reward-tilting by minimizing objective (12), without regularization, over drifts of the form

$$u_t(x) = \phi_1 v_t(x) + \phi_2 \nabla \log q_t(x) + \phi_3 \nabla r(x,t). \tag{18}$$

Approximating the expectations in the objective with the weighted empirical distribution yields a closed-form solution by solving a 3×3 linear system. However, it requires evaluating the Stein operator $S_{p_t}u_t$ at the particles and involves $\nabla \cdot v_t$, $\Delta \log q_t$ and Δr. These are coarsely approximated with Hutchinson's trace estimator, and as the corrective drift does not drive the reweighting term close to zero, reweighting and resampling are necessary.

Gradient field. Restricting the corrective drift to a gradient field $u_t(x) = \nabla\psi_t(x)$ for $\psi_t \colon \mathbb{R}^d \to \mathbb{R}$ allows using an alternate objective

$$\mathcal{L}(\psi_t) = \mathbb{E}_{p_t}\left[\frac{1}{2}\|\nabla\psi_t(x)\|^2 - \psi_t(x)(g_t(x) - \mathbb{E}_{p_t}[g_t])\right]. \tag{19}$$

This has been explored for ψ_t parameterized by a neural network (Albergo & Vanden-Eijnden, 2025) and as a linear combination over predefined bases (Ren et al., 2026). However, these do not perform as well as directly parameterizing u_t. The objective (19) also admits a closed-form solution for $\psi_t \in \mathcal{H}_k$, which resembles the kernel Fisher–Rao flow of Maurais & Marzouk (2024) for sampling unnormalized probability densities, similar to how IPG relates to Stein transport.

4.3 Predictor-corrector methods

In SMC methods, resample-move (Gilks & Berzuini, 2001) rejuvenates particles after resampling by applying a p_t-invariant MCMC kernel. The Langevin terms in equation (7) can be seen as an infinitesimal such move applied jointly with the transport rather than as a separate step as in predictor-corrector methods (Song et al., 2021). Consequently, we can replace the Langevin terms by any continuous time process that leaves p_t invariant, and apply any number of such corrector steps at a fixed time t after resampling in SMC, or directly in the case of IPG.

Interacting correctors. More generally, we can consider the N-particle system on the augmented state space $\bar{x} = (x^1, \ldots, x^N)$ with $p_t^N(\bar{x}) = \prod_{i=1}^N p_t(x^i)$, and use processes that leave p_t^N invariant as correctors. This includes interacting dynamics such as affine invariant Langevin dynamics (Garbuno-Inigo et al., 2020) and suitably noise-perturbed SVGD (Gallego & Insua, 2018; Nüsken & Renger, 2023). In Appendix D.1, we evaluate using noisy SVGD in place of the Langevin terms.

5 Experiments

We evaluate our interacting particle guidance (IPG) and its control variate version (IPG-CV), as described in Algorithm 1, on high-dimensional Gaussian mixture models (GMMs) with known posteriors, and on image inpainting and protein structure inference problems. In all our experiments, we

Table 1: Mean and standard deviation of the metrics on the GMM experiment. We use systematic resampling and "Adaptive" refers to resampling whenever ESS/N falls below 0.5.

Method	Resampling	ESS/N ↑	Mean error ↓	MMD ↓	SWD ↓
FKC	None	0.006 ± 0.002	7.593 ± 1.337	0.876 ± 0.165	0.682 ± 0.112
	Adaptive	0.771 ± 0.203	4.485 ± 0.872	0.184 ± 0.042	0.311 ± 0.055
	Every step	1.000 ± 0.000	3.685 ± 0.741	0.148 ± 0.021	0.260 ± 0.043
DriftLite	None	0.011 ± 0.004	5.691 ± 1.194	0.619 ± 0.139	0.509 ± 0.092
	Adaptive	0.864 ± 0.169	2.178 ± 0.357	0.097 ± 0.007	0.168 ± 0.021
	Every step	1.000 ± 0.000	2.483 ± 0.243	0.085 ± 0.004	0.182 ± 0.012
IPG	None	–	$\mathbf{0.841 \pm 0.051}$	$\mathbf{0.012 \pm 0.002}$	$\mathbf{0.093 \pm 0.002}$
IPG-CV	None	–	0.842 ± 0.031	0.013 ± 0.002	$\mathbf{0.093 \pm 0.002}$

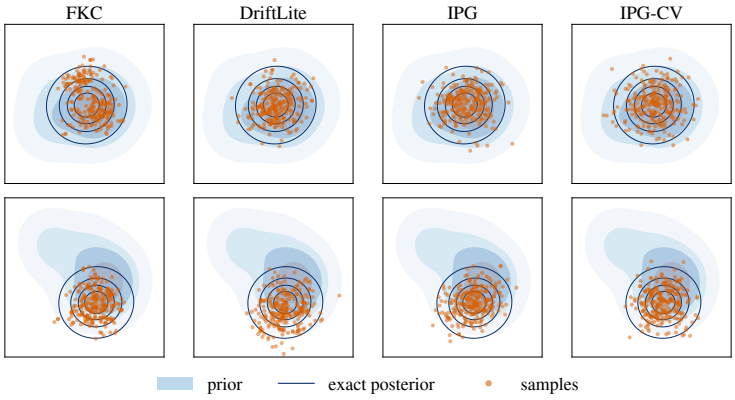

Figure 1: Samples on random 2-D slices of the 256-dimensional posterior.

choose the radial basis function (RBF) kernel, with the bandwidth set using the median heuristic at each time step. We compare against methods designed to be consistent, such as Feynman–Kac correctors (FKC) (Skreta et al., 2025) which simulates the weighted SDE in equation (7), and DriftLite (Ren et al., 2026) which adds a corrective drift restricted to predefined bases.

5.1 Gaussian mixture models

We consider a 256-dimensional GMM as the prior $q_1(x)$, and a linear-Gaussian observation $y = Ax + \sigma_{\text{obs}}\epsilon$, where $y \in \mathbb{R}^{128}$. To sample from the posterior $p_1(x) = q_1(x|y)$, we set the log-likelihood as the reward $R(x)$, and propagate particles along the interpolation $p_t(x) \propto q_t(x)\exp(tR(x))$ from $t = 0$ to $t = 1$. The intermediate marginal densities $q_t(x)$ correspond to those of a noising Ornstein–Uhlenbeck (OU) process, whose time-reversed SDE is known. We use the associated probability flow ODE as the generative process instead of training a flow-based model, and compare samples against the exact posterior, which is available in closed form.

We evaluate FKC and DriftLite under different resampling policies, and compare them against both versions of interacting particle guidance, which do not require resampling. We run the methods with $N = 256$ particles and report the effective sample size (ESS), mean error, maximum mean discrepancy (MMD), and sliced 2-Wasserstein distance (SWD), over 5 randomly generated priors and observations in Table 1. Figure 1 compares samples from FKC and DriftLite, both using systematic resampling at every step, with those from IPG and IPG-CV. Further details on the hyperparameters and metrics can be found in Appendix D. Additional experiments replacing the Langevin terms with noisy SVGD (Gallego & Insua, 2018; Nüsken & Renger, 2023), and using a neural corrective drift, are in Appendices D.1 and D.2, respectively.

Without resampling, the baselines suffer from severe weight degeneracy, with one to three effective particles out of 256. While resampling at each step trivially yields ESS/$N = 1$, it does not imply

Table 2: Results on the image inpainting tasks with $N = 16$ particles. "Avg." refers to the mean of the metrics computed between each of the N samples and the ground truth, and "Best" takes the best-of-N sample metrics before averaging across the test images.

		PSNR ↑		SSIM ↑		LPIPS ↓		Diversity ↑	
Mask	Method	Avg.	Best	Avg.	Best	Avg.	Best	Cov. trace	LPIPS div.
Half	FKC	**15.99**	15.99	**0.4534**	0.4537	**0.3352**	0.3341	24.61	0.0041
	DriftLite	15.51	15.52	0.4392	0.4401	0.3466	0.3443	70.56	0.0125
	IPG	15.74	**17.98**	0.4427	**0.4894**	0.3455	**0.3058**	**7990.21**	0.2307
	IPG-CV	15.68	17.79	0.4418	0.4889	0.3462	0.3060	7742.43	**0.2311**
Box	FKC	**24.53**	24.53	0.6737	0.6739	**0.1866**	0.1859	14.02	0.0020
	DriftLite	24.45	24.45	**0.6742**	**0.6750**	0.1884	0.1863	46.97	0.0065
	IPG	24.29	25.22	0.6624	0.6708	0.1981	0.1830	1005.08	0.0957
	IPG-CV	24.26	**25.28**	0.6622	0.6711	0.1994	**0.1819**	**1028.86**	**0.0964**

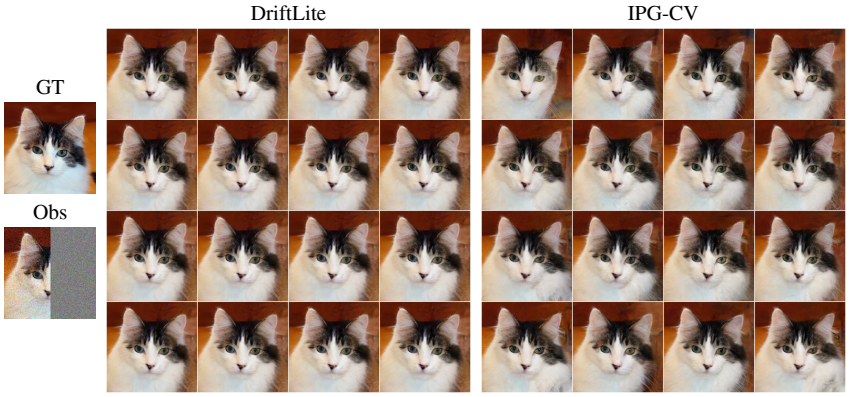

Figure 2: DriftLite suffers from particle collapse, producing indistinguishable samples, while IPG-CV produces samples with perceptible differences.

accurate samples as shown by the other metrics. With a small regularization constant, the weight updates in IPG and IPG-CV are negligible, and they outperform the baselines across every metric.

5.2 Image Inpainting

We now evaluate the methods on inpainting high-resolution (256×256) images, using a flow-based model (Liu et al., 2023) trained on AFHQ Cat (Choi et al., 2020) as the generative prior $q_1(x)$. Given a noisy observation $y = Ax + \sigma_{\text{obs}}\epsilon$, we aim to sample the posterior $p_1(x) = q_1(x|y)$, which corresponds to a reward-tilted prior with the log-likelihood as the reward $R(x)$. Here, A denotes an operator that either applies a box mask at the center of the image or masks the entire right half of the image. We propagate particles along the interpolation $p_t(x) \propto q_t(x)\exp(tR(\hat{x}_t))$ from $t = 0$ to $t = 1$, where $\hat{x}_t = x_t + (1-t)v_t(x_t)$ is the denoised estimate.

All methods use $N = 16$ particles, limited by GPU memory. Unlike in the GMM case, evaluating $S_{p_t}u_t(x)$ exactly for DriftLite is prohibitively expensive, as it requires $\nabla \cdot v_t(x), \Delta \log q_t(x)$ and $\Delta r(x,t)$, which we approximate using Hutchinson's trace estimator. Following Skreta et al. (2025), we apply systematic resampling at each time step for both FKC and DriftLite.

We report the commonly used performance metrics, peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM) and learned perceptual image patch similarity (LPIPS). Additionally, we measure the diversity among the samples generated per observation using the trace of the sample covariance in the raw pixel space and the mean pairwise LPIPS. The average values of these metrics, computed over 50 test images, are given in Table 2. Samples from DriftLite are compared against those from IPG-CV in Figure 2. Further details on the metrics and additional visualizations of the generated samples are provided in Appendix E.

The diversity metrics indicate that FKC and DriftLite suffer from severe particle collapse, producing near identical samples, whereas IPG and IPG-CV yield perceptually diverse samples. While the baselines achieve marginally higher average metrics, we note that the mean PSNR, SSIM and LPIPS computed between the generated samples and ground truth image are imperfect measures of posterior sample quality. Samples from a diverse posterior might yield worse average scores than collapsed samples that happen to be close to the ground truth. We therefore also report best-of-N metrics, which show that IPG and IPG-CV produce samples that fit the ground truth better as well.

We note that both IPG and IPG-CV incur negligible computational overhead (122.9s per test image) compared to FKC (122.1s per test image). This is because the cost of the $N \times N$ linear solve is low for the typical number of particles used when sampling reward-tilted generative priors, and the matrix entries only require the kernel, its gradients, and terms already used in the particle updates. In contrast, DriftLite requires evaluating Hutchinson estimates of $S_{p_t} u_t(x)$ at the particle positions and is much slower (444.3s per test image).

5.3 Protein structures from pairwise distance measurements

We compare the methods on inferring the backbone structure of M-residue proteins from inter-residue distances, using the flow-based model Proteina (Geffner et al., 2025) as the generative prior $q_1(x)$. The backbone structure $x = (x_1, \dots, x_M) \in \mathbb{R}^{M\times 3}$ is given by the position of α-carbon in each residue, and the observation $y = \mathcal{A}(x) + \sigma_{\text{obs}}\epsilon \in \mathbb{R}^K$ consists of K noisy pairwise distances, $[\mathcal{A}(x)]_k = \|x_{i_k} - x_{j_k}\|_2$ for distinct pairs (i_k, j_k) of residues drawn uniformly at random, with K as 3% of all possible pairs. As in the image inpainting task, we sample the posterior $p_1(x) = q_1(x|y)$, which corresponds to setting the log-likelihood as the reward $R(x)$, by propagating particles along the interpolation $p_t(x) \propto q_t(x)\exp(tR(\hat{x}_t))$. Here, $\hat{x}_t = x_t + (1-t)v_t(x_t)$ is the denoised estimate.

As the observation operator depends only on pairwise distances, which are invariant to rotations, and Proteina is trained with random rotations as data augmentation, the orientation of the structures is uniform under the posterior. This is a deliberate choice which lets us test whether the methods preserve this diversity. We also note that Proteina produces designable structures at a rate below 20%, unless heuristic noise reduction is applied during generation (Geffner et al., 2025). As we focus on evaluating inference-time steering, we use the unmodified prior for all methods and accept the low designability as a byproduct.

All methods use $N = 16$ particles. As in the image inpainting task, the divergences required for DriftLite are approximated with Hutchinson's trace estimator, and systematic resampling is applied at each time step for both FKC and DriftLite. We measure accuracy using the root mean square deviation (RMSD) between the generated structures $\{x^i\}_{i=1}^N$ and the ground truth structure after rotational alignment (RMSD_{gt}), and the RMSD between the pairwise distances in the generated samples $\{\mathcal{A}(x^i)\}_{i=1}^N$ and the observation y (RMSD_{obs}). To measure the diversity, for each pair of generated structures, we compute the angle of rotation needed to align them in degrees (Rot. div.) and the RMSD between them after this alignment (RMSD div.). The expected Rot. div. is 126.5°, which is the average angle of rotation between two independent random orientations of a structure. Following standard practice, a structure is considered designable when its self-consistency RMSD (see Appendix F) is below 2 Å. We report the fraction of test cases for which a method generates at least one designable structure. The average metrics computed over 15 protein structures are given in Table 3. Further details on the test proteins and evaluation metrics, along with visualizations of the generated samples, are provided in Appendix F.

Both the diversity metrics in Table 3 and the samples visualized in Figure 3 show that FKC and DriftLite suffer from severe particle collapse, producing structures with near identical shapes and orientations. In contrast, both IPG and IPG-CV produce samples with diverse orientations, with Rot. div. close to the expected value of 126.5°. The average RMSD_{gt} is high for all the methods, which is consistent with the low designability of the Proteina prior. However, the best-of-N RMSD_{gt} shows that both IPG and IPG-CV generate structures close to the ground truth, and IPG-CV produces at least one designable structure in every test case. The baselines, on the other hand, collapse around structures that are often undesignable, resulting in a much higher best-of-N RMSD_{gt}. In terms of runtime, IPG and IPG-CV (126.0s per test structure) are nearly as fast as FKC (125.7s per test structure), while DriftLite is much slower (460.6s per test structure).

Table 3: Results on the protein structure inference task with $N = 16$ particles. "Avg." refers to the mean of the metrics computed between each of the N samples and the ground truth, and "Best" takes the best-of-N sample metrics for each test protein before averaging.

	RMSD_{obs} (Å) ↓		RMSD_{gt} (Å) ↓		Diversity ↑		
Method	Avg.	Best	Avg.	Best	Rot. div. (°)	RMSD div. (Å)	Any designable ↑
FKC	1.225	1.200	8.119	8.060	0.6	0.677	7/15
DriftLite	**1.188**	1.163	**6.907**	6.840	0.7	0.756	8/15
IPG	1.308	1.053	7.881	1.298	122.1	**7.981**	13/15
IPG-CV	1.290	**1.048**	7.543	**1.232**	**126.9**	7.800	**15/15**

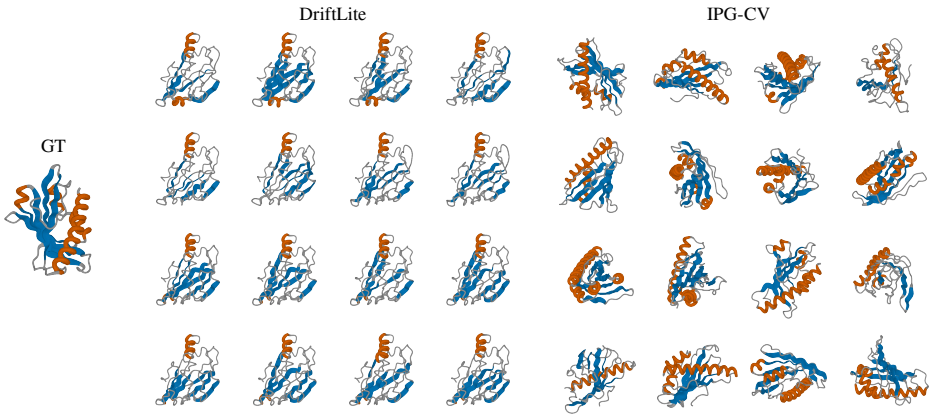

Figure 3: Samples produced on the backbone structure inference task for protein `3OJI`. They are intentionally not aligned with the ground truth to highlight the particle collapse in DriftLite. Samples from IPG-CV contain structures close to the ground truth, and have random orientations as expected.

6 Conclusion

We introduce interacting particle guidance (IPG), a training-free method for sampling reward-tilted generative priors, in which an interacting corrective drift is derived to cancel the reweighting term in the Feynman–Kac PDE. Solving for the drift in an RKHS yields a closed-form solution with negligible computational overhead compared to existing SMC methods. We evaluate IPG on high-dimensional Gaussian mixtures, image inpainting and protein structure inference problems, and show significant improvements over prior work. Specifically, we see improvements over SMC-based formulations, despite the fact that IPG and SMC/FKC are derived from the same Feynman–Kac PDE. We conjecture that the continuous "nudging" of particles in combination with Langevin-style guidance allows IPG to track the evolution of p_t closely despite finite-sample errors in the forces. This is in contrast with a pure SMC-based method where any lag in the sampling dynamics with respect to p_t accumulates in the importance weights until they degenerate or trigger resampling.

Despite the compelling empirical performance, like any sampling algorithm, the method has practical limitations. By replacing reweighting and resampling with transport, the method avoids particle collapse, but may struggle to move particles across large energy barriers in a multi-modal distribution when the intermediate reward gradients $\|\nabla r(x,t)\|$ are much weaker than the prior score $\|\nabla \log q_t(x)\|$. This could be remedied either by tuning the interpolating path p_t, or by increasing the regularization constant λ, which reintroduces particle weights, and thereby resampling, while retaining the variance-reducing corrective transport. Here, we demonstrated the method using a simple RBF kernel, which performs well but is agnostic to the data. A promising direction for future work is to use learned kernels (Galashov et al., 2025) or to operate in a pretrained latent space, either of which could improve the performance further.

AI USE STATEMENT

In this work, we used coding agents to implement the methods used in our experiments, and verified the outputs for correctness. Additionally, we used generative AI tools to polish the writing, with minor revisions to grammar and phrasing. We did not use generative AI tools for any other tasks that require disclosure. We take full responsibility for the final content of this work, including all text and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We clearly describe our algorithm in Section 3. Details of our experiments, including the models, datasets, and hyperparameters used, are described in Section 5 and the appendix. Our code is not included with this submission, but will be made publicly available upon publication.

REFERENCES

Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

Michael Samuel Albergo and Eric Vanden-Eijnden. NETS: A non-equilibrium transport sampler. In *International Conference on Machine Learning*, 2025.

Michael Arbel, Alex Matthews, and Arnaud Doucet. Annealed flow transport Monte Carlo. In *International Conference on Machine Learning*, 2021.

Benjamin Boys, Mark Girolami, Jakiw Pidstrigach, Sebastian Reich, Alan Mosca, and Omer Deniz Akyildiz. Tweedie moment projected diffusions for inverse problems. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse image synthesis for multiple domains. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8185–8194, 2020.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023.

Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using ProteinMPNN. *Science*, 378(6615):49–56, 2022.

Carles Domingo i Enrich, Michal Drozdzal, Brian Karrer, and Ricky TQ Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *International Conference on Learning Representations*, volume 2025, pp. 53791–53846, 2025.

Zehao Dou and Yang Song. Diffusion posterior sampling for linear inverse problem solving: A filtering perspective. In *International Conference on Learning Representations*, 2024.

Filip Ekström Kelvinius, Zheng Zhao, and Fredrik Lindsten. Solving linear-Gaussian Bayesian inverse problems with decoupled diffusion sequential Monte Carlo. In *International Conference on Machine Learning*, 2025.

Alexandre Galashov, Valentin De Bortoli, and Arthur Gretton. Deep MMD gradient flow without adversarial training. In *International Conference on Learning Representations*, 2025.

Victor Gallego and David Rios Insua. Stochastic gradient MCMC with repulsive forces. *arXiv preprint arXiv:1812.00071*, 2018.

Alfredo Garbuno-Inigo, Nikolas Nüsken, and Sebastian Reich. Affine invariant interacting Langevin dynamics for Bayesian inference. *SIAM Journal on Applied Dynamical Systems*, 19(3):1633–1658, 2020.

Tomas Geffner, Kieran Didi, Zuobai Zhang, Danny Reidenbach, Zhonglin Cao, Jason Yim, Mario Geiger, Christian Dallago, Emine Kucukbenli, Arash Vahdat, and Karsten Kreis. Proteina: Scaling flow-based protein structure generative models. In *International Conference on Learning Representations*, 2025.

Walter R Gilks and Carlo Berzuini. Following a moving target—Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63 (1):127–146, 2001.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Peter Holderrieth, Douglas Chen, Luca Eyring, Ishin Shah, Giri Anantharaman, Yutong He, Zeynep Akata, Tommi Jaakkola, Nicholas Matthew Boffi, and Max Simchowitz. Diamond maps: Efficient reward alignment via stochastic flow maps. In *International Conference on Machine Learning*, 2026.

Adhithyan Kalaivanan, Zheng Zhao, Jens Sjölund, and Fredrik Lindsten. ESS-Flow: Training-free guidance of flow-based models as inference in source space. *arXiv preprint arXiv:2510.05849*, 2025.

Jeongsol Kim, Bryan Sangwoo Kim, and Jong Chul Ye. FlowDPS: Flow-driven posterior sampling for inverse problems. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12328–12337, 2025.

Patrick Kunzmann and Kay Hamacher. Biotite: a unifying open source computational biology framework in Python. *BMC bioinformatics*, 19(1):346, 2018.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

Aimee Maurais and Youssef Marzouk. Sampling in unit time with kernel Fisher-Rao flow. In *International Conference on Machine Learning*, 2024.

Andrew Millard, Fredrik Lindsten, and Zheng Zhao. Particle-guided diffusion models for partial differential equations. In *Forty-third International Conference on Machine Learning*, 2026.

Christian A Naesseth, Fredrik Lindsten, and Thomas B Schön. Elements of sequential Monte Carlo. *Foundations and Trends® in Machine Learning*, 12(3):187–306, 2019.

Van Khoa Nguyen, Lionel Blondé, and Alexandros Kalousis. Stein diffusion guidance: Training-free posterior correction for sampling beyond high-density regions. In *Forty-third International Conference on Machine Learning*, 2026.

Nikolas Nüsken. Stein transport for Bayesian inference. *arXiv preprint arXiv:2409.01464*, 2024.

Nikolas Nüsken and Michiel Renger. Stein variational gradient descent: Many-particle and long-time asymptotics. *Foundations of Data Science*, 5(3):286–320, 2023.

Chris J Oates, Mark Girolami, and Nicolas Chopin. Control functionals for Monte Carlo integration. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 79(3):695–718, 2017.

Zhengkai Pan, Peter Potaptchik, Wenxi Yao, Michael S Albergo, and Jakiw Pidstrigach. Itô maps for any-step SDEs. *arXiv preprint arXiv:2606.11156*, 2026.

Ashwini Pokle, Matthew J. Muckley, Ricky T. Q. Chen, and Brian Karrer. Training-free linear image inverses via flows. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Peter Potaptchik, Lee Cheuk Kit, and Michael Samuel Albergo. Tilt matching for scalable sampling and fine-tuning. In *International Conference on Machine Learning*, 2026a.

Peter Potaptchik, Adhi Saravanan, Abbas Mammadov, Alvaro Prat, Michael Samuel Albergo, and Yee Whye Teh. Meta flow maps enable scalable reward alignment. In *International Conference on Machine Learning*, 2026b.

Yinuo Ren, Wenhao Gao, Lexing Ying, Grant Rotskoff, and Jiequn Han. DriftLite: Lightweight drift control for inference-time scaling of diffusion models. In *International Conference on Learning Representations*, 2026.

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. In *International Conference on Machine Learning*, 2025.

Marta Skreta, Tara Akhound-Sadegh, Viktor Ohanesian, Roberto Bondesan, Alan Aspuru-Guzik, Arnaud Doucet, Rob Brekelmans, Alexander Tong, and Kirill Neklyudov. Feynman-Kac correctors in diffusion: Annealing, guidance, and product of experts. In *International Conference on Machine Learning*, 2025.

Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, et al. AlphaFold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research*, 50(D1):D439–D444, 2022.

Francisco Vargas, Shreyas Padhy, Denis Blessing, and Nikolas Nüsken. Transport meets variational inference: Controlled Monte Carlo diffusions. In *International Conference on Learning Representations*, pp. 55236–55278, 2024.

Siddarth Venkatraman, Moksh Jain, Luca Scimeca, Minsu Kim, Marcin Sendera, Mohsin Hasan, Luke Rowe, Sarthak Mittal, Pablo Lemos, Emmanuel Bengio, et al. Amortizing intractable inference in diffusion models for vision, language, and control. *Advances in Neural Information Processing Systems*, 37:76080–76114, 2024.

Zifan Wang, Alice Harting, Matthieu Barreau, Michael Zavlanos, and Karl H Johansson. Source-guided flow matching. In *International Conference on Learning Representations*, 2026.

Luhuan Wu, Brian Trippe, Christian Naesseth, David Blei, and John P Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. *Advances in Neural Information Processing Systems*, 36:31372–31403, 2023.

A Reward-tilted Feynman–Kac PDE

For the interpolation $p_t(x) \propto q_t(x)\exp(r(x,t))$ defined in Section 2.2, with the normalizing constant $Z_t = \int_{\mathbb{R}^d} q_t(x)\exp(r(x,t))\,\mathrm{d}x$, we have

$$\log p_t(x) = \log q_t(x) + r(x,t) - \log Z_t. \tag{20}$$

Differentiating with respect to time gives

$$\begin{aligned}\frac{\partial_t p_t(x)}{p_t(x)} &= \frac{\partial_t q_t(x)}{q_t(x)} + \partial_t r(x,t) - \partial_t \log Z_t \\ &= -\nabla\cdot v_t(x) - \langle v_t(x), \nabla \log q_t(x)\rangle + \partial_t r(x,t) - \partial_t \log Z_t \\ &= -\nabla\cdot v_t(x) - \langle v_t(x), \nabla \log p_t(x) - \nabla r(x,t)\rangle + \partial_t r(x,t) - \partial_t \log Z_t,\end{aligned} \tag{21}$$

where we use the continuity equation for q_t and substitute $\nabla \log q_t = \nabla \log p_t - \nabla r(x,t)$. This can be rewritten as

$$\partial_t p_t(x) = -\nabla\cdot(v_t(x)p_t(x)) + p_t(x)(\partial_t r(x,t) + \langle v_t(x), \nabla r(x,t)\rangle - \partial_t \log Z_t). \tag{22}$$

Since $\int_{\mathbb{R}^d} p_t(x)\,\mathrm{d}x = 1$ for all t, differentiating both sides with respect to t gives $\int_{\mathbb{R}^d} \partial_t p_t(x)\,\mathrm{d}x = 0$, which yields

$$\partial_t \log Z_t = \mathbb{E}_{p_t}[\partial_t r(x,t) + \langle v_t(x), \nabla r(x,t)\rangle]. \tag{23}$$

Substituting the expression for $\partial_t \log Z_t$ back, and defining $g_t(x) = \partial_t r(x,t) + \langle v_t(x), \nabla r(x,t)\rangle$, we obtain

$$\partial_t p_t(x) = -\nabla\cdot(v_t(x)p_t(x)) + p_t(x)(g_t(x) - \mathbb{E}_{p_t}[g_t]). \tag{24}$$

B Derivation of IPG and IPG-CV drifts

We obtain the minimizer of objective (12) for a general regularization constant $\lambda > 0$ with weighted particles $\{(X_t^i, w_t^i)\}_{i=1}^N$, adapting the proof from Nüsken (2024, Appendix A.1). In both IPG and IPG-CV, we choose a small λ such that the weight updates can be ignored, and setting $w_t^i = 1/N$ recovers equations (13) and (15).

Let $W = \mathrm{diag}(w_t) \in \mathbb{R}^{N\times N}$, and $\langle x, y\rangle_{w_t} = \sum_{i=1}^N w_t^i x_i y_i$ denote the weighted inner product on $\mathbb{R}^N$. We define the sample Stein operator $S_{p_t,N}\colon \mathcal{H}_k^d \to \mathbb{R}^N$ by $(S_{p_t,N}u_t)_i = S_{p_t}u_t(X_t^i)$, the weighted centering operator $\Pi_w = I_N - \mathbf{1}w_t^\top$, and the vector $\mathrm{g}_t \in \mathbb{R}^N$ by $[\mathrm{g}_t]_i = g_t(X_t^i) - \sum_{j=1}^N w_t^j g_t(X_t^j)$. Approximating $\mathbb{E}_{p_t}[\cdot]$ in objective (12) with the weighted particles, we get the empirical version

$$\hat{\mathcal{L}}(u_t) = \|S_{p_t,N}u_t + \mathrm{g}_t\|_{w_t}^2 + \lambda\|u_t\|_{\mathcal{H}_k^d}^2. \tag{25}$$

For IPG-CV, using the optimal $\mu_t = \sum_{i=1}^N w_t^i(S_{p_t}u_t(X_t^i) + g_t(X_t^i))$ yields

$$\hat{\mathcal{L}}(u_t) = \|\Pi_w S_{p_t,N}u_t + \mathrm{g}_t\|_{w_t}^2 + \lambda\|u_t\|_{\mathcal{H}_k^d}^2. \tag{26}$$

Therefore, both the objectives are of the form

$$\hat{\mathcal{L}}(u_t) = \|PS_{p_t,N}u_t + \mathrm{g}_t\|_{w_t}^2 + \lambda\|u_t\|_{\mathcal{H}_k^d}^2, \tag{27}$$

with $P = I_N$ for IPG and $P = \Pi_w$ for IPG-CV.

Following the Tikhonov regression approach of Nüsken (2024, Appendix A.1.2), the minimizer of equation (27) is

$$u_t = (PS_{p_t,N})^*(\lambda I_N + (PS_{p_t,N})(PS_{p_t,N})^*)^{-1}(-\mathrm{g}_t), \tag{28}$$

where the adjoint is taken with respect to $\langle\cdot,\cdot\rangle_{w_t}$ and the inner product of $\mathcal{H}_k^d$. For any $x, y \in \mathbb{R}^N$,

$$\langle x, \Pi_w y\rangle_{w_t} = \sum_{i=1}^N w_t^i x_i y_i - \Big(\sum_{i=1}^N w_t^i x_i\Big)\Big(\sum_{j=1}^N w_t^j y_j\Big) = \langle \Pi_w x, y\rangle_{w_t}, \tag{29}$$

from which it follows that Π_w is self-adjoint, and in both cases $P^* = P$ and $(PS_{p_t,N})^* = S^*_{p_t,N}P$. The adjoint $S^*_{p_t,N}\colon \mathbb{R}^N \to \mathcal{H}^d_k$, satisfying $\langle c, S_{p_t,N}v\rangle_{w_t} = \langle S^*_{p_t,N}c, v\rangle_{\mathcal{H}^d_k}$ for all $v \in \mathcal{H}^d_k$ and $c \in \mathbb{R}^N$, is given by Nüsken (2024, Remark 11)

$$S^*_{p_t,N}c = \sum_{j=1}^N w_t^j c_j [k(\cdot, X_t^j)\nabla \log p_t(X_t^j) + \nabla_{X_t^j} k(\cdot, X_t^j)]. \tag{30}$$

By direct calculation, $(S_{p_t,N}S^*_{p_t,N}c)_i = \sum_{j=1}^N \xi_{ij} w_t^j c_j$, i.e., $S_{p_t,N}S^*_{p_t,N} = \xi W$, and

$$(PS_{p_t,N})(PS_{p_t,N})^* = PS_{p_t,N}S^*_{p_t,N}P = P\xi W P. \tag{31}$$

Using these results, equation (28) can be rewritten as

$$u_t = S^*_{p_t,N}P\phi, \quad (P\xi W P + \lambda I_N)\phi = -\mathrm{g}_t. \tag{32}$$

For IPG, this gives $u_t(x) = \sum_j w_t^j \phi_j [k(x, X_t^j)\nabla \log p_t(X_t^j) + \nabla_{X_t^j} k(x, X_t^j)]$ with $(\xi W + \lambda I_N)\phi = -\mathrm{g}_t$, and for IPG-CV, the same u_t with ϕ replaced by $\Pi_w\phi$, and $(\Pi_w \xi W \Pi_w + \lambda I_N)\phi = -\mathrm{g}_t$. Setting $w_t^i = 1/N$, such that $W = I_N/N$ and $\Pi_w = \Pi$, recovers our equations (13) and (15).

On expanding the Stein operators, the entries in the Gram matrix are given by,

$$\begin{aligned}[\xi]_{ij} = \langle \nabla \log p_t(X_t^i), \nabla_{X_t^j} k(X_t^i, X_t^j)\rangle + \langle \nabla \log p_t(X_t^j), \nabla_{X_t^i} k(X_t^i, X_t^j)\rangle \\ + k(X_t^i, X_t^j)\langle \nabla \log p_t(X_t^i), \nabla \log p_t(X_t^j)\rangle + \nabla_{X_t^i} \cdot \nabla_{X_t^j} k(X_t^i, X_t^j),\end{aligned} \tag{33}$$

and are cheap to compute as they involve the kernel, its gradients, and terms already used in the particle updates.

C Adding diffusion-reweighting

The methods discussed so far are motivated by the different degrees of freedom in the Feynman–Kac PDE (3), such as adding drift-diffusion and drift-reweighting terms, and so it is natural to consider adding diffusion-reweighting terms.

For a symmetric positive semi-definite diffusion tensor $D_t(x) \in \mathbb{R}^{d\times d}$, let us add and subtract $\nabla\cdot\nabla\cdot(D_t(x)p_t(x))$ in equation (3), and write $-\nabla\cdot\nabla\cdot(D_t(x)p_t(x)) = -p_t(x)(\mathcal{A}_{p_t}D_t(x))$ where

$$\begin{aligned}\mathcal{A}_{p_t}D_t(x) = \nabla\cdot\nabla\cdot D_t(x) + 2\langle \nabla\cdot D_t(x), \nabla \log p_t(x)\rangle + \langle D_t(x), \nabla^2 \log p_t(x)\rangle_F \\ + \nabla \log p_t(x)^\top D_t(x) \nabla \log p_t(x),\end{aligned} \tag{34}$$

where $(\nabla\cdot D_t(x))_i = \sum_{j=1}^d \partial_{x_j} D_t(x)_{ij}$, $\nabla\cdot\nabla\cdot D_t(x) = \sum_{i,j=1}^d \partial_{x_i}\partial_{x_j} D_t(x)_{ij}$ and $\langle\cdot,\cdot\rangle_F$ denotes the Frobenius inner product. The modified PDE for p_t is given by

$$\partial_t p_t(x) = -\nabla\cdot(v_t(x)p_t(x)) + \nabla\cdot\nabla\cdot(D_t(x)p_t(x)) + p_t(x)(g_t(x) - \mathcal{A}_{p_t}D_t(x) - \mathbb{E}_{p_t}[g_t]). \tag{35}$$

Following our earlier procedures, p_t can be approximated with a weighted stochastic particle system

$$\mathrm{d}X_t^i = v_t(X_t^i)\,\mathrm{d}t + \sqrt{2D_t(X_t^i)}\,\mathrm{d}W_t, \quad \mathrm{d}\omega_t^i = (g_t(X_t^i) - \mathcal{A}_{p_t}D_t(X_t^i))\,\mathrm{d}t. \tag{36}$$

As before, D_t can be estimated to minimize the variance of the particle weights. However, since $\nabla\cdot\nabla\cdot(D_t(x)p_t(x)) = -\nabla\cdot(u_t(x)p_t(x))$ for $u_t(x) = -D_t(x)\nabla\log p_t(x) - \nabla\cdot D_t(x)$, diffusion-reweighting can be treated as a specific instance of drift-reweighting, and may not offer any additional capabilities.

D Additional details on the GMM experiment

Prior. We consider a 256-dimensional GMM with 10 equally weighted components as the prior $q_1(x)$. The component means are drawn i.i.d. from $U[-1.3, 1.3]^{256}$, and are then centered so that the mixture mean is zero. The isotropic per-component variance is chosen such that the total variance of

the GMM is 1. The intermediate densities q_t correspond to the marginals of the noising OU process at time $s = 1 - t$,

$$\mathrm{d}Z_s = -aZ_s\,\mathrm{d}s + b\,\mathrm{d}W_s, \quad Z_0 \sim q_1, \tag{37}$$

where $a = 3$ and $b = \sqrt{6}$. The generative process is given by the probability flow ODE associated with the time-reversal of the noising SDE, $\mathrm{d}X_t = v_t(X_t)\,\mathrm{d}t$, initialized as $X_0 \sim q_0$, with

$$v_t(x) = ax + \frac{b^2}{2}\nabla \log q_t(x). \tag{38}$$

In this case, $q_t(x)$ is available in closed form for $t \in [0, 1]$.

Reward. We consider a linear-Gaussian observation $y = Ax + \sigma_{\text{obs}}\epsilon \in \mathbb{R}^{128}$ with $\sigma^2_{\text{obs}} = 0.1$, for which the likelihood function is $\mathcal{N}(y; Ax, \sigma^2_{\text{obs}} I)$. The entries in observation matrix $A \in \mathbb{R}^{128\times 256}$ are drawn i.i.d. from $\mathcal{N}(0, 1)$, and then the rows are unit normalized. The posterior GMM $p_1(x)$ corresponds to a reward-tilted prior where the reward $R(x)$ is the log-likelihood, and is known analytically.

All methods propagate particles along time $t \in [0, 1]$, uniformly discretized into 500 steps, with a constant Langevin noise scale $\sigma_t = 3$. For both IPG and IPG-CV, we use $\lambda = 10^{-3}$ and RBF kernel with adaptive bandwidth chosen at each time step as the median of pairwise distances

$$\sigma_k^2 = \text{median}^2/\log N. \tag{39}$$

Metrics. The effective sample size (ESS) is computed using the self-normalized particle weights at final time $t = 1$. The other metrics are reported over 256 unweighted particles obtained after systematic resampling, which has no effect if the particles are equally weighted at $t = 1$. The mean error is the L_2 norm of the difference between the empirical sample mean and the exact posterior mean. MMD^2 is computed using the unbiased estimator, against $M = 256$ samples drawn from the exact posterior. We use an RBF kernel with the bandwidth

$$\sigma^2_{\text{MMD}} = \text{median}^2/(2\log M), \tag{40}$$

where the median is over pairwise distances of the posterior samples. The sliced 2-Wasserstein distance (SWD) is also reported against the posterior samples with 10^3 random projections.

D.1 Replacing overdamped Langevin with noisy SVGD

The equations (13) and (15) resemble the SVGD drift targeting p_t at each time t, up to coefficients $\{\phi_j\}_{j=1}^N$. Therefore, as mentioned in Section 4.3, we consider replacing the overdamped Langevin terms in the particle updates with a suitably noise perturbed version of SVGD (Gallego & Insua, 2018; Nüsken & Renger, 2023). The resulting modified particle update is given by

$$\begin{aligned}\mathrm{d}X_t^i = \left(v_t(X_t^i) + u_t(X_t^i) + \frac{1}{N}\sum_{j=1}^N [k(X_t^i, X_t^j)\nabla \log p_t(X_t^j) + \nabla_{X_t^j} k(X_t^i, X_t^j)]\right)\mathrm{d}t \\ + \sum_{j=1}^N \left(\sqrt{\frac{2}{N}D_t}\right)_{ij} \mathrm{d}W_t^j,\end{aligned} \tag{41}$$

where $D_t \in \mathbb{R}^{dN\times dN}$ is a block-structured matrix with the (i, j)-th block as $k(X_t^i, X_t^j)I_d$, $(\sqrt{D_t})_{ij}$ denotes the (i, j)-th block of the matrix square root, and $W_t^j \in \mathbb{R}^d$ are independent standard Brownian motions. We use an RBF kernel with unit bandwidth and compare methods with different corrective drifts. The performance metrics computed over 5 randomly generated priors and observations are shown in Table 4.

FKC and DriftLite show improvements, but both are still behind IPG and IPG-CV, which perform as well as they do when using overdamped Langevin. This indicates that the SVGD-like interactions alone are not enough to replace using the IPG corrective drift.

Table 4: Mean and standard deviation of the metrics when overdamped Langevin is replaced with noisy SVGD on the GMM experiment.

Method	Resampling	ESS/N ↑	Mean error ↓	MMD ↓	SWD ↓
FKC	None	0.006 ± 0.002	7.593 ± 1.337	0.876 ± 0.165	0.682 ± 0.112
	Every step	1.000 ± 0.000	0.867 ± 0.070	0.184 ± 0.018	0.173 ± 0.011
DriftLite	None	0.011 ± 0.005	5.973 ± 1.857	0.658 ± 0.188	0.534 ± 0.136
	Every step	1.000 ± 0.000	1.226 ± 0.117	0.124 ± 0.009	0.146 ± 0.007
IPG	None	–	$\mathbf{0.850 \pm 0.054}$	0.013 ± 0.003	0.093 ± 0.002
IPG-CV	None	–	$\mathbf{0.850 \pm 0.047}$	$\mathbf{0.011 \pm 0.002}$	$\mathbf{0.092 \pm 0.002}$

Table 5: Mean and standard deviation of the metrics using neural network based corrective drifts.

Method	Resampling	ESS/N ↑	Mean error ↓	MMD ↓	SWD ↓
MLP	None	0.982 ± 0.003	0.649 ± 0.089	0.022 ± 0.002	0.089 ± 0.004
MLP-CV	None	0.983 ± 0.004	0.697 ± 0.035	0.019 ± 0.004	0.089 ± 0.001
Transformer	None	0.986 ± 0.002	0.675 ± 0.033	0.020 ± 0.002	0.088 ± 0.001
Transformer-CV	None	0.985 ± 0.001	0.687 ± 0.020	0.021 ± 0.003	0.089 ± 0.002

D.2 Neural network based corrective drifts

We briefly explore parameterizing the corrective drift u_t with a neural network. At each time step t, the network weights are warm-started from the previous time step and updated for 50 iterations by minimizing the objective (12) without regularization. We consider approximating $\mathbb{E}_{p_t}[g_t]$ in the objective both with and without the Stein control variate. Evaluating the objective requires computing $S_{p_t} u_t(x)$ involving $\nabla \cdot u_t(x)$, which we approximate using Hutchinson's trace estimator, where the Jacobian-vector product (JVP) is obtained with `torch.func.jvp`.

We consider an MLP with 2 hidden layers of size 512 and GELU activations. Additionally, to obtain an interacting drift, we consider a small transformer with no positional encoding that treats each particle position as a token. The network consists of 2 pre-norm encoder layers with 8 attention heads, a feedforward width of size 4×128, GELU activations and a linear output layer. Similar to DriftLite, as the reweighting term may not be driven close to zero here, these are run as weighted particle systems, but without resampling. We denote the drifts trained with the control variate in the objective using the suffix "-CV", and report the metrics in Table 5.

The neural network based corrective drifts achieve marginally lower mean error and SWD than IPG and IPG-CV, but higher MMD. Moreover, computing the divergence $\nabla \cdot u_t(x)$ and optimizing the objective with inner iterations at each time step significantly increases the computational cost without a corresponding large gain in performance.

E Additional details on image inpainting

Prior and Reward. We use the 1-rectified flow model of Liu et al. (2023), trained on high-resolution (256×256) AFHQ Cat (Choi et al., 2020) images, as our prior $q_1(x)$. For the inpainting task, observations are obtained by either applying a square mask with side length $0.3\times$ the image size at the center of the image, or masking the right half of the image. In both cases, we add Gaussian noise with $\sigma^2_{\text{obs}} = 0.05$.

All methods are run by uniformly discretizing time $t \in [0, 1]$ into 200 steps. For the Langevin noise schedule, we use $\sigma_t = 1 - t$, since the score $\nabla \log q_t(x)$, and in turn $\nabla \log p_t(x)$, blows up as t approaches 1. In DriftLite, the divergences required to evaluate $S_{p_t} u_t(x)$ are approximated with Hutchinson's trace estimator using the central difference method

$$\nabla \cdot f(x) \approx \frac{1}{2\delta} \langle \epsilon, f(x + \delta\epsilon) - f(x - \delta\epsilon) \rangle, \tag{42}$$

where $\delta = 0.01$ and $\epsilon \sim \mathcal{N}(0, I)$. For IPG and IPG-CV, we use $\lambda = 10^{-1}$ which keeps the linear solve well-conditioned, and verify that ESS/$N \geq 0.999$ if weighted particles are used.

Metrics. PSNR, SSIM and LPIPS are computed against the ground truth image using their standard implementation in TorchMetrics. To measure diversity among the generated samples $\{x^i\}_{i=1}^N$ per test image, we report the sample covariance trace

$$\text{Cov. trace} = \frac{1}{N-1}\sum_{i=1}^{N}\|x^i - \bar{x}\|_2^2, \quad \bar{x} = \frac{1}{N}\sum_{i=1}^{N} x^i, \tag{43}$$

where the per-pixel values are in $[-1, 1]$, and the mean pairwise LPIPS,

$$\text{LPIPS div.} = \frac{2}{N(N-1)}\sum_{i<j}\text{LPIPS}(x^i, x^j). \tag{44}$$

For both measures, values close to zero indicate particle collapse.

We visualize the samples produced by different methods for inpainting in Figures 5 and 6. While IPG and IPG-CV produce diverse samples from the posterior, both FKC and DriftLite suffer from severe particle collapse.

F Additional details on protein backbone structure inference

Prior and Reward. We use Proteina (Geffner et al., 2025), specifically the 60M parameter flow-based model that generates protein backbone structures, as our prior $q_1(x)$. It is trained on a subset of AFDB (Varadi et al., 2022) which includes synthetic structures up to 256 residues in length, and predicts only the α-carbon position in each residue. For the observation, we take K pairwise distances between uniformly sampled distinct pairs of residues, with K as 3% of all available pairs, and add Gaussian noise with $\sigma_{\text{obs}} = 1.0$ Å.

Metrics. To ensure that the generated structures do not deviate substantially from the observation, we measure the RMSD between the pairwise distances in the samples and those in the observation y (RMSD_{obs}). We also measure the RMSD between the generated samples and the ground truth after rotational alignment using the Kabsch algorithm (RMSD_{gt}). Following common practice, the quality of generated protein structures is assessed with the following procedure.

1. For each generated structure, 8 amino-acid sequences are predicted using the inverse folding model ProteinMPNN (Dauparas et al., 2022).
2. Each of these sequences is refolded using ESMFold (Lin et al., 2023), and the RMSD between the refolded structure and the original generated structure is measured.
3. If the lowest such RMSD, defined as the self-consistency RMSD, is below 2 Å, then the structure is considered designable.

We report the fraction of test cases in which a method produces at least one designable structure as "Any designable".

The methods are run by uniformly discretizing time $t \in [0, 1]$ into 500 steps. As in the image inpainting task, we use the Langevin noise schedule $\sigma_t = 1 - t$, and compute the divergences required for DriftLite with Hutchinson's trace estimator using the central difference method. For IPG and IPG-CV, we use $\lambda = 10^{-2}$ which keeps the linear solve well-conditioned and yields $\text{ESS}/N = 0.999$ with weighted particles. We note that Proteina, when used with the generative ODE, produces designable samples at a rate below 20%. Geffner et al. (2025) suggest using a heuristic generative SDE with noise level $\gamma < 1$ to improve designability at the cost of diversity,

$$\mathrm{d}X_t = (v_t(X_t) + \sigma_t \nabla \log q_t(X_t))\,\mathrm{d}t + \sqrt{2\sigma_t\gamma}\,\mathrm{d}W_t. \tag{45}$$

However, for $\gamma \neq 1$, the marginal densities of this SDE are not $q_t(x)$, and the score in the Langevin term within our framework can no longer be estimated from $v_t(x)$. While equation (45) can be used for the particle updates by adjusting the reweighting term accordingly, we leave this for future work and do not use it in our experiments for simplicity.

Table 6: Target proteins used for the structure inference task. In some cases, the terminal residues are not resolved, resulting in a lower modeled residue count than the deposited count.

	Residue count		
PDB ID	Modeled	Deposited	Resolution (Å)
12FG	166	171	1.43
28HS	138	142	1.10
28RZ	186	186	2.50
30JI	173	173	1.75
30ZU	223	246	1.05
32HC	190	197	1.40
38LJ	210	216	1.40
9SM9	129	129	1.96
9WGR	134	145	1.53
9WRY	205	211	2.02
9XFS	230	241	2.20
9YKH	165	170	1.02
9YZ6	128	128	2.47
9ZKD	208	208	2.32
9ZU4	103	129	1.05

Test cases. We collect 15 structures from the Protein Data Bank that were deposited after the release date of the model weights, with lengths between 100 and 250 residues. All structures were determined using X-ray diffraction, with the exception of `9ZKD`, which was determined using electron microscopy, and all were deposited at a resolution at or below 2.5 Å. Further details on the individual structures are provided in Table 6.

Samples produced by different methods are visualized in Figures 7 and 8. As Proteina predicts only the α-carbon positions, secondary structures are assigned based on the P-SEA algorithm from the `biotite` package (Kunzmann & Hamacher, 2018). Samples are intentionally not rotation aligned with the ground truth structure to highlight the particle collapse in FKC and DriftLite, while IPG and IPG-CV produce random orientations, as expected. Histograms of pairwise diversity metrics for IPG and IPG-CV on one test case (`30JI`) are shown in Figure 4. The bimodal distribution of RMSD div. indicates that the methods produce some undesignable structures close to reflected versions of the ground truth. This is to be expected when the designability of the Proteina prior is below 20% as discussed above.

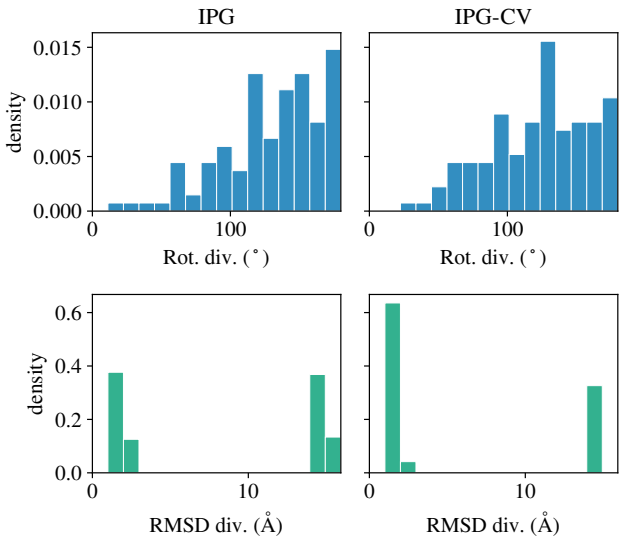

Figure 4: Distribution of pairwise diversity metrics for IPG and IPG-CV on `30JI`. The bimodal RMSD div. indicates that structures close to reflections of the ground truth are occasionally sampled due to the imperfect generative prior.

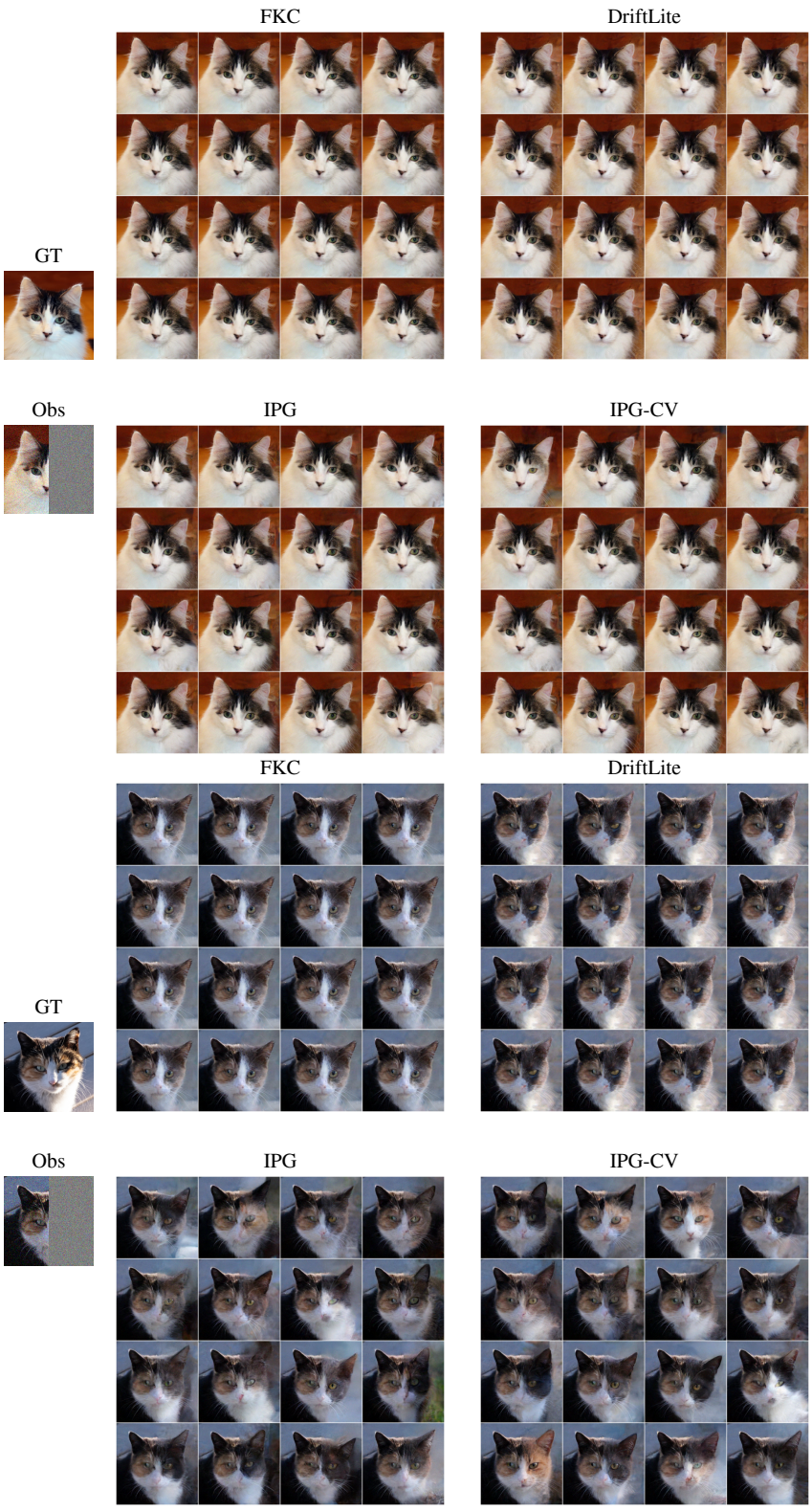

Figure 5: Samples produced by the methods along with the ground truth (GT) and observation (Obs) for the half-mask inpainting task.

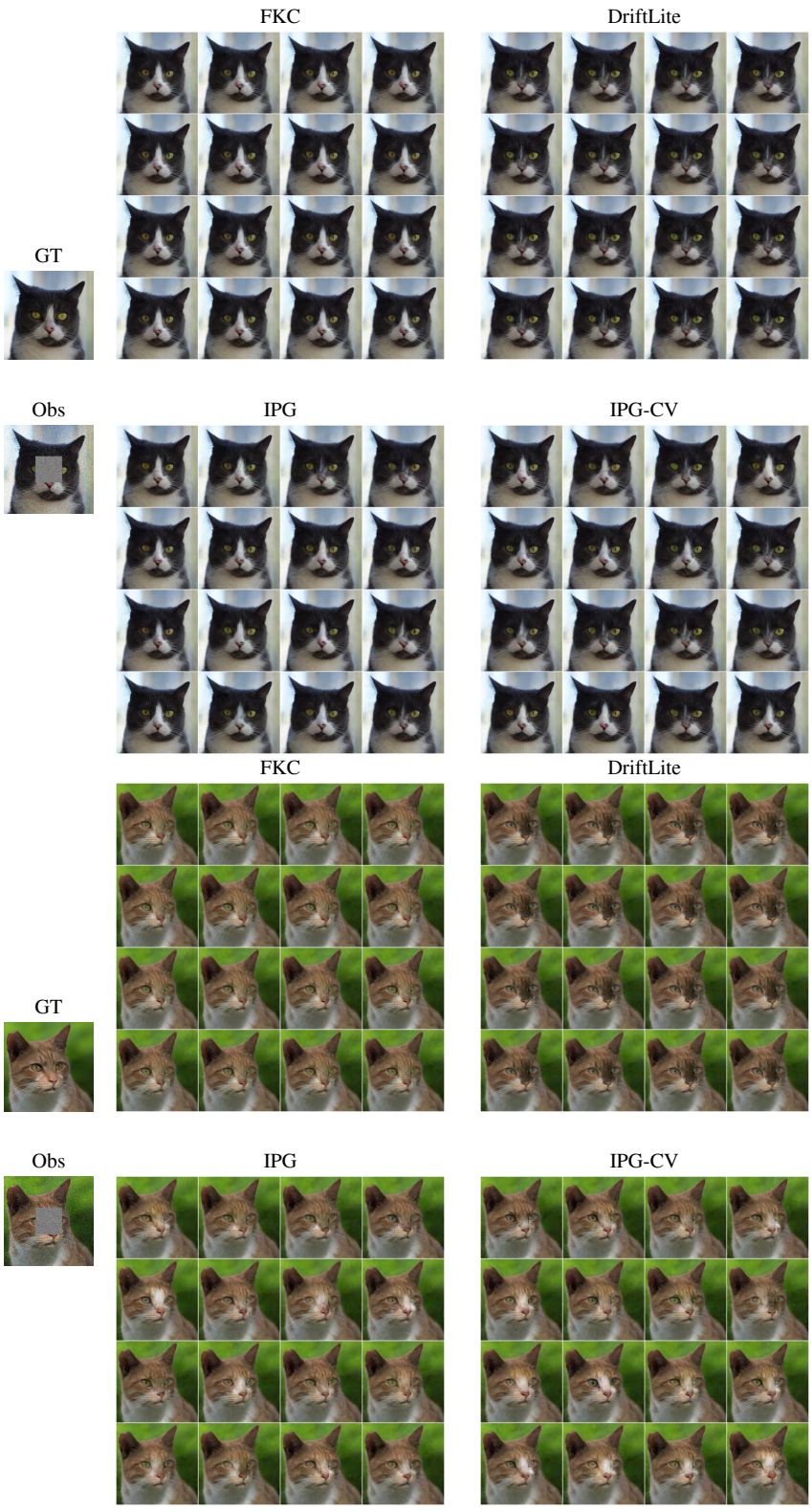

Figure 6: Samples produced by the methods along with the ground truth (GT) and observation (Obs) for the box-mask inpainting task.

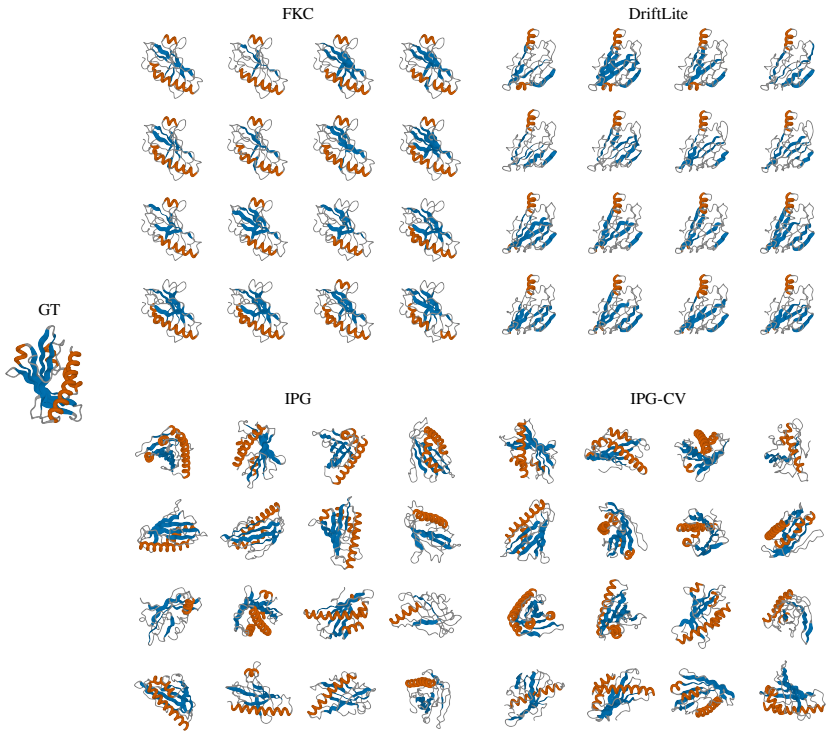

Figure 7: Samples produced from observing random inter-residue distances for protein `3OJI`, visualized without Kabsch alignment to the ground truth (GT).

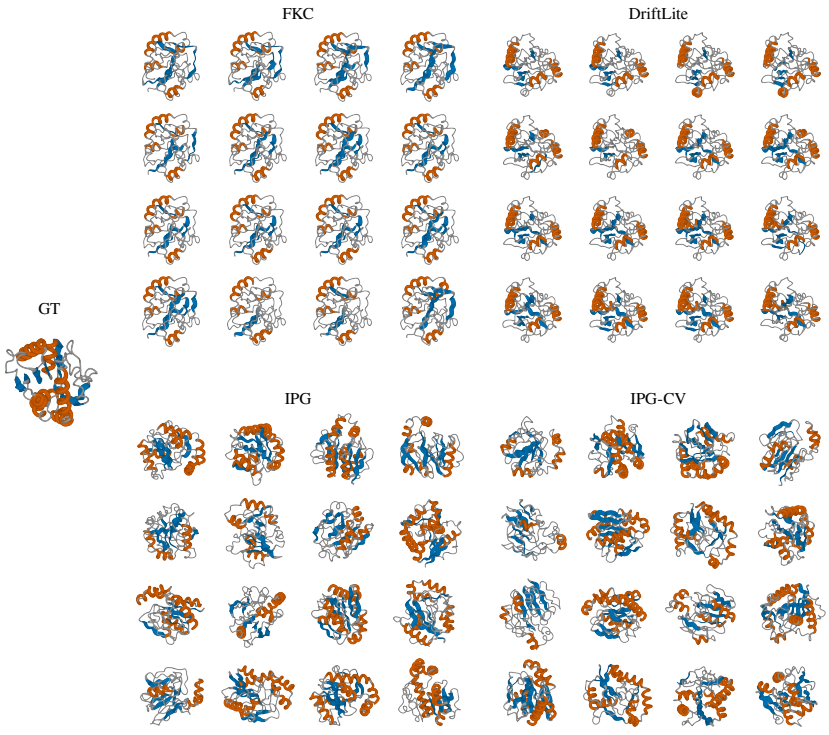

Figure 8: Samples produced from observing random inter-residue distances for protein `9XFS`, visualized without Kabsch alignment to the ground truth (GT).