Feynman-Kac Correctors in Diffusion: Annealing, Guidance, and Product of Experts

Anonymous authors
Paper under double-blind review

Abstract

While score-based generative models are the model of choice across diverse domains, there are limited tools available for controlling inference-time behavior in a principled manner, e.g. for composing multiple pretrained models. Existing classifier-free guidance methods use a simple heuristic to mix conditional and unconditional scores to approximately sample from conditional distributions. However, such methods do not approximate the intermediate distributions, necessitating additional 'corrector' steps. In this work, we provide an efficient and principled method for sampling from a sequence of *annealed*, *geometric-averaged*, or *product* distributions derived from pretrained score-based models. We derive a weighted simulation scheme which we call Feynman-Kac Correctors (FKCs) based on the celebrated Feynman-Kac formula by carefully accounting for terms in the appropriate partial differential equations (PDEs). To simulate these PDEs, we propose Sequential Monte Carlo (SMC) resampling algorithms that leverage inference-time scaling to improve sampling quality. We empirically demonstrate the utility of our methods by proposing amortized sampling via inference-time temperature annealing, improving multi-objective molecule generation using pretrained models, and improving classifier-free guidance for text-to-image generation.

1 Introduction

Score-based generative models, also known as diffusion models, have emerged as the model of choice across diverse generative tasks such as image generation, natural language, and protein simulation (Saharia et al., 2022; Sahoo et al., 2024; Abramson et al., 2024). These models leverage the ability to estimate scores of the sequence of noise-corrupted distributions and then use the learned scores to reverse the corruption process enabling high quality generation. Thus, diffusion models aim to produce new samples from the same distribution as the training data.

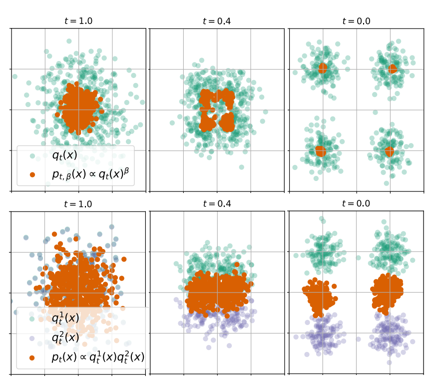

Figure 1: Feynman-Kac Corrector Inference for annealed $p_{t,\beta}(x) \propto q_t(x)^{\beta=10}$ and product $p_t(x) \propto q_t^1(x) q_t^2(x)$ densities.

However, the classical paradigm of generative modeling as the problem of reproducing the training data distribution becomes less relevant for many applications including drug discovery and text-to-image generation. In practice, generative models demonstrate the best performance when tailored to specific needs at inference time. For instance, linear combinations of scores allow for concept composition (Liu et al., 2022) or for increasing image-prompt consistency as in classifier-free guidance (CFG) (Ho & Salimans, 2021). However, by modifying the scores, one loses the control over the marginal distributions of the generated samples. Various approaches from the Monte Carlo sampling literature have been adapted to 'correct' samples along a trajectory to more closely match the prescribed intermediate distributions. Assuming access to an exact score, additional Langevin corrector steps with the desired invariant distribution can be applied with additional simulation steps as the only practical overhead (Song et al., 2021; Bradley & Nakkiran, 2024). However, these corrector schemes are only exact in the limit of infinite intermediate steps. Accept-reject or Sequential Monte Carlo techniques may be used when the score is parameterized through a scalar energy function (Du et al., 2023; Phillips et al., 2024), although these parameterizations require extra computation during training and may sacrifice expressivity in practice (Salimans & Ho, 2021).

While methods for sampling from mixtures or equiprobable regions of diffusion models have been proposed (Skreta et al., 2024), general solutions for accurately sampling from combinations or temperings of flexibly-parameterized diffusion models with limited computational overhead remain elusive.

To address these challenges, we introduce FEYNMAN-KAC CORRECTOR (FKCs), which enable efficient and principled sampling from a sequence of *annealed*, *geometric-averaged*, or *product* distributions derived from pretrained diffusion models. To develop FEYNMAN-KAC CORRECTORS and test their efficacy, we make the following contributions:

- We propose a flexible recipe to construct weighted stochastic differential equations (SDEs), which account for additional terms appearing when manipulating the distribution of generated samples.
- As our primary examples, we derive the correction terms for multiple heuristic schemes commonly used to approximate annealed, product, or geometric averaged distributions, including CFG (Sec. 3).
- To simulate these weighted SDEs, we propose a family of Sequential Monte Carlo (SMC) resampling schemes, which 'correct' a batch of simulated samples to closely approximate the intermediate target distributions (Sec. 4,App. A).
- For the problem of sampling from an unnormalized density, we demonstrate that FKC allows for sampling from a variety temperatures without retraining (Sec. 5.1). Moreover, we demonstrate that a high-temperature learning, low-temperature inference scheme can be more efficient than the notoriously difficult task of directly training a sampler at the lower temperature.
- For pretrained diffusion models, we demonstrate that adding FKC terms enhances compositional generation of molecules with multiple properties (Sec. 5.2) and classifier-free guidance for image generation (Sec. 5.3).

2 BACKGROUND

2.1 DIFFUSION MODELS

Generative modeling via diffusion models can be formulated as the simulation of the Stochastic Differential Equation (SDE) corresponding to the reverse-time process. In particular, during training, one gradually destroys samples from the data-distribution $p_{\text{data}}(x)$ by simulating the following noising SDE:

$$dx_\tau = f_\tau(x_\tau)d\tau + \sigma_\tau d\overline{W}_\tau\,, \quad x_{\tau=0} \sim p_{\text{data}}(x)\,, \tag{1}$$

where $f_\tau(x_\tau)$ is usually some linear drift function $f_\tau(x_\tau) = \alpha_\tau x_\tau$, σ_τ defines the scale of noise through time, and $d\overline{W}_\tau$ is the standard Wiener process. The drift f_τ and the diffusion coefficient σ_τ are chosen so the final density is close to the standard normal distribution $p_{\tau=1} \approx \mathcal{N}(0, I_d)$.

The generation process then can be defined as the family of denoising SDEs in the opposite time direction ($t = 1 - \tau$),

$$dx_t = \big(-f_t(x_t) + \sigma_t^2 \nabla \log p_t(x_t)\big)dt + \sigma_t dW_t\,, \tag{2}$$

where $p_t = p_{1-\tau}$ is the density of the marginals induced by the noising process in Eq. (1); hence, the process starts with $x_0 \sim \mathcal{N}(x \,|\, 0, I_d)$. By training a model of the score functions $\nabla \log p_t(\cdot)$, one can generate new samples from $p_{\text{data}}(x)$ using Eq. (2) (Song et al., 2021).

2.2 FEYNMAN-KAC PDES

While Eq. (2) describes a procedure for simulating individual particles, we can also derive Partial Differential Equations (PDEs) which describe the time-evolution of the density of samples $p_t(x)$ under this SDE. We begin by describing the relevant equations for the standard SDE case.

(1) Continuity Equation, which describes how the density changes when the samples move in space according to a flow or ODE with drift v_t,

$$dx_t = v_t(x_t)dt \implies \frac{\partial p_t^{\text{ode}}(x)}{\partial t} = -\Big\langle \nabla, p_t^{\text{ode}}(x) v_t(x) \Big\rangle. \tag{3}$$

where p_t^{ode} indicates the evolution only according to a flow.

(2) Diffusion Equation, which describes the change of the density for the pure Brownian motion with coefficient σ_t,

$$dx_t = \sigma_t dW_t \implies \frac{\partial p_t^{\text{diff}}(x)}{\partial t} = \frac{\sigma_t^2}{2} \Delta p_t^{\text{diff}}(x)\,. \tag{4}$$

where p_t^{diff} denotes evolution due to the diffusion term only.

The SDE in Eq. (2) can be viewed as the composition of a flow and diffusion terms, where the corresponding Fokker-Planck PDE describes the combined evolution

$$\frac{\partial p_t^{\text{sde}}(x)}{\partial t} = -\left\langle \nabla, p_t^{\text{sde}}(x) v_t(x) \right\rangle + \frac{\sigma_t^2}{2} \Delta p_t^{\text{sde}}(x). \tag{5}$$

However, our main focus in this work will be to study a third type of PDE, which will yield *weighted* SDEs that we eventually use to simulate a sequence of marginals other those the forward noising process $p_{1-\tau}$ (Sec. 3).

(3) Reweighting Equation, which describes the change of density when samples have time-dependent log-weights w_t which are updated based on the positions of samples x_t,

$$dw_t = \bar{g}_t(x_t)dt \quad \Longrightarrow \quad \frac{\partial p_t^w(x)}{\partial t} = \bar{g}_t(x)p_t^w(x)\,, \qquad \text{where} \quad \bar{g}_t(x) = g_t(x) - \int g_t(x)p_t^w(x)dx \tag{6}$$

where the last equation ensures conservation of the normalization constant, $\int dx\; \bar{g}_t(x)p_t^w(x) = 0$.

Feynman-Kac Formula We now focus on the combination of all three components to describe the *Feynman-Kac PDE*,

$$\frac{\partial p_t^{\text{FK}}(x)}{\partial t} = -\left\langle \nabla, p_t^{\text{FK}}(x)v_t(x)\right\rangle + \frac{\sigma_t^2}{2}\Delta p_t^{\text{FK}}(x) + \bar{g}_t(x)p_t^{\text{FK}}(x)\,, \tag{7}$$

where to sample from $p_t^{\text{FK}}(x)$, one first has to sample x_t via the following SDE

$$dx_t = v_t(x_t)dt + \sigma_t dW_t\,,\quad dw_t = \bar{g}_t(x_t)dt\,, \tag{8}$$

and then reweight the obtained samples using w_t. Thus, $p_t^{\text{FK}}(x)$ reflects the density of *weighted* samples, which differs from the density $p_t^{\text{sde}}(x)$ obtained via the Fokker-Planck PDE in Eq. (5) due to the addition of reweighting terms.

In practice, we account for this difference by reweighting a collection of K particles, i.e., for estimating the expectation of test functions ϕ, we account for the weights using

$$\mathbb{E}_{p_T}[\phi(x)] \approx \sum_{k=1}^{K} \frac{\exp(w_T^k)}{\sum_j \exp(w_T^j)}\phi(x_T^k)\,. \tag{9}$$

This expression corresponds to Self-Normalized Importance Sampling (SNIS) estimation, which converges to exact expectation estimators when $K \to \infty$ (e.g. Naesseth et al. (2019)). For justification of the validity of this weighting scheme for Feynman-Kac PDEs, we refer to Lelièvre et al. (2010, Ch. 4). We discuss more refined resampling techniques in App. A.

2.3 Flexibility of Simulation for Given Marginals

Given a PDE describing the time-evolution of a particular density $p_t(x)$, there may exist multiple simulation methods (Song et al., 2021). While it is well-known that the diffusion equation (4) can be simulated using an ODE, $dx_t = -\frac{\sigma_t^2}{2}\nabla \log p_t(x_t)dt$, we emphasize conversions to the reweighting equation below.

Diffusion → Continuity Through simple manipulations, we can rewrite the diffusion equation using a continuity equation and change the simulation scheme accordingly

$$\frac{\partial p_t(x)}{\partial t} = \frac{\sigma_t^2}{2}\Delta p_t(x) = -\left\langle \nabla, p_t(x)\left(-\frac{\sigma_t^2}{2}\nabla \log p_t(x)\right)\right\rangle \implies dx_t = -\frac{\sigma_t^2}{2}\nabla \log p_t(x_t)dt\,. \tag{10}$$

Continuity → Reweighting We first recast the continuity equation in terms of reweighting, in which case the simulation changes the density solely by adjusting the weights of samples (without transport),

$$\frac{\partial p_t(x)}{\partial t} = -\langle \nabla, p_t(x)v_t(x)\rangle = \left(\frac{-1}{p_t(x)}\langle \nabla, p_t(x)v_t(x)\rangle\right)p_t(x) \implies dw_t = (-\langle \nabla, v_t(x_t)\rangle - \langle \nabla \log p_t(x_t), v_t(x_t)\rangle)dt \tag{11}$$

Diffusion → Reweighting We further observe that diffusion terms may be captured in the weights via

$$\frac{\partial p_t(x)}{\partial t} = \frac{\sigma_t^2}{2}\Delta p_t(x) = \frac{\sigma_t^2}{2}p_t(x)\left(\Delta \log p_t(x) + \|\nabla \log p_t(x)\|^2\right) \implies dw_t = \frac{\sigma_t^2}{2}(\Delta \log p_t(x_t) + \|\nabla \log p_t(x_t)\|^2)\,dt \tag{12}$$

In particular, using Eqs. (11) and (12) we now have an approach for translating arbitrary flow v_t or diffusion σ_t terms into the reweighting factors, assuming access to an exact score function $\nabla \log p_t$. Such manipulations will play a key role in deriving our proposed methods in Sec. 3.

Table 1: Conversion rules for different terms of the original Feynman-Kac PDEs (FK-PDEs) and the corresponding weighted SDE (wSDE). For every term corresponding to the original densities q_t (first two columns), we present the terms corresponding to the annealed marginals $p_{t,\beta}(x) \propto q_t(x)^\beta$ (top part) and the terms corresponding to the product of marginals $p_t(x) \propto q_t^1(x)q_t^2(x)$ (bottom part). Importantly, the *correctors are additive* in the weight space, e.g. when transforming the Fokker-Planck equation, we transform both the continuity & diffusion equation terms and sum the corresponding correctors. References to proofs are provided in the right-most column.

Original FK-PDE	Original wSDE	Annealed PDE	Annealed SDE $dx_t =$	FK Corrector dw_t +=	Proof
$-\langle\nabla, q_t v_t\rangle$	$v_t(x_t)dt$	$-\langle\nabla, p_{t,\beta} v_t\rangle$	$v_t(x_t)dt$	$-(\beta-1)\langle\nabla, v_t\rangle dt$	Prop. D.1
		$-\langle\nabla, p_{t,\beta}\beta v_t\rangle$	$\beta v_t(x_t)dt$	$\beta(\beta-1)\langle\nabla\log q_t, v_t\rangle dt$	Prop. D.2
$\frac{\sigma_t^2}{2}\Delta q_t$	$\sigma_t dW_t$	$\frac{\sigma_t^2}{2}\Delta p_{t,\beta}$	$\sigma_t dW_t$	$-\beta(\beta-1)\frac{\sigma_t^2}{2}\|\nabla\log q_t\|^2 dt$	Prop. D.3
		$\frac{\sigma_t^2}{2\beta}\Delta p_{t,\beta}$	$\frac{\sigma_t}{\sqrt{\beta}}dW_t$	$(\beta-1)\frac{\sigma_t^2}{2}\Delta\log q_t dt$	Prop. D.4
$g_t q_t$	$dw_t = g_t dt$	$\beta g_t p_{t,\beta}$	—	$\beta g_t dt$	Prop. D.5
—	—	time-dependent annealing: $\beta \to \beta_t$		$\frac{\partial\beta_t}{\partial t}\log q_t dt$	Prop. D.6
Original FK-PDE	**Original wSDE**	**Product PDE**	**Product SDE $dx_t =$**	**FK Corrector dw_t +=**	
$-\left\langle\nabla, q_t v_t^{1,2}\right\rangle$	$v_t^{1,2}dt$	$-\langle\nabla, p_t(v_t^1+v_t^2)\rangle$	$(v_t^1+v_t^2)dt$	$(\langle\nabla\log q_t^1, v_t^2\rangle + \langle\nabla\log q_t^2, v_t^1\rangle)dt$	Prop. D.7
$\frac{\sigma_t^2}{2}\Delta q_t^{1,2}$	$\sigma_t dW_t$	$\frac{\sigma_t^2}{2}\Delta p_t$	$\sigma_t dW_t$	$-\sigma_t^2\langle\nabla\log q_t^1, \nabla\log q_t^2\rangle dt$	Prop. D.8
$g_t^{1,2}q_t^{1,2}$	$dw_t = g_t^{1,2}dt$	$(g_t^1+g_t^2)p_t$	—	$(g_t^1+g_t^2)dt$	Prop. D.9

3 Modifying Diffusion Inference using Feynman-Kac Correctors

In this section, we propose new sampling tools for combining or modifying diffusion models at inference time using the Feynman-Kac PDEs in Sec. 2.2. To this end, consider several different pretrained diffusion models with marginals $\{q_t^i\}_{i=1}^M$ following

$$\frac{\partial q_t^i}{\partial t} = -\left\langle\nabla, q_t^i\big(-f_t + \sigma_t^2\nabla\log q_t^i\big)\right\rangle + \frac{\sigma_t^2}{2}\Delta q_t^i\,, \tag{13a}$$

$$dx_t = \big(-f_t(x_t) + \sigma_t^2\nabla\log q_t^i(x_t)\big)dt + \sigma_t dW_t\,, \tag{13b}$$

which is the denoising SDE from Eq. (2). Note that q_t^i may arise from training on different datasets or correspond to conditional models with different conditioning. Throughout this work, we assume access to an exact score model $s_t^i(x;\theta^i) = \nabla\log q_t^i(x)$, in part to facilitate the conversion rules introduced in Sec. 2.3 and summarized in Table 1.

At inference time, we would like to sample from a modified target distribution involving these given models. While other variants are possible, we focus on the following examples:

$$p_{t,\beta}^{\text{anneal}}(x) = \frac{1}{Z_t(\beta)}q_t(x)^\beta \quad p_t^{\text{prod}}(x) = \frac{1}{Z_t}q_t^1(x)q_t^2(x) \quad p_{t,\beta}^{\text{geo}}(x) = \frac{1}{Z_t(\beta)}q_t^1(x)^{1-\beta}q_t^2(x)^\beta. \tag{14}$$

A common heuristic for sampling from the distributions in the form of Eq. (14) is to simulate according to the score function corresponding to the target density. For example, in classifier-free guidance (Ho & Salimans, 2021) we use the score of the geometric average $\nabla\log p_{t,\beta}^{\text{geo}} = (1-\beta)\nabla\log q_t^1 + \beta\nabla\log q_t^2$ to simulate the following SDE

$$dx_t = (-f_t(x_t) + \sigma_t^2\nabla\log p_{t,\beta}^{\text{geo}}(x_t))dt + \sigma_t dW_t\,. \tag{15}$$

However, despite the similarity to Eq. (2), this heuristic does not sample from the prescribed marginals including the final distributions, except in special cases. We proceed by using the $p_{t,\beta}^{\text{geo}}$ example to illustrate our approach.

3.1 Outline of Our Approach

To remedy this, we inspect the PDE corresponding to $p_{t,\beta}^{\text{geo}}$, which can be written in terms of the evolution of q_t^1 and q_t^2

$$\frac{\partial p_{t,\beta}^{\text{geo}}(x)}{\partial t} = \frac{\partial}{\partial t}\frac{1}{Z_t(\beta)}\,q_t^1(x)^{(1-\beta)}q_t^2(x)^\beta. \tag{16}$$

Expanding and using our expressions for the Fokker-Planck equation of q_t^i in (13), we proceed to locate terms corresponding to simulation of an SDE with the drift $v_t = -f_t(x_t) + \sigma_t^2\nabla\log p_{t,\beta}^{\text{geo}}$. Collecting all remaining terms of PDE (16) into weights $\bar{g}_t$ we obtain the following Feynman-Kac PDE, which can be simulated using the weighted SDE in Eq. (8), along with the resampling schemes described in App. A

$$\frac{\partial p_{t,\beta}^{\text{geo}}}{\partial t} = -\left\langle\nabla, p_{t,\beta}^{\text{geo}}\, v_t\right\rangle + \frac{\sigma_t^2}{2}\Delta p_{t,\beta}^{\text{geo}} + p_{t,\beta}^{\text{geo}}\,\bar{g}_t\,. \tag{17}$$

Conversion Rules To facilitate constructing the Feynman-Kac PDEs corresponding to existing simulation schemes, in Table 1 we present the conversion rules that describe how the corresponding PDEs change for the annealed densities and the product of densities. We use these rules as building blocks when deriving our practical schemes.

3.2 Classifier-Free Guidance (CFG)

CFG (Ho & Salimans, 2021) is a widely-used procedure that simulates an SDE combining the scores of conditional and unconditional models with a guidance weight β,

$$\nabla \log p_{t,\beta}(x) = (1-\beta)\nabla \log q_t^1(x \mid \emptyset) + \beta \nabla \log q_t^2(x \mid c)$$

In practice, $q_t^1(x|\emptyset)$ may represent an unconditional model (or a model with an empty prompt) whereas $q_t^2(x|c)$ is conditioned on a text prompt, class, or other random variables (Ho & Salimans, 2021). Alternatively, in autoguidance techniques, q_t^1 may be an undertrained version of a stronger conditional or unconditional model q_t^2 (Karras et al., 2024).

For our purposes, we will view CFG as it is usually presented — an attempt to sample from the geometric average distributions $p_{t,\beta}^{\text{geo}}(x) \propto q_t^1(x)^{1-\beta} q_t^2(x)^\beta$. Using the conversion rules in Table 1, we derive the reweighting terms which facilitate consistent sampling along the trajectory.

Proposition 3.1 (Classifier-Free Guidance + FKC). *Consider two diffusion models $q_t^1(x), q_t^2(x)$ defined via (13). The weighted SDE corresponding to the geometric average of the marginals $p_{t,\beta}^{geo}(x) \propto q_t^1(x)^{1-\beta} q_t^2(x)^\beta$ is*

$$dx_t = -f_t(x_t)dt + \sigma_t^2((1-\beta)\nabla \log q_t^1(x_t) + \beta \nabla \log q_t^2(x_t))dt + \sigma_t dW_t\,, \tag{18}$$

$$dw_t = \frac{\sigma_t^2}{2}\beta(\beta-1)\left\|\nabla \log q_t^1(x_t) - \nabla \log q_t^2(x_t)\right\|^2 dt\,.$$

See proof in Prop. E.3. As a further example, we combine CFG with a product of experts in Prop. E.4.

3.3 Annealed Distribution

Next, we consider a single diffusion model with the learned score $\nabla \log q_t(x)$, which we use to sample from the *annealed* or *tempered* density

$$p_{t,\beta}^{\text{anneal}}(x) = q_t(x)^\beta / Z_t(\beta)\,. \tag{19}$$

For $\beta > 1$, this can be used to generate samples from modes or high-probability regions of given models (Karczewski et al., 2024), while in Sec. 5.1 we explore the use of annealed inference in learning diffusion samplers from Boltzmann densities. The annealed target can be shown to admit the following Feynman-Kac weighted simulation scheme.

Proposition 3.2 (Annealed SDE + FKC). *Consider a diffusion model $q_t(x)$ defined via (13). Sampling from the annealed marginals $p_{t,\beta}^{anneal}(x) \propto q_t(x)^\beta$, $\beta > 0$ can be performed by simulating the following weighted SDE*

$$dx_t = (-f_t(x_t) + \eta\sigma_t^2 \nabla \log q_t(x_t))dt + \zeta\sigma_t dW_t\,,$$

$$dw_t = (\beta-1)\langle \nabla, f_t(x_t)\rangle dt + \frac{\sigma_t^2}{2}\beta \|\nabla \log q_t(x_t)\|^2 dt\,,$$

with the coefficients (for $a \in [0, 1/2]$)

$$\eta = \beta + (1-\beta)a\,, \quad \zeta = \sqrt{(\beta + (1-\beta)2a)/\beta}\,. \tag{20}$$

See Prop. E.1 for proof, and note that linear drifts $f_t(x)$ will lead to constant divergence terms which cancel upon reweighting in (9). We detail two choices of a.

Target Score Simulation For $a = 0$, we have $\eta = \beta$ and $\zeta = 1$, which yields the *target score* SDE whose drift corresponds to the score of the annealed target,

$$dx_t = (-f_t(x_t) + \beta\sigma_t^2 \nabla \log q_t(x_t))dt + \sigma_t dW_t\,. \tag{21}$$

Tempered Noise Simulation For $a = 1/2$, we have $\eta = (1+\beta)/2, \zeta = 1/\sqrt{\beta}$). We refer to this as an SDE with *tempered noise*, namely

$$dx_t = (-f_t(x_t) + \frac{\beta+1}{2}\sigma_t^2 \nabla \log q_t(x_t))dt + \frac{\sigma_t}{\sqrt{\beta}} dW_t\,. \tag{22}$$

We focus on these two choices of a, but note that for different β, we found that either target score or tempered-noise simulation could perform better in practice (Sec. 5).

3.4 Product of Experts (PoE)

Intuitively, samples from the product of densities correspond to the generations that have high likelihood values under *both* models. The product can also be interpreted as unanimous vote of experts, since a sample is not accepted if one of the densities is zero. Formally, consider the density

$$p_t^{\text{prod}}(x) = q_t^1(x)q_t^2(x)/Z_t\,. \tag{23}$$

For conditional generative models, the product of densities can describe samples satisfying several conditions. For example, in image generation, we could use $q(x\,|\,\text{“}\texttt{horse}\text{”})q(x\,|\,\text{“}\texttt{a sandy beach}\text{”})$ to generate images of “a horse on a sandy beach” (Du et al., 2023). In Sec. 5.2, we demonstrate that the PoE target can be used to improve molecule generations which multiple conditions.

Again, a natural heuristic is to use the score of the target product density in the reverse-time SDE (2),

$$\nabla \log p_t^{\text{prod}}(x) = \nabla \log q_t^1(x_t) + \nabla \log q_t^2(x_t)\,, \tag{24}$$

In the following proposition, we further combine these rules with the annealing procedure to present the weighted SDE that samples from the marginals $p_{t,\beta}^{\text{prod}}(x) \propto (q_t^1(x)q_t^2(x))^\beta$.

Proposition 3.3 (Product of Experts + FKC). *Consider two diffusion models $q_t^1(x), q_t^2(x)$ defined via (13). The weighted SDE corresponding to the product of the marginals $p_{t,\beta}^{\text{prod}}(x) \propto (q_t^1(x)q_t^2(x))^\beta$, with $\beta > 0$ is*

$$dx_t = \; -f_t(x_t)dt + \sigma_t^2\eta\big(\nabla \log q_t^1(x_t) + \nabla \log q_t^2(x_t)\big)dt + \zeta\sigma_t dW_t\,, \tag{25}$$

$$dw_t = \beta(\beta-1)\frac{\sigma_t^2}{2}\big\|\nabla \log q_t^1(x_t) + \nabla \log q_t^2(x_t)\big\|^2 dt + \beta\sigma_t^2\langle\nabla \log q_t^1(x_t), \nabla \log q_t^2(x_t)\rangle dt + (2\beta-1)\langle\nabla, f_t(x_t)\rangle dt$$

with the coefficients (for $a \in [0, 1/2]$)

$$\eta = \beta + (1-\beta)a\,, \;\; \zeta = \sqrt{(\beta + (1-\beta)2a)/\beta}\,. \tag{26}$$

See proof in Prop. E.2. Again, note that for linear drifts, the divergence term $\langle\nabla, f_t(x)\rangle$ is constant and can be ignored. Similarly to Eqs. (21) and (22) for annealing, we have the *target score* SDE ($a = 0, \eta = \beta, \zeta = 1$) and the *tempered noise* SDE ($a = 1/2, \eta = (\beta+1)/2, \zeta = 1/\sqrt{\beta}$).

4 Resampling Methods

In this section, we describe several options for utilizing the weights to improve sampling with a batch of K particles. While the simplest technique would be to simulate the weighted SDE in Eq. (8) for K independent particles across the full time interval $t \in [0, 1]$ and reweight using SNIS in (9), we expect these full-trajectory weights to have high variance in practice due to error accumulation.

Sequential Monte Carlo Since our weights provide a proper weighting scheme for all intermediate distributions (Naesseth et al., 2019), we can leverage SMC techniques which reweight particles along our trajectories. We find resampling only over an ‘active interval’ $t \in [t_{\text{min}}, t_{\text{max}}]$ useful for improving sample quality and preserving diversity, and set weights to zero outside of this interval. Within the active interval, we resample at each step based on the increment $w_t^{(k)} = g_t(x_t^{(k)})dt$, using systematic sampling proportional to $\exp\{w_t^{(k)}\}$ (Douc & Cappé, 2005). For small discretizations dt, we expect relatively low-variance weights. From this perspective, systematic resampling is an attractive selection mechanism as all particles are preserved in the case of uniform weights.

5 Empirical Study

Throughout this section, we compare our Feynman-Kac corrector (FKC) resampling schemes against their corresponding SDEs without resampling. We consider both target score and tempered noise SDEs. We describe the various resampling schemes in App. A and compare them on the GMM task in App. F.2 Table 6. For the remainder of our experiments, we proceed with systematic resampling.

5.1 Samplers from the Boltzmann Density

As described in Sec. 1, our FKC inference techniques suggest flexible schemes for learning diffusion samplers at a given temperature and sampling according to a different temperature. Since we are given an energy function in this setting, we are not restricted to learning with temperature 1 for our base model q_t. Thus, we use (T_L, T_S) to refer to the learning (q_t) and sampling target ($p_{t,\beta}$) distributions, with $\beta = T_S/T_L$ in the notation of Sec. 3.3.

Mixture of 40 Gaussians with Ground-Truth q_t^β To verify our tools in a tractable setting, we consider a highly multimodal distribution where we can calculate the optimal q_t and $\nabla \log q_t$ for (small) integer β. We show qualitative results in Fig. 2. We find that target score + FKC performs best, while tempered noise has a tendency to drop modes. We also find that FKC outperforms SDE-only simulation in both tempered noise and target score settings. This is further supported by quantitative results in Table 6.

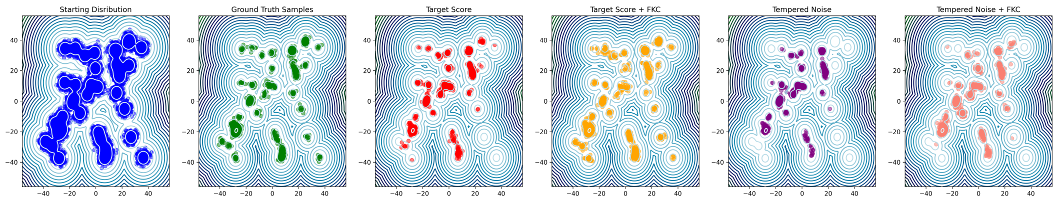

Figure 2: Samples from Mixture of 40 Gaussians.

Table 2: LJ-13 sampling task with various SDEs, with performance measured by mean $\pm$ standard deviation over 3 seeds. The starting temperature is $T_L = 2$, annealed to target temperatures $T_S = 0.8$ and $T_S = 1.5$. The DEM samples are generated with a model trained at those corresponding target temperatures.

Target Temp.	SDE Type	FKC	Distance-$\mathcal{W}_2$	Energy-$\mathcal{W}_1$	Energy-$\mathcal{W}_2$
0.8 ($\beta = 2.5$)	Target Score	✗	**0.912 ± 0.016**	14.521 ± 0.085	14.602 ± 0.076
		✓	0.928 ± 0.009	**5.513 ± 0.586**	**5.591 ± 0.563**
	Tempered Noise	✗	0.924 ± 0.001	6.206 ± 0.007	6.272 ± 0.017
		✓	0.930 ± 0.020	6.438 ± 0.994	6.620 ± 0.998
	DEM	—	0.010 ± 0.001	9.910 ± 0.004	9.921 ± 0.004
1.5 ($\beta = 1.33$)	Target Score	✗	0.222 ± 0.011	5.152 ± 0.040	5.211 ± 0.049
		✓	0.225 ± 0.009	3.249 ± 0.003	3.269 ± 0.004
	Tempered Noise	✗	**0.215 ± 0.004**	2.075 ± 0.010	2.236 ± 0.005
		✓	0.217 ± 0.009	**0.703 ± 0.017**	**0.888 ± 0.048**
	DEM	—	0.074 ± 0.001	4.461 ± 0.024	5.144 ± 0.042

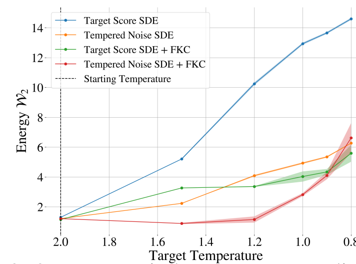

Figure 3: 2-Wasserstein between energy distributions of MCMC samples from the annealed target distribution and our methods at different temperatures. Note the training temperature $T_L = 2$.

Sampling LJ-13 To demonstrate the utility of first learning a sampler at a high temperature then annealing to a lower temperature vs. directly learning at a lower temperature, we consider a Lennard-Jones (LJ) system of 13 particles at a base temperature $T_L = 2$. We train a Denoising Energy Matching (DEM) model (Akhound-Sadegh et al., 2024) at $T_L = 2$ and perform temperature-annealed inference to lower temperatures. In Table 2 and 7 we compare the performance of a DEM model trained at a lower temperature against a DEM model trained at a higher temperature and annealed to the lower temperature using various SDEs. We evaluate methods using the 2-Wasserstein metric between distance distributions, and the 1- and 2-Wasserstein metrics between energy histograms to a reference distribution (App. F.3). We find that tempered noise+FKC performs best at higher target temperatures. However, at lower temperatures, the target score SDE+FKC performs best. Both methods outperform DEM directly trained at the lower temperature. We find DEM is qualitatively easier to learn at higher temperatures requiring much less tuning compared to lower temperatures (Fig. 5). This makes the train-then-anneal approach attractive in this setting.

We find that FKC in this setting is able to successfully sample from temperatures $T_S \in [2.0, 0.8]$ (Fig. 3). This is attractive as, with FKC, practitioners can train a single amortized model, then sample at a variety of temperatures post-hoc. For extended results and discussion see App. F.

5.2 Multi-Property Molecule Generation

We apply FKC to the setting of multi-property molecule generation, which requires molecules to satisfy multiple constraints simultaneously. Here, we look at the setting of dual-target drug design, where a molecule needs to interact with two proteins simultaneously. Dual-target drug design has become increasingly investigated for targeting complex disease pathways (Zhou et al., 2024).

We use our PoE scheme introduced in Prop. 3.3 to take the product of two single property distributions. We select LDMol (Chang & Ye, 2024) to generate molecules, which is a latent diffusion model conditioned on natural language descriptions of molecule properties; this gives flexibility of generating molecules with a wide range of properties. To generate molecules that inhibit a specific protein, we prompt the model with "This molecule inhibits {protein_name}", following Wang et al. (2024).

First, we consider three proteins oracles from TDC (Huang et al., 2021): `JNK3`, `GSK3`β, `DRD2`. Our goal is to generate molecules that are simultaneously predicted to inhibit each pair of proteins. We apply PoE using both target score and tempered noise SDEs at various β; we showcase our best results in Table 3 and the full ablation in Table 8. We primarily evaluate the generated molecules on their predicted ability to bind to two proteins $\mathtt{P}_1$ and $\mathtt{P}_2$, taken as the product of individual predictions. We also look at the number of valid and unique molecules generated, their diversity, and the drug-like quality of the molecules (Lee et al., 2025). For more details on the metrics, see App. F.4. As a baseline, we consider the target score SDE with $\beta = 0.5$, which corresponds to a simple averaging of scores (Liu et al., 2022). We find that the tempered noise SDE at higher β generates molecules that have higher fitness for binding to each pair of proteins. When we incorporate FKC, the average performance of the molecules further increases. Details of our experimental procedure are listed in App. F.4. We also note that PoE+FKC tends to generate more molecules that are unique, valid and have drug-like qualities, although their diversity decreases slightly, which is a common tradeoff.

Table 3: Multi-property molecule generation results. For a set of two target properties ($\mathtt{P}_1$ and $\mathtt{P}_2$), we take the set of the top-10 best performing molecules from a batch-size of 512 as the molecules with the highest $\mathtt{P}_1 * \mathtt{P}_2$ scores. We report averages of the top-10 molecules from 5 runs and the top-1 molecule overall. We also report the diversity, validity & uniqueness, and quality of all molecules.

$\mathtt{P}_1$ / $\mathtt{P}_2$	SDE Type	β	FKC	$\mathtt{P}_1$ top-10 (↑)	$\mathtt{P}_2$ top-10 (↑)	($\mathtt{P}_1$, $\mathtt{P}_2$) top-1 (↑)	Div. (↑)	Val. & Uniq. (↑)	Qual. (↑)
JNK3 GSK3β	Target Score	0.5	✗	$0.212_{\pm 0.016}$	$0.356_{\pm 0.046}$	$(0.500, 0.580)$	$\mathbf{0.910_{\pm 0.000}}$	$0.713_{\pm 0.027}$	$0.127_{\pm 0.015}$
	Tempered Noise	1.5	✗	$0.341_{\pm 0.039}$	$0.468_{\pm 0.041}$	$(0.590, 0.560)$	$0.881_{\pm 0.002}$	$0.813_{\pm 0.025}$	$0.352_{\pm 0.012}$
			✓	$\mathbf{0.342_{\pm 0.012}}$	$\mathbf{0.502_{\pm 0.034}}$	$(\mathbf{0.500}, \mathbf{0.720})$	$0.882_{\pm 0.002}$	$\mathbf{0.832_{\pm 0.021}}$	$\mathbf{0.360_{\pm 0.021}}$
JNK3 DRD2	Target Score	0.5	✗	$0.090_{\pm 0.018}$	$0.434_{\pm 0.065}$	$(0.150, 0.472)$	$\mathbf{0.915_{\pm 0.001}}$	$\mathbf{0.671_{\pm 0.022}}$	$0.228_{\pm 0.011}$
	Tempered Noise	1.5	✗	$0.132_{\pm 0.032}$	$0.550_{\pm 0.036}$	$(0.280, 0.469)$	$0.884_{\pm 0.001}$	$0.650_{\pm 0.021}$	$\mathbf{0.258_{\pm 0.020}}$
			✓	$\mathbf{0.141_{\pm 0.020}}$	$\mathbf{0.617_{\pm 0.040}}$	$(\mathbf{0.360}, \mathbf{0.655})$	$0.884_{\pm 0.005}$	$0.661_{\pm 0.018}$	$0.252_{\pm 0.014}$
GSK3β DRD2	Target Score	0.5	✗	$0.146_{\pm 0.034}$	$0.528_{\pm 0.077}$	$(0.051, 0.908)$	$\mathbf{0.914_{\pm 0.001}}$	$0.709_{\pm 0.021}$	$0.203_{\pm 0.015}$
	Tempered Noise	1.5	✗	$0.228_{\pm 0.016}$	$0.649_{\pm 0.084}$	$(0.550, 0.655)$	$0.884_{\pm 0.002}$	$\mathbf{0.774_{\pm 0.015}}$	$0.303_{\pm 0.012}$
			✓	$\mathbf{0.266_{\pm 0.061}}$	$\mathbf{0.638_{\pm 0.036}}$	$(\mathbf{0.520}, \mathbf{0.796})$	$0.885_{\pm 0.002}$	$\mathbf{0.774_{\pm 0.017}}$	$\mathbf{0.307_{\pm 0.012}}$

Table 4: Docking scores of 32 generated molecules to $\mathtt{P}_1$=ATP1A1 and $\mathtt{P}_2$=CPT2. We used the tempered noise SDE with $\beta = 1.5$.

FKC	($\mathtt{P}_1$, $\mathtt{P}_2$) top-10 (↓)	($\mathtt{P}_1$, $\mathtt{P}2$) top-1 (↓)	Div. (↑)
✗	$-6.65 \pm 1.05, -7.36 \pm 0.854$	$(-8.87, -8.13)$	**0.921**
✓	$(\mathbf{-7.49 \pm 0.71}, \mathbf{-8.31_{\pm 0.94}})$	$(\mathbf{-8.41}, \mathbf{-9.73})$	0.895

Table 5: Image generation using SDXL with classifier-free guidance (CFG). For all metrics mean values are reported.

β	FKC	CLIP	ImageReward	Human Eval
2.5	✗	33.89	0.25	4.85
7.5	✗	**36.00**	0.74	6.15
2.5	✓	35.87	**0.79**	**6.73**

Finally, we consider a more challenging setting of protein-ligand docking, generating binders for proteins ATP1A1 and CPT2. The protein pockets were obtained from Zhou et al. (2024) and the final generated molecules were docked using AutoDock Vina (Eberhardt et al., 2021). Table 4 shows the docking scores of molecules, and we find that incorporating FKC generates molecules with better scores. We visualize the top molecules in App. F.4.

5.3 Image Generation with Stable Diffusion XL

We apply CFG from Prop. 3.1 and study the effect of FKC on generating images with Stable Diffusion XL (SDXL). For generation, we integrate variance-preserving SDE with 100 steps of the Euler-Maruyama solver. We find that FKC performs the best for the guidance scale $\beta = 2.5$ and compare it to CFG with the same scale and the default scale $\beta = 7.5$. To quantitatively evaluate the generated images, we consider three metrics: CLIP Score (Radford et al., 2021), ImageReward (Xu et al., 2024), and Human Evaluation. CLIP Score measures the cosine similarity between an image embedding and a text prompt embedding. ImageReward evaluates generated images by assigning a score that reflects how closely they align with human preferences, including aesthetic quality and prompt adherence.

Figure 4: Samples: CFG(top), CFG+FKC(ours, bottom)

We report all three metrics in Table 5. Our method outperforms the baseline methods in ImageReward and Human Evaluation while achieving comparable performance in terms of the CLIP score. Examples of generated images and prompts are presented in Fig. 4. Additional examples and comparisons with both baselines are included in App. F.5.

6 Conclusion

In this work, we proposed Feynman-Kac Correctors, an array of tools allowing for a fine control over the sample distributions of diffusion processes. These target distributions may arise in compositional generative modeling (Du & Kaelbling, 2024), where we seek to combine specialist models capturing various chemical properties of molecules or different aspects of a complex prompt. Geometric averaging appears in widely-used CFG techniques while, via annealing we demonstrate that an approach of first learning an amortized sampler at a higher temperature then annealing using FKCs down to a lower temperature opens up a new dimension for the construction of amortized samplers.

Finally, our framework allows for the use of reward models (see Prop. E.5), and for time-dependent annealing schedule β_t (Prop. D.6), where the log-density terms which appear in the resulting weights can be efficiently estimated using techniques from (Skreta et al., 2024).

7 Impact Statement

This goal of this paper is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, pp. 1–3, 2024.

Tara Akhound-Sadegh, Jarrid Rector-Brooks, Joey Bose, Sarthak Mittal, Pablo Lemos, Cheng-Hao Liu, Marcin Sendera, Siamak Ravanbakhsh, Gauthier Gidel, Yoshua Bengio, et al. Iterated denoising energy matching for sampling from Boltzmann densities. In *Forty-first International Conference on Machine Learning*, 2024.

Michael S Albergo and Eric Vanden-Eijnden. Nets: A non-equilibrium transport sampler. *arXiv preprint arXiv:2410.02711*, 2024.

Letizia Angeli. *Interacting particle approximations of Feynman-Kac measures for continuous-time jump processes*. PhD thesis, University of Warwick, 2020.

Letizia Angeli, Stefan Grosskinsky, Adam M Johansen, and Andrea Pizzoferrato. Rare event simulation for stochastic dynamics in continuous time. *Journal of Statistical Physics*, 176(5): 1185–1210, 2019.

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

Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep universal probabilistic programming. *arXiv preprint arXiv:1810.09538*, 2018.

Valentin De Bortoli, Michael Hutchinson, Peter Wirnsberger, and Arnaud Doucet. Target score matching. *arXiv preprint arXiv:2402.08667*, 2024.

Arwen Bradley and Preetum Nakkiran. Classifier-free guidance is a predictor-corrector. *arXiv preprint arXiv:2408.09000*, 2024.

Gabriel V Cardoso, Yazid Janati El Idrissi, Sylvain Le Corff, and Eric Moulines. Monte Carlo guided diffusion for Bayesian linear inverse problems. In *ICLR International Conference on Learning Representations*, 2024.

Jinho Chang and Jong Chul Ye. Ldmol: Text-conditioned molecule diffusion model leveraging chemically informative latent space. *arXiv preprint arXiv:2405.17829*, 2024.

Jannis Chemseddine, Christian Wald, Richard Duong, and Gabriele Steidl. Neural sampling from Boltzmann densities: Fisher-Rao curves in the Wasserstein geometry. *arXiv preprint arXiv:2410.03282*, 2024.

Junhua Chen, Lorenz Richter, Julius Berner, Denis Blessing, Gerhard Neumann, and Anima Anandkumar. Sequential controlled Langevin diffusions. *International Conference on Machine Learning*, 2025.

Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. An interpolating distance between optimal transport and Fisher–Rao metrics. *Foundations of Computational Mathematics*, 18:1–44, 2018.

Gavin Earl Crooks. *Excursions in statistical dynamics*. University of California, Berkeley, 1999.

Mark HA Davis. Piecewise-deterministic Markov processes: A general class of non-diffusion stochastic models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 46(3): 353–376, 1984.

Pierre Del Moral. *Mean field simulation for Monte Carlo integration*. Chapman and Hall, CRC press, 2013.

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

Randal Douc and Olivier Cappé. Comparison of resampling schemes for particle filtering. In *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, pp. 64–69, 2005.

Yilun Du and Leslie Kaelbling. Compositional generative modeling: A single model is not all you need. *arXiv preprint arXiv:2402.01103*, 2024.

Yilun Du, Conor Durkan, Robin Strudel, Joshua B Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. In *International conference on machine learning*, pp. 8489–8510. PMLR, 2023.

Jerome Eberhardt, Diogo Santos-Martins, Andreas F Tillack, and Stefano Forli. Autodock vina 1.2. 0: New docking methods, expanded force field, and python bindings. *Journal of chemical information and modeling*, 61(8):3891–3898, 2021.

Stewart N Ethier and Thomas G Kurtz. *Markov Processes: Characterization and Convergence*. John Wiley & Sons, 2009.

Mingzhou Fan, Ruida Zhou, Chao Tian, and Xiaoning Qian. Path-guided particle-based sampling. *International Conference on Machine Learning*, 2024.

Crispin Gardiner. *Stochastic Methods*, volume 4. 2009.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Matthew D. Hoffman and Andrew Gelman. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. 2011.

Peter Holderrieth, Marton Havasi, Jason Yim, Neta Shaul, Itai Gat, Tommi Jaakkola, Brian Karrer, Ricky TQ Chen, and Yaron Lipman. Generator matching: Generative modeling with arbitrary Markov processes. *arXiv preprint arXiv:2410.20587*, 2024.

Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *Proceedings of Neural Information Processing Systems, NeurIPS Datasets and Benchmarks*, 2021.

Christopher Jarzynski. Equilibrium free-energy differences from nonequilibrium measurements: A master-equation approach. *Physical Review E*, 56(5):5018, 1997.

Rafał Karczewski, Markus Heinonen, and Vikas Garg. Diffusion models as cartoonists! the curious case of high density regions. *arXiv preprint arXiv:2411.01293*, 2024.

Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. *arXiv preprint arXiv:2406.02507*, 2024.

Sunwoo Kim, Minkyu Kim, and Dongmin Park. Alignment without over-optimization: Training-free solution for diffusion models. *arXiv preprint arXiv:2501.05803*, 2025.

Jonas Köhler, Leon Klein, and Frank Noé. Equivariant flows: exact likelihood generative learning for symmetric densities. In *International Conference on Machine Learning*, 2020.

Stanislav Kondratyev, Léonard Monsaingeon, and Dmitry Vorotnikov. A new optimal transport distance on the space of finite Radon measures. *arXiv preprint arXiv:1505.07746*, 2015.

Seul Lee, Karsten Kreis, Srimukh Prasad Veccham, Meng Liu, Danny Reidenbach, Yuxing Peng, Saee Paliwal, Weili Nie, and Arash Vahdat. Genmol: A drug discovery generalist with discrete diffusion. *arXiv preprint arXiv:2501.06158*, 2025.

Tony Lelièvre, Mathias Rousset, and Gabriel Stoltz. *Free Energy Computations: A Mathematical Perspective*. World Scientific, 2010.

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

Matthias Liero, Alexander Mielke, and Giuseppe Savaré. Optimal entropy-transport problems and a new Hellinger–Kantorovich distance between positive measures. *Inventiones mathematicae*, 211 (3):969–1117, 2018.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, pp. 423–439. Springer, 2022.

Yulong Lu, Jianfeng Lu, and James Nolen. Accelerating Langevin sampling with birth-death. *arXiv preprint arXiv:1905.09863*, 2019.

Bálint Máté and François Fleuret. Learning interpolations between Boltzmann densities. *Transactions on Machine Learning Research*, 2023.

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

Laurence Illing Midgley, Vincent Stimper, Gregor NC Simm, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Flow annealed importance sampling bootstrap. *International Conference on Learning Representations (ICLR)*, 2023.

Christian A Naesseth, Fredrik Lindsten, Thomas B Schön, et al. Elements of sequential Monte Carlo. *Foundations and Trends® in Machine Learning*, 12(3):307–392, 2019.

Radford M Neal. Annealed importance sampling. *Statistics and Computing*, 11:125–139, 2001.

RuiKang OuYang, Bo Qiang, and José Miguel Hernández-Lobato. Bnem: A boltzmann sampler based on bootstrapped noised energy matching. *arXiv preprint arXiv:2409.09787*, 2024.

Angus Phillips, Hai-Dang Dau, Michael John Hutchinson, Valentin De Bortoli, George Deligiannidis, and Arnaud Doucet. Particle denoising diffusion sampler. In *Forty-first International Conference on Machine Learning*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Lorenz Richter and Julius Berner. Improved sampling via learned diffusions. In *The Twelfth International Conference on Learning Representations*, 2024.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010.

Mathias Rousset. On the control of an interacting particle estimation of Schrödinger ground states. *SIAM journal on mathematical analysis*, 38(3):824–844, 2006.

Mathias Rousset and Gabriel Stoltz. Equilibrium sampling from nonequilibrium dynamics. *Journal of Statistical Physics*, 123:1251–1272, 2006.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

Subham Sekhar Sahoo, Marianne Arriola, Aaron Gokaslan, Edgar Mariano Marroquin, Alexander M Rush, Yair Schiff, Justin T Chiu, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Tim Salimans and Jonathan Ho. Should EBMs model the energy or the score? In *Energy Based Models Workshop-ICLR 2021*, 2021.

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

Marta Skreta, Lazar Atanackovic, Avishek Joey Bose, Alexander Tong, and Kirill Neklyudov. The superposition of diffusion models using the Itô density estimator. *arXiv preprint arXiv:2412.17762*, 2024.

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.

Yifeng Tian, Nishant Panda, and Yen Ting Lin. Liouville flow importance sampler. *International Conference on Machine Learning*, 2024.

Masatoshi Uehara, Yulai Zhao, Tommaso Biancalani, and Sergey Levine. Understanding reinforcement learning-based fine-tuning of diffusion models: A tutorial and review. *arXiv preprint arXiv:2407.13734*, 2024.

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

Suriyanarayanan Vaikuntanathan and Christopher Jarzynski. Escorted free energy simulations: Improving convergence by reducing dissipation. *Physical Review Letters*, 100(19):190601, 2008.

Suriyanarayanan Vaikuntanathan and Christopher Jarzynski. Escorted free energy simulations. *The Journal of chemical physics*, 134(5), 2011.

Francisco Vargas, Will Sussman Grathwohl, and Arnaud Doucet. Denoising diffusion samplers. In *The Eleventh International Conference on Learning Representations*, 2023.

Francisco Vargas, Shreyas Padhy, Denis Blessing, and Nikolas Nusken. Transport meets variational inference: Controlled Monte Carlo diffusions. In *The Twelfth International Conference on Learning Representations: ICLR 2024*, 2024.

Haorui Wang, Marta Skreta, Cher-Tian Ser, Wenhao Gao, Lingkai Kong, Felix Strieth-Kalthoff, Chenru Duan, Yuchen Zhuang, Yue Yu, Yanqiao Zhu, et al. Efficient evolutionary search over chemical space with large language models. *arXiv preprint arXiv:2406.16976*, 2024.

Dongyeop Woo and Sungsoo Ahn. Iterated energy-based flow matching for sampling from Boltzmann densities. *arXiv preprint arXiv:2408.16249*, 2024.

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, 2024.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36, 2024.

Qinsheng Zhang and Yongxin Chen. Path integral sampler: A stochastic control approach for sampling. In *International Conference on Learning Representations*, 2022.

Xiangxin Zhou, Jiaqi Guan, Yijia Zhang, Xingang Peng, Liang Wang, and Jianzhu Ma. Reprogramming pretrained target-specific diffusion models for dual-target drug design. *arXiv preprint arXiv:2410.20688*, 2024.

A Resampling Methods

In this section, we describe several options for utilizing the weights to improve sampling with a batch of K particles. While the simplest technique would be to simulate the weighted SDE in Eq. (8) for K independent particles across the full time interval $t \in [0, 1]$ and reweight using SNIS in (9), we expect these full-trajectory weights to have high variance in practice due to error accumulation.

Sequential Monte Carlo Since our weights provide a proper weighting scheme for all intermediate distributions (Naesseth et al., 2019), we can leverage SMC techniques which reweight particles along our trajectories. We find resampling only over an 'active interval' $t \in [t_{\min}, t_{\max}]$ useful for improving sample quality and preserving diversity, and set weights to zero outside of this interval.

Within the active interval, we resample at each step based on the increment $w_t^{(k)} = g_t(x_t^{(k)})dt$, using systematic sampling proportional to $\exp\{w_t^{(k)}\}$ (Douc & Cappé, 2005). For small discretizations dt, we expect relatively low-variance weights. From this perspective, systematic resampling is an attractive selection mechanism as all particles are preserved in the case of uniform weights.

Jump Process Interpretation of Reweighting Finally, by reframing the reweighting equation in terms of a Markov jump process (Ethier & Kurtz (2009, Ch. 4.2)), a variety of further simulation algorithms for Feynman-Kac PDEs are possible (Del Moral (2013, Ch. 1.2.2, 5); Rousset & Stoltz (2006); Angeli (2020)).

A Markov jump process is determined by a rate function $\lambda_t(x)$, which governs the frequency of jump events, and a Markov transition kernel $J_t(y|x)$, which is used to sample the next state when a jump occurs. The forward Kolmogorov equation for a jump process is given by

$$\frac{\partial p_t^{\text{jump}}(x)}{\partial t} = \left(\int \lambda_t(y) J_t(x|y) p_t(y) dy \right) - p_t(x)\lambda_t(x)$$

where the terms can intuitively be seen to measure the inflow and outflow of probability, respectively.

Our goal is to find choices of $\lambda_t(x), J_t(y|x)$ such that the evolution of p_t^{jump} matches that of p_t^w in Eq. (6) for a given choice of g_t. As emphasized in Del Moral (2013, Ch. 5); Angeli et al. (2019), there are many possible jump processes which satisfy this property. We present a particular choice here, with proof in App. C.2.

Proposition A.1. *For a given g_t in Eq. (6), define the jump process rate and transition as*

$$\lambda_t(x) = \big(g_t(x) - \mathbb{E}_{p_t}[g_t]\big)^- \tag{27a}$$

$$J_t(y|x) = \frac{\big(g_t(y) - \mathbb{E}_{p_t}[g_t]\big)^+ p_t(y)}{\int \big(g_t(z) - \mathbb{E}_{p_t}[g_t]\big)^+ p_t(z) dz} \tag{27b}$$

where $(u)^- :=$ max$(0, -u)$ and $(u)^+ :=$ max$(0, u)$. Then,

$$\frac{\partial p_t^{\textit{jump}}(x)}{\partial t} = \frac{\partial p_t^w(x)}{\partial t} = p_t(x)\big(g_t(x) - \mathbb{E}_{p_t}[g_t]\big) \tag{28}$$

which matches Eq. (6).

In continuous time and the mean-field limit, this jump process formulation of reweighting corresponds to simulating

$$x_{t+dt} = \begin{cases} x_t & \text{w.p. } \; 1 - \lambda_t(x_t)dt + o(dt) \\ \sim J_t(y|x_t) & \text{w.p. } \; \lambda_t(x_t)dt + o(dt). \end{cases} \tag{29}$$

We expect this process to improve the sample population in efficient fashion (Angeli et al., 2019), since jump events are triggered only in states where $(g_t(x) - \mathbb{E}_{p_t}[g_t])^- \geq 0 \implies g_t(x) \leq \mathbb{E}_{p_t}[g_t]$, and transitions are more likely to jump to states with high excess weight $(g_t(y) - \mathbb{E}_{p_t}[g_t])^+ > 0$.

In practice, we use an empirical approximation $p_t^K(z) = \frac{1}{K}\sum_{k=1}^K \delta_z(x^{(k)})$ to approximate the jump rate $\lambda_t(x)$ and transition $J_t(y|x)$. Instead of simulating Eq. (29) directly, one can also adopt an implementation based on birth-death 'exponential clocks' (BDC, Del Moral (2013, Ch. 5.3-4)).

B Related Work

Sequential Monte Carlo methods have proven useful across a wide range tasks involving diffusion models, including for reward-guided generation (Uehara et al., 2024; 2025; Singhal et al., 2025; Kim et al., 2025), conditional generation (Wu et al., 2024), or inverse problems (Dou & Song, 2024; Cardoso et al., 2024), with recent extensions to discrete diffusion models (Singhal et al., 2025; Li et al., 2024; Uehara et al., 2025).

Within the context of diffusion samplers from Boltzmann densities, Phillips et al. (2024) consider SMC for energy-based score parameterizations. Chen et al. (2025); Albergo & Vanden-Eijnden (2024) consider SMC resampling along trajectories with respect to a prescribed geometric annealing path, where Albergo & Vanden-Eijnden (2024) is presented through the Feynman-Kac perspective. The approaches in (Vargas et al., 2024; Albergo & Vanden-Eijnden, 2024) correspond to the *escorted* Jarynski equality (Vaikuntanathan & Jarzynski, 2008; 2011), where additional transport terms are learned to more closely match the evolution of a given density path (Arbel et al., 2021; Chemseddine et al., 2024; Máté & Fleuret, 2023; Tian et al., 2024; Fan et al., 2024; Maurais & Marzouk, 2024). Indeed, the celebrated Jarzynski equality (Jarzynski, 1997; Crooks, 1999) and its variants admit an elegant proof using the Feynman-Kac formula (Lelièvre et al. (2010, Ch. 4),Vaikuntanathan & Jarzynski (2008)).

Predictor-corrector simulation (Song et al., 2021) performs additional Langevin steps to promote matching the intermediate marginals of p_t of a diffusion model. These schemes can be adapted for annealed or product targets, although Du et al. (2023) found best performance using Metropolis corrections. Finally, Bradley & Nakkiran (2024) interpret standard CFG SDE simulation (18) as a predictor-corrector where the corrector targets a different guidance or geometric mixture weight $\beta' = \frac{1}{2}(\beta + 1)$. Our resampling correctors are instead tailored to the original guidance weight β.

Amortized Sampling Recently, there has been renewed interested in learning amortized samplers, and particularly diffusion-based amortized samplers particularly towards molecular systems. Midgley et al. (2023) explored learning a normalizing flow using an α-divergence trained with samples using annealed importance sampling Neal (2001). Zhang & Chen (2022); Vargas et al. (2023); Richter & Berner (2024); Akhound-Sadegh et al. (2024); Albergo & Vanden-Eijnden (2024); Bortoli et al. (2024) learn diffusion annealed bridges between distributions using various methods.

While we use DEM in this work as it achieves state of the art results for our LJ-13 setting, there are several works that build upon DEM using bootstrapping OuYang et al. (2024) and learning the energy function instead of the score Woo & Ahn (2024). We note that our FKC sampler applies to *any* diffusion based sampler.

(Wasserstein)-Fisher-Rao Gradient Flows The reweighting portion of our Feynman-Kac weighted SDEs corresponds to a non-parametric Fisher-Rao gradient flow of a linear functional $\mathcal{G}[p_t] = \int g_t \, p_t dx$, whereas gradient flows in the Wasserstein Fisher-Rao metric (Kondratyev et al., 2015; Chizat et al., 2018; Liero et al., 2018) have a form similar to our weighted PDEs (Lu et al., 2019) for an appropriate ODE simulation term $v_t = \nabla g_t$. In sampling applications, Chemseddine et al. (2024) study the problem of when a given tangent direction in the Fisher-Rao space can be simulated using transport via a tangent direction in the Wasserstein space.

C Feynman-Kac Processes

C.1 Markov Generators for Feynman-Kac Processes

In Sec. 2, we described the adjoint generators $\mathcal{L}_t^{*(v)}[p_t], \mathcal{L}_t^{*(\sigma)}[p_t], \mathcal{L}_t^{*(g)}[p_t]$ corresponding to flows with vector field v_t, diffusions with coefficient σ_t, and reweighting with respect to g_t. In particular, the Kolmogorov forward equation $\frac{\partial p_t}{\partial t}(x) = \mathcal{L}_t^*[p_t](x)$ corresponds to our PDEs presented in Eqs. (3), (5) and (6). In the lemma below, we recall the generators which are adjoint to those in Sec. 2 and operate over smooth, bounded test functions with compact support, e.g. $\mathcal{L}_t^{(v)}[\phi]$.

Lemma C.1 (Adjoint Generators)**.** *Using the identity* $\int \phi(x)\ \mathcal{L}_t^*[p_t](x)\ dx = \int \mathcal{L}_t[\phi](x)\ p_t(x)\ dx$

$$\textbf{\textit{Flow:}}\quad \mathcal{L}_t^{(v)}[\phi](x) = \langle \nabla\phi(x), v_t(x)\rangle \tag{30}$$

$$\mathcal{L}_t^{*(v)}[p_t](x) = -\langle \nabla, p_t(x)\ v_t(x)\rangle$$

$$\textbf{\textit{Diffusion:}}\quad \mathcal{L}_t^{(\sigma)}[\phi](x) = \frac{\sigma_t^2}{2}\Delta\phi(x) \tag{31}$$

$$\mathcal{L}_t^{*(\sigma)}[p_t](x) = \frac{\sigma_t^2}{2}p_t(x) \tag{32}$$

$$\textbf{\textit{Reweighting:}}\quad \mathcal{L}_t^{(g,p)}[\phi](x) = \phi_t(x)\Big(g_t(x) - \int g_t(x)\ p_t(x)\ dx\Big) \tag{33}$$

$$\mathcal{L}_t^{*(g)}[p_t](x) = p_t(x)\Big(g_t(x) - \int g_t(x)\ p_t(x)dx\Big)$$

Proof. The proofs for flows and diffusions follow using integration by parts, with proofs found in, for example, Holderrieth et al. (2024, Sec. A.5). For the reweighting generator, we have

$$\begin{aligned}
\int \phi(x)\mathcal{L}_t^{*(g)}[p_t](x)dx &= \int \phi(x)\Big(p_t(x)\Big(g_t(x) - \int g_t(y)\ p_t(y)dy\Big)\Big)dx \\
&= \int p_t(x)\Big(\phi(x)\Big(g_t(x) - \int g_t(y)\ p_t(y)dy\Big)\Big)dx \\
&=: \int p_t(x)\ \mathcal{L}_t^{(g,p)}[\phi](x)\ dx
\end{aligned}$$

Note that the weights g_t are often chosen in relation to the unnormalized density of p_t (Lelièvre et al. (2010, Sec. 4)), and our attention will be focused on the pair of generator actions $\mathcal{L}_t^{*(g)}[p_t], \mathcal{L}_t^{(g,p)}[\phi]$ for possibly time-dependent ϕ. □

C.2 Jump Process Interpretation of Reweighting

One way to perform simulation of the reweighting equation will be to rewrite it in terms of a jump process. We first recall the definition of the Markov generator of a jump process (Ethier & Kurtz (2009, 4.2), Del Moral (2013, 1.1), Holderrieth et al. (2024, A.5.3)) and derive its adjoint generator.

Lemma C.2 (Jump Process Generators)**.** *Using the definition of the jump process generator and the identity* $\int \phi(x)\ \mathcal{J}_t^*[p_t](x)\ dx = \int \mathcal{J}_t[\phi](x)\ p_t(x)\ dx$*. Letting* $W_t(x,y) = \lambda_t(x)J_t(y|x)$ *for normalized* $J_t(y|x)$*,*

$$\textbf{\textit{Jump Process:}}\quad \mathcal{J}_t^{(W)}[\phi](x) := \int \Big(\phi(y) - \phi(x)\Big)\lambda_t(x)J_t(y|x)dy \tag{34a}$$

$$\mathcal{J}_t^{*(W)}[p_t](x) = \Big(\int \lambda_t(y)J_t(x|y)p_t(y)dy\Big) - p_t(x)\lambda_t(x) \tag{34b}$$

Proof. Through simple manipulations and changing the variables of integration, we obtain

$$\begin{aligned}
\int \phi(x)\ \mathcal{J}_t^*[p_t](x)\ dx &= \int \mathcal{J}_t[\phi](x)\ p_t(x)\ dx \\
&= \int\Big(\int \Big(\phi(y) - \phi(x)\Big)\lambda_t(x)J_t(y|x)dy\Big)p_t(x)\ dx \\
&= \int\int \phi(y)\lambda_t(x)J_t(y|x)p_t(x)\ dydx - \int\int \phi(x)\lambda_t(x)J_t(y|x)p_t(x)\ dydx \\
&= \int\int \phi(x)\lambda_t(y)J_t(x|y)p_t(y)\ dxdy - \int\int \phi(x)\lambda_t(x)J_t(y|x)p_t(x)\ dydx \\
&= \int \phi(x)\Big(\Big(\int \lambda_t(y)J_t(x|y)p_t(y)dy\Big) - p_t(x)\lambda_t(x)\Big(\int J_t(y|x)dy\Big)\Big)dx
\end{aligned}$$

$$\implies \quad \mathcal{J}_t^*[p_t](x) = \Big(\int \lambda_t(y)J_t(x|y)p_t(y)dy\Big) - p_t(x)\lambda_t(x)$$

using the assumption that $J_t(y|x)$ is normalized. □

Reweighting → Jump Process Our goal is to derive a jump process such that the adjoint generators are equivalent $\mathcal{J}_t^{*(W)}[p_t](x) = \mathcal{L}_t^{*(g)}[p_t](x)$ for a given reweighting generator with weights g_t (Eq. (32)).

While Del Moral (2013); Angeli (2020) emphasize the freedom of choice in such generators,[1] Sec. 4 of (Angeli et al., 2019) argues for a particular choice to reduce the expected number of resampling events. To define this process, consider the following thresholding operations,

$$(u)^- := \max(0, -u) \qquad (u)^+ := \max(0, u), \qquad \text{which satisfy:} \quad (u)^+ - (u)^- = u. \tag{35}$$

We can now define the Markov generator using

$$W_t(x,y) = \lambda_t(x) J_t(y|x) \quad \lambda_t(x) := \Big(g_t(x) - \mathbb{E}_{p_t}[g_t]\Big)^- \quad J_t(y|x) := \frac{(g_t(y) - \mathbb{E}_{p_t}[g_t])^+ p_t(y)}{\int (g_t(z) - \mathbb{E}_{p_t}[g_t])^+ p_t(z) dz} \tag{36}$$

Since jump events are triggered based on $\lambda_t(x_t) = (g_t(x) - \mathbb{E}_{p_t}[g_t])^-$ and are more likely to transition to events with high excess weight $(g_t(y) - \mathbb{E}_{p_t}[g_t])^+ p_t(y)$, we expect this process to improve the sample population in efficient fashion (Angeli et al., 2019).

Proposition C.3. *For a given weighting function g_t and the adjoint generator $\mathcal{L}_t^{*(g)}$, the adjoint generator $\mathcal{J}_t^{*(W)}$ derived using in Eq. (36) satisfies $\mathcal{J}_t^{*(W)}[p_t](x) = \mathcal{L}_t^{*(g)}[p_t](x)$. More explicitly, we have*

$$\mathcal{L}_t^{*(g)}[p_t](x) = \mathcal{J}_t^{*(W)}[p_t](x) \tag{37}$$

$$p_t(x)\Big(g_t(x) - \int g_t(x)\, p_t(x) dx\Big) =$$

$$\left(\int \Big(g_t(y) - \mathbb{E}_{p_t}[g_t]\Big)^- \frac{(g_t(x) - \mathbb{E}_{p_t}[g_t])^+ p_t(x)}{\int (g_t(z) - \mathbb{E}_{p_t}[g_t])^+ p_t(z) dz} p_t(y) dy\right) p_t(x) \Big(g_t(x) - \mathbb{E}_{p_t}[g_t]\Big)^-.$$

Proof. We start by expanding the definition of $\mathcal{J}_t^{*(W)}[p_t](x)$

$$\mathcal{J}_t^{*(W)}[p_t](x) = \left(\int \lambda_t(y) J_t(x|y) p_t(y) dy\right) - p_t(x)\lambda_t(x) \tag{38a}$$

$$= \left(\int \Big(g_t(y) - \mathbb{E}_{p_t}[g_t]\Big)^- \frac{(g_t(x) - \mathbb{E}_{p_t}[g_t])^+ p_t(x)}{\int (g_t(z) - \mathbb{E}_{p_t}[g_t])^+ p_t(z) dz} p_t(y) dy\right)$$
$$- p_t(x)\Big(g_t(x) - \mathbb{E}_{p_t}[g_t]\Big)^- \tag{38b}$$

$$= \left(\int \Big(g_t(y) - \mathbb{E}_{p_t}[g_t]\Big)^- p_t(y) dy\right) \left(\frac{(g_t(x) - \mathbb{E}_{p_t}[g_t])^+ p_t(x)}{\int (g_t(z) - \mathbb{E}_{p_t}[g_t])^+ p_t(z) dz}\right)$$
$$- p_t(x)\Big(g_t(x) - \mathbb{E}_{p_t}[g_t]\Big)^- \tag{38c}$$

$$= \left(\frac{\int (g_t(y) - \mathbb{E}_{p_t}[g_t])^- p_t(y) dy}{\int (g_t(z) - \mathbb{E}_{p_t}[g_t])^+ p_t(z) dz}\right) p_t(x) \Big(g_t(x) - \mathbb{E}_{p_t}[g_t]\Big)^+ - p_t(x)\Big(g_t(x) - \mathbb{E}_{p_t}[g_t]\Big)^- \tag{38d}$$

Using Eq. (35), note that

$$\int \Big(g_t(z) - \mathbb{E}_{p_t}[g_t]\Big)^+ p_t(z) dz - \int dp_t(z) \Big(g_t(z) - \mathbb{E}_{p_t}[g_t]\Big)^- = \int (g_t(z) - \mathbb{E}_{p_t}[g_t]) p_t(z) dz = 0 \tag{39}$$

which implies $\int (g_t(z) - \mathbb{E}_{p_t}[g_t])^+ p_t(z) dz = \int (g_t(z) - \mathbb{E}_{p_t}[g_t])^- p_t(z) dz$. We proceed in two cases, handling separately the trivial case where the denominator in Eq. (38d) is zero.

[1] For example, see Rousset (2006); Rousset & Stoltz (2006) for a particular instantiation combining separate birth and death processes.

Case 1 ($\lambda_t(x) = 0\ \forall z \in supp(p_t)$): Note that $\int \big(g_t(z) - \mathbb{E}_{p_t}[g_t]\big)^{-} p_t(z)dz = 0$ if and only if $g_t(z) = \mathbb{E}_{p_t}[g_t],\ \forall z$, since $(u)^{-} \geq 0$. In this case, the generators become trivial and we can confirm

$$\begin{aligned}
\mathcal{L}_t^{*(g)}[p_t](x) &= p_t(x)\Big(g_t(x) - \int g_t(x)\, p_t(x)dx\Big) = p_t(x)(\mathbb{E}_{p_t}[g_t] - \mathbb{E}_{p_t}[g_t]) = 0 \\
\mathcal{J}_t^{*(W)}[p_t](x) &= \int 0 \cdot 0\, p_t(y)dy - p_t(x) \cdot 0 = 0
\end{aligned} \tag{40}$$

and thus Eq. (37) holds, as desired.

Case 2 ($\exists x \in supp(p_t)$ s.t. $\lambda_t(x) > 0$): Under the assumption, $\exists x \in \text{supp}(\mu_t)$ s.t. $\big(g_t(x) - \mathbb{E}_{p_t}[g_t]\big)^{-} > 0$. This implies $\int \big(g_t(z) - \mathbb{E}_{p_t}[g_t]\big)^{-} p_t(z)dz = \int \big(g_t(z) - \mathbb{E}_{p_t}[g_t]\big)^{+} p_t(z)dz > 0$.

In this case, we can conclude using Eq. (39) that $\frac{\int dp_t(z)\big(g_t(z) - \mathbb{E}_{p_t}[g_t]\big)^{-}}{\int dp_t(z)\big(g_t(z) - \mathbb{E}_{p_t}[g_t]\big)^{+}} = 1$.

Continuing from Eq. (38d)

$$\begin{aligned}
\mathcal{J}_t^{*(W)}[p_t](x) &= \left(\frac{\int (g_t(y) - \mathbb{E}_{p_t}[g_t])^{-} p_t(y)dy}{\int (g_t(z) - \mathbb{E}_{p_t}[g_t])^{+} p_t(z)dz}\right) p_t(x)\Big(g_t(x) - \mathbb{E}_{p_t}[g_t]\Big)^{+} \\
&\quad - p_t(x)\Big(g_t(x) - \mathbb{E}_{p_t}[g_t]\Big)^{-} && \text{(41a)} \\
&= p_t(x)\Big(\Big(g_t(x) - \mathbb{E}_{p_t}[g_t]\Big)^{+} - \Big(g_t(x) - \mathbb{E}_{p_t}[g_t]\Big)^{-}\Big) && \text{(41b)} \\
&= p_t(x)(g_t(x) - \mathbb{E}_{p_t}[g_t]) && \text{(41c)} \\
&= \mathcal{L}_t^{*(g)}[p_t](x) && \text{(41d)}
\end{aligned}$$

as desired. Note that, in the second to last line, we used the identity in Eq. (35) that $(u)^{+} - (u)^{-} = u$. □

C.3 Simulation Schemes

In practice, we use an empirical mean over K particles with as an approximation to the expectation $\mathbb{E}_{p_t}[g_t]$, with

$$\begin{aligned}
\Big(g_t(x^{(k)}) - \mathbb{E}_{p_t}[g_t]\Big)^{-} &\approx \Big(g_t(x^{(k)}) - \frac{1}{K}\sum_{i=1}^{K} g_t(x^{(i)})\Big)^{-}, \\
\Big(g_t(x^{(k)}) - \mathbb{E}_{p_t}[g_t]\Big)^{+} &\approx \Big(g_t(x^{(k)}) - \frac{1}{K}\sum_{i=1}^{K} g_t(x^{(i)})\Big)^{+}
\end{aligned} \tag{42}$$

See Del Moral (2013, Sec. 5.4) for discussion.

Discretization of the Continuous-Time Jump Process To simulate a jump process with generator $\mathcal{J}_t^{(J,p)}[\phi]$, we can consider the following infinitesimal sampling procedure (Gardiner (2009, Ch. 12); Davis (1984); Holderrieth et al. (2024)). With rate $\lambda_t(x) = \Big(g_t(x) - \mathbb{E}_{p_t}[g_t]\Big)^{-}$, the particle jumps to a new configuration,

$$x_{t+dt} = \begin{cases} x_t & \text{with probability } \ 1 - dt \cdot \lambda_t(x_t) + o(dt) \\ y_{t+dt} \sim \dfrac{\Big(g_t(x^{(k)}) - \frac{1}{K}\sum_{i=1}^{K} g_t(x^{(i)})\Big)^{+}}{\sum_{j=1}^{K}\Big(g_t(x^{(j)}) - \frac{1}{K}\sum_{i=1}^{K} g_t(x^{(i)})\Big)^{+}} & \text{with probability } \ dt \cdot \lambda_t(x_t) + o(dt) \end{cases} \tag{43}$$

The new configuration is sampled according to an empirical approximation of $J_t(y|x)$ using $p_t^K(y) = \frac{1}{K}\sum_{k=1}^{K} \delta_y(x^{(k)})$, where the outer $\frac{1}{K}$ factor cancels.

Note that the jump rate is zero for particles with $g_t(x) \geq \mathbb{E}_{p_t}[g_t]$. Resampling a new particle proportional to $(g_t(x^{(k)}) - \frac{1}{K}\sum_j g_t(x^{(j)}))^{+}$ thus promotes the replacement of low importance-weight samples with more promising samples.

Interacting Particle System Following Del Moral (2013, Sec 5.4), the process may also be simulated using 'exponential clocks'. In particular, we sample an exponential random variable with rate 1, $\tau^{(k)} \sim \text{exponential}(1)$ as the time when the next jump event will occur (see Gardiner (2009, Ch. 12)). We record artificial time by accumulating the rate function $\lambda_{t_{\text{last}}:s} = \sum_{t=t_{\text{last}}}^{s} \lambda_t(x_t)dt$ for samples x_t along our simulated diffusion. Upon exceeding the threshold time $\lambda^{(k)}_{t_{\text{last}}:s} \geq \tau^{(k)}$, we sample a transition according the empirical approximaton of $J_t(y|x)$ in Eq. (43). We report results using this scheme in App. F.2 Table 6, but found it to underperform relative to systematic resampling in these initial experiments.

D PROOFS FOR TABLE 1

D.1 ANNEALING

Proposition D.1 (Annealed Continuity Equation)**.** *Consider the marginals generated by the continuity equation*

$$\frac{\partial q_t(x)}{\partial t} = -\big\langle \nabla, q_t(x) v_t(x) \big\rangle \,. \tag{44}$$

The marginals $p_{t,\beta}(x) \propto q_t^{\beta}(x)$ satisfy the following PDE

$$\frac{\partial}{\partial t} p_{t,\beta}(x) = -\big\langle \nabla, p_{t,\beta}(x) v_t(x) \big\rangle + p_{t,\beta}(x)\big[g_t(x) - \mathbb{E}_{p_{t,\beta}} g_t(x)\big]\,, \tag{45}$$

$$g_t(x) = (1-\beta)\big\langle \nabla, v_t(x) \big\rangle \,. \tag{46}$$

Proof. We want to find the partial derivative of the annealed density

$$p_{t,\beta}(x) = \frac{q_t(x)^\beta}{\int dx\; q_t(x)^\beta}\,, \quad \frac{\partial}{\partial t} p_{t,\beta}(x) = ? \tag{47}$$

By the straightforward calculations we have

$$\frac{\partial}{\partial t} \log p_{t,\beta} = \beta \frac{\partial}{\partial t} \log q_t - \int dx\; p_{t,\beta} \beta \frac{\partial}{\partial t} \log q_t \tag{48}$$

$$= -\beta\big\langle \nabla, v_t \big\rangle - \beta\big\langle \nabla \log q_t, v_t \big\rangle - \int dx\; p_{t,\beta}\big[-\beta\big\langle \nabla, v_t\big\rangle - \beta\big\langle \nabla \log q_t, v_t \big\rangle\big] \tag{49}$$

$$\begin{aligned} &= -\big\langle \nabla, v_t \big\rangle - \big\langle \nabla \log p_{t,\beta}, v_t \big\rangle + (1-\beta)\big\langle \nabla, v_t \big\rangle \\ &\quad - \int dx\; p_{t,\beta}\big[-\beta\big\langle \nabla, v_t \big\rangle - \big\langle \nabla \log p_{t,\beta}, v_t \big\rangle\big] \end{aligned} \tag{50}$$

$$= -\big\langle \nabla, v_t \big\rangle - \big\langle \nabla \log p_{t,\beta}, v_t \big\rangle + (1-\beta)\big\langle \nabla, v_t \big\rangle - \int dx\; p_{t,\beta}\big[(1-\beta)\big\langle \nabla, v_t \big\rangle\big]\,. \tag{51}$$

Thus, we have

$$\frac{\partial}{\partial t} p_{t,\beta}(x) = -\big\langle \nabla, p_{t,\beta}(x) v_t(x) \big\rangle + p_{t,\beta}(x)\big[(1-\beta)\big\langle \nabla, v_t(x) \big\rangle - \mathbb{E}_{p_{t,\beta}}(1-\beta)\big\langle \nabla, v_t(x) \big\rangle\big]\,, \tag{52}$$

which can be simulated as

$$dx_t = v_t(x_t)dt\,, \tag{53}$$

$$dw_t = -(\beta - 1)\big\langle \nabla, v_t(x_t) \big\rangle dt\,. \tag{54}$$

$\square$

Proposition D.2 (Scaled Annealed Continuity Equation)**.** *Consider the marginals generated by the continuity equation*

$$\frac{\partial q_t(x)}{\partial t} = -\big\langle \nabla, q_t(x) v_t(x) \big\rangle . \tag{55}$$

The marginals $p_{t,\beta}(x) \propto q_t^\beta(x)$ satisfy the following PDE

$$\frac{\partial}{\partial t} p_{t,\beta}(x) = -\big\langle \nabla, p_{t,\beta}(x)\beta v_t(x)\big\rangle + p_{t,\beta}(x)\big[g_t(x) - \mathbb{E}_{p_{t,\beta}} g_t(x)\big] , \tag{56}$$

$$g_t(x) = -(1-\beta)\big\langle \nabla \log p_{t,\beta}(x), v_t(x)\big\rangle . \tag{57}$$

Proof. We want to find the partial derivative of the annealed density

$$p_{t,\beta}(x) = \frac{q_t(x)^\beta}{\int dx\; q_t(x)^\beta} , \quad \frac{\partial}{\partial t} p_{t,\beta}(x) = ? \tag{58}$$

By the straightforward calculations we have

$$\frac{\partial}{\partial t} \log p_{t,\beta} = \beta \frac{\partial}{\partial t} \log q_t - \int dx\; p_{t,\beta} \beta \frac{\partial}{\partial t} \log q_t \tag{59}$$

$$= -\beta\big\langle \nabla, v_t\big\rangle - \beta\big\langle \nabla \log q_t, v_t\big\rangle - \int dx\; p_{t,\beta}\big[-\beta\big\langle \nabla, v_t\big\rangle - \beta\big\langle \nabla \log q_t, v_t\big\rangle\big] \tag{60}$$

$$= -\big\langle \nabla, \beta v_t\big\rangle - \big\langle \nabla \log p_{t,\beta}, v_t\big\rangle - \int dx\; p_{t,\beta}\big[-\beta\big\langle \nabla, v_t\big\rangle - \big\langle \nabla \log p_{t,\beta}, v_t\big\rangle\big] \tag{61}$$

$$= -\big\langle \nabla, \beta v_t\big\rangle - \big\langle \nabla \log p_{t,\beta}, \beta v_t\big\rangle - (1-\beta)\big\langle \nabla \log p_{t,\beta}, v_t\big\rangle \tag{62}$$

$$- \int dx\; p_{t,\beta}\big[-(1-\beta)\big\langle \nabla \log p_{t,\beta}, v_t\big\rangle\big] . \tag{63}$$

Thus, we have

$$\frac{\partial}{\partial t} p_{t,\beta}(x) = -\big\langle \nabla, p_{t,\beta}(x)\beta v_t(x)\big\rangle + p_{t,\beta}(x)\big[g_t(x) - \mathbb{E}_{p_{t,\beta}} g_t(x)\big] , \tag{64}$$

$$g_t(x) = -(1-\beta)\big\langle \nabla \log p_{t,\beta}, v_t\big\rangle , \tag{65}$$

which can be simulated as

$$dx_t = \beta v_t(x_t) dt , \tag{66}$$

$$dw_t = \beta(\beta-1)\big\langle \nabla \log q_t(x_t), v_t(x_t)\big\rangle dt . \tag{67}$$

□

Proposition D.3 (Annealed Diffusion Equation)**.** *Consider the marginals generated by the diffusion equation*

$$\frac{\partial q_t(x)}{\partial t} = \frac{\sigma_t^2}{2} \Delta q_t(x) . \tag{68}$$

The marginals $p_{t,\beta}(x) \propto q_t^\beta(x)$ satisfy the following PDE

$$\frac{\partial}{\partial t} p_{t,\beta}(x) = \frac{\sigma_t^2}{2} \Delta p_{t,\beta}(x) + p_{t,\beta}(x)\big[g_t(x) - \mathbb{E}_{p_{t,\beta}} g_t(x)\big] , \tag{69}$$

$$g_t(x) = -\beta(\beta-1)\frac{\sigma_t^2}{2}\|\nabla \log q_t(x)\|^2 . \tag{70}$$

Proof. We want to find the partial derivative of the annealed density

$$p_{t,\beta}(x) = \frac{q_t(x)^\beta}{\int dx\; q_t(x)^\beta} , \quad \frac{\partial}{\partial t} p_{t,\beta}(x) = ? \tag{71}$$

By the straightforward calculations we have

$$\frac{\partial}{\partial t}\log p_{t,\beta} = \beta\frac{\partial}{\partial t}\log q_t - \int dx\; p_{t,\beta}\beta\frac{\partial}{\partial t}\log q_t \tag{72}$$

$$= \beta\frac{\sigma_t^2}{2}\Delta\log q_t + \beta\frac{\sigma_t^2}{2}\|\nabla\log q_t\|^2 - \int dx\; p_{t,\beta}\left[\beta\frac{\sigma_t^2}{2}\Delta\log q_t + \beta\frac{\sigma_t^2}{2}\|\nabla\log q_t\|\right] \tag{73}$$

$$= \frac{\sigma_t^2}{2}\Delta\log p_{t,\beta} + \frac{\sigma_t^2}{2\beta}\|\nabla\log p_{t,\beta}\|^2 - \int dx\; p_{t,\beta}\left[\frac{\sigma_t^2}{2}\Delta\log p_{t,\beta} + \frac{\sigma_t^2}{2\beta}\|\nabla\log p_{t,\beta}\|^2\right] \tag{74}$$

$$= \frac{\sigma_t^2}{2}\Delta\log p_{t,\beta} + \frac{\sigma_t^2}{2}\|\nabla\log p_{t,\beta}\|^2 - \left(1-\frac{1}{\beta}\right)\frac{\sigma_t^2}{2}\|\nabla\log p_{t,\beta}\|^2 \tag{75}$$

$$- \int dx\; p_{t,\beta}\left[-\left(1-\frac{1}{\beta}\right)\frac{\sigma_t^2}{2}\|\nabla\log p_{t,\beta}\|^2\right]. \tag{76}$$

Thus, we have

$$\frac{\partial}{\partial t}p_{t,\beta}(x) = \frac{\sigma_t^2}{2}\Delta p_{t,\beta}(x) + p_{t,\beta}(x)\left[g_t(x) - \mathbb{E}_{p_{t,\beta}}g_t(x)\right], \tag{77}$$

$$g_t(x) = -\beta(\beta-1)\frac{\sigma_t^2}{2}\|\nabla\log q_t(x)\|^2, \tag{78}$$

which can be simulated as

$$dx_t = \sigma_t dW_t, \tag{79}$$

$$dw_t = -\beta(\beta-1)\frac{\sigma_t^2}{2}\|\nabla\log q_t(x_t)\|^2 dt. \tag{80}$$

□

Proposition D.4 (Scaled Annealed Diffusion Equation)**.** *Consider the marginals generated by the diffusion equation*

$$\frac{\partial q_t(x)}{\partial t} = \frac{\sigma_t^2}{2}\Delta q_t(x). \tag{81}$$

The marginals $p_{t,\beta}(x) \propto q_t^\beta(x)$ satisfy the following PDE

$$\frac{\partial}{\partial t}p_{t,\beta}(x) = \frac{\sigma_t^2}{2\beta}\Delta p_{t,\beta}(x) + p_{t,\beta}(x)\left[g_t(x) - \mathbb{E}_{p_{t,\beta}}g_t(x)\right], \tag{82}$$

$$g_t(x) = (\beta-1)\frac{\sigma_t^2}{2}\Delta\log q_t(x). \tag{83}$$

Proof. We want to find the partial derivative of the annealed density

$$p_{t,\beta}(x) = \frac{q_t(x)^\beta}{\int dx\; q_t(x)^\beta}, \quad \frac{\partial}{\partial t}p_{t,\beta}(x) = ? \tag{84}$$

By the straightforward calculations we have

$$\frac{\partial}{\partial t}\log p_{t,\beta} = \beta\frac{\partial}{\partial t}\log q_t - \int dx\; p_{t,\beta}\beta\frac{\partial}{\partial t}\log q_t \tag{85}$$

$$= \beta\frac{\sigma_t^2}{2}\Delta\log q_t + \beta\frac{\sigma_t^2}{2}\|\nabla\log q_t\|^2 - \int dx\; p_{t,\beta}\left[\beta\frac{\sigma_t^2}{2}\Delta\log q_t + \beta\frac{\sigma_t^2}{2}\|\nabla\log q_t\|\right] \tag{86}$$

$$= \frac{\sigma_t^2}{2}\Delta\log p_{t,\beta} + \frac{\sigma_t^2}{2\beta}\|\nabla\log p_{t,\beta}\|^2 - \int dx\; p_{t,\beta}\left[\frac{\sigma_t^2}{2}\Delta\log p_{t,\beta} + \frac{\sigma_t^2}{2\beta}\|\nabla\log p_{t,\beta}\|^2\right] \tag{87}$$

$$= \frac{\sigma_t^2}{2\beta}\Delta\log p_{t,\beta} + \frac{\sigma_t^2}{2\beta}\|\nabla\log p_{t,\beta}\|^2 + \left(1-\frac{1}{\beta}\right)\frac{\sigma_t^2}{2}\Delta\log p_{t,\beta} \tag{88}$$

$$- \int dx\; p_{t,\beta}\left[\left(1-\frac{1}{\beta}\right)\frac{\sigma_t^2}{2}\Delta\log p_{t,\beta}\right]. \tag{89}$$

Thus, we have

$$\frac{\partial}{\partial t}p_{t,\beta}(x) = \frac{\sigma_t^2}{2\beta}\Delta p_{t,\beta}(x) + p_{t,\beta}(x)\left[g_t(x) - \mathbb{E}_{p_{t,\beta}}g_t(x)\right], \tag{90}$$

$$g_t(x) = (\beta-1)\frac{\sigma_t^2}{2}\Delta\log q_t(x), \tag{91}$$

which can be simulated as

$$dx_t = \frac{\sigma_t}{\sqrt{\beta}}dW_t, \tag{92}$$

$$dw_t = (\beta-1)\frac{\sigma_t^2}{2}\Delta\log q_t(x_t)dt. \tag{93}$$

□

Proposition D.5 (Annealed Re-weighting). *Consider the marginals generated by the re-weighting equation*

$$\frac{\partial q_t(x)}{\partial t} = q_t(x)\big(g_t(x) - \mathbb{E}_{q_t(x)}g_t(x)\big). \tag{94}$$

The marginals $p_{t,\beta}(x) \propto q_t^\beta(x)$ satisfy the following PDE

$$\frac{\partial}{\partial t}p_{t,\beta}(x) = p_{t,\beta}\left[\beta g_t(x) - \mathbb{E}_{p_{t,\beta}}\beta g_t(x)\right]. \tag{95}$$

Proof. We want to find the partial derivative of the annealed density

$$p_{t,\beta}(x) = \frac{q_t(x)^\beta}{\int dx\; q_t(x)^\beta}, \quad \frac{\partial}{\partial t}p_{t,\beta}(x) = ? \tag{96}$$

By the straightforward calculations we have

$$\frac{\partial}{\partial t}\log p_{t,\beta} = \beta\frac{\partial}{\partial t}\log q_t - \int dx\; p_{t,\beta}\beta\frac{\partial}{\partial t}\log q_t \tag{97}$$

$$= \beta\big(g_t(x) - \mathbb{E}_{q_t(x)}g_t(x)\big) - \int dx\; p_{t,\beta}\left[\beta\big(g_t(x) - \mathbb{E}_{q_t(x)}g_t(x)\big)\right] \tag{98}$$

$$= \beta g_t(x) - \int dx\; p_{t,\beta}\beta g_t(x). \tag{99}$$

Thus, we have

$$\frac{\partial}{\partial t}p_{t,\beta}(x) = p_{t,\beta}\left[\beta g_t(x) - \mathbb{E}_{p_{t,\beta}}\beta g_t(x)\right], \tag{100}$$

which can be simulated as

$$dx_t = 0, \tag{101}$$

$$dw_t = \beta g_t(x_t). \tag{102}$$

$\square$

Proposition D.6 (Time-dependent annealing). *Consider the annealed marginals $p_{t,\beta}(x) \propto q_t(x)^\beta$ following some F*

$$dx_t = v_{t,\beta}(x_t) + \sigma_{t,\beta} dW_t \,, \tag{103}$$
$$dw_t = g_{t,\beta}(x_t) \,. \tag{104}$$

Then, for the time-dependent schedule β_t, we have

$$dx_t = v_{t,\beta_t}(x_t) + \sigma_{t,\beta_t} dW_t \,, \tag{105}$$
$$dw_t = g_{t,\beta_t}(x_t) + \frac{\partial \beta_t}{\partial t} \log q_t(x_t) \,, \tag{106}$$

sampling from $p_{t,\beta_t}(x) \propto q_t(x)^{\beta_t}$.

Proof. First, let's note that for the annealed marginals $p_{t,\beta}(x) \propto q_t(x)^\beta$ with constant β, we have

$$\frac{\partial}{\partial t} \log p_{t,\beta} = \beta \frac{\partial}{\partial t} \log q_t - \int dx\; p_{t,\beta} \left[\beta \frac{\partial}{\partial t} \log q_t \right] \tag{107}$$
$$= -\frac{1}{p_{t,\beta}} \langle \nabla, p_{t,\beta} v_{t,\beta} \rangle + \frac{1}{p_{t,\beta}} \frac{\sigma_{t,\beta}^2}{2} \Delta p_{t,\beta} + \left(g_{t,\beta} - \mathbb{E}_{p_{t,\beta}} g_{t,\beta} \right) . \tag{108}$$

Thus, for the time-dependent β_t, we have

$$\frac{\partial}{\partial t} \log p_{t,\beta_t} = \beta_t \frac{\partial}{\partial t} \log q_t + \frac{\partial \beta_t}{\partial t} \log q_t - \int dx\; p_{t,\beta_t} \left[\beta_t \frac{\partial}{\partial t} \log q_t + \frac{\partial \beta_t}{\partial t} \log q_t \right] \tag{109}$$
$$\begin{aligned} &= -\frac{1}{p_{t,\beta_t}} \langle \nabla, p_{t,\beta_t} v_{t,\beta_t} \rangle + \frac{1}{p_{t,\beta_t}} \frac{\sigma_{t,\beta_t}^2}{2} \Delta p_{t,\beta_t} \\ &\quad + \left[\left(g_{t,\beta_t} + \frac{\partial \beta_t}{\partial t} \log q_t \right) - \mathbb{E}_{p_{t,\beta_t}} \left(g_{t,\beta_t} + \frac{\partial \beta_t}{\partial t} \log q_t \right) \right] . \end{aligned} \tag{110}$$

From which we have the statement of the proposition. $\square$

D.2 PRODUCT

Proposition D.7 (Product of Continuity Equations). *Consider marginals $q_t^{1,2}(x)$ generated by two different continuity equations*

$$\frac{\partial q_t^1(x)}{\partial t} = -\langle \nabla, q_t^1(x) v_t^1(x) \rangle \,, \quad \frac{\partial q_t^2(x)}{\partial t} = -\langle \nabla, q_t^2(x) v_t^2(x) \rangle \,. \tag{111}$$

The product of densities $p_t(x) \propto q^1(x) q^2(x)$ satisfies the following PDE

$$\frac{\partial}{\partial t} p_t(x) = -\left\langle \nabla, p_t(x) \big(v_t^1(x) + v_t^2(x) \big) \right\rangle + p_t(x) \big(g_t(x) - \mathbb{E}_{p_t(x)} g_t(x) \big) \,, \tag{112}$$
$$g_t(x) = \left\langle \nabla \log q_t^1(x), v_t^2(x) \right\rangle + \left\langle \nabla \log q_t^2(x), v_t^1(x) \right\rangle . \tag{113}$$

Proof. For the continuity equations

$$\frac{\partial}{\partial t} q_t^{1,2}(x) = -\left\langle \nabla, q_t^{1,2}(x) v_t^{1,2}(x) \right\rangle , \tag{114}$$

we want to find the partial derivative of the annealed density

$$p_t(x) = \frac{q_t^1(x) q_t^2(x)}{\int dx\; q_t^1(x) q_t^2(x)} \,, \quad \frac{\partial}{\partial t} p_t(x) = ? \tag{115}$$

By the straightforward calculations we have

$$\frac{\partial}{\partial t}\log p_t = \frac{\partial}{\partial t}\log q_t^1 + \frac{\partial}{\partial t}\log q_t^2 - \int dx\; p_t\left[\frac{\partial}{\partial t}\log q_t^1 + \frac{\partial}{\partial t}\log q_t^2\right] \tag{116}$$

$$= -\left\langle\nabla, v_t^1 + v_t^2\right\rangle - \left\langle\nabla\log q_t^1, v_t^1\right\rangle - \left\langle\nabla\log q_t^2, v_t^2\right\rangle - \tag{117}$$

$$-\int dx\; p_t\big[-\left\langle\nabla, v_t^1 + v_t^2\right\rangle - \left\langle\nabla\log q_t^1, v_t^1\right\rangle - \left\langle\nabla\log q_t^2, v_t^2\right\rangle\big] \tag{118}$$

$$= -\left\langle\nabla, v_t^1 + v_t^2\right\rangle - \left\langle\nabla\log p_t, v_t^1 + v_t^2\right\rangle + \left\langle\nabla\log q_t^1, v_t^2\right\rangle + \left\langle\nabla\log q_t^2, v_t^1\right\rangle - \tag{119}$$

$$-\int dx\; p_t\big[\left\langle\nabla\log q_t^1, v_t^2\right\rangle + \left\langle\nabla\log q_t^2, v_t^1\right\rangle\big]\,. \tag{120}$$

Thus, we have

$$\frac{\partial}{\partial t}p_t(x) = -\left\langle\nabla, p_t(x)\big(v_t^1(x) + v_t^2(x)\big)\right\rangle + p_t(x)\big(g_t(x) - \mathbb{E}_{p_t(x)}g_t(x)\big)\,, \tag{121}$$

$$g_t(x) = \left\langle\nabla\log q_t^1(x), v_t^2(x)\right\rangle + \left\langle\nabla\log q_t^2(x), v_t^1(x)\right\rangle\,, \tag{122}$$

which can be simulated as

$$dx_t = \big(v_t^1(x_t) + v_t^2(x_t)\big)dt\,, \tag{123}$$

$$dw_t = \big[\left\langle\nabla\log q_t^1(x_t), v_t^2(x_t)\right\rangle + \left\langle\nabla\log q_t^2(x_t), v_t^1(x_t)\right\rangle\big]dt\,. \tag{124}$$

□

Proposition D.8 (Product of Diffusion Equations). *Consider marginals $q_t^{1,2}(x)$ generated by two different diffusion equations*

$$\frac{\partial q_t^1(x)}{\partial t} = \frac{\sigma_t^2}{2}\Delta q_t^1(x)\,,\quad \frac{\partial q_t^2(x)}{\partial t} = \frac{\sigma_t^2}{2}\Delta q_t^2(x)\,. \tag{125}$$

The product of densities $p_t(x) \propto q^1(x)q^2(x)$ satisfies the following PDE

$$\frac{\partial}{\partial t}p_t(x) = \frac{\sigma_t^2}{2}\Delta p_t(x) + p_t(x)\big(g_t(x) - \mathbb{E}_{p_t(x)}g_t(x)\big)\,, \tag{126}$$

$$g_t(x) = -\sigma_t^2\left\langle\nabla\log q_t^1(x), \nabla\log q_t^2(x)\right\rangle\,. \tag{127}$$

Proof. We want to find the partial derivative of the annealed density

$$p_t(x) = \frac{q_t^1(x)q_t^2(x)}{\int dx\; q_t^1(x)q_t^2(x)}\,,\quad \frac{\partial}{\partial t}p_t(x) = ? \tag{128}$$

By straightforward calculations we have

$$\begin{aligned}
\frac{\partial}{\partial t}\log p_t &= \frac{\partial}{\partial t}\log q_t^1 + \frac{\partial}{\partial t}\log q_t^2 - \int dx\; p_t\left[\frac{\partial}{\partial t}\log q_t^1 + \frac{\partial}{\partial t}\log q_t^2\right]\\
&= \frac{\sigma_t^2}{2}\Delta\log q_t^1 + \frac{\sigma_t^2}{2}\left\|\nabla\log q_t^1\right\|^2 + \frac{\sigma_t^2}{2}\Delta\log q_t^2 + \frac{\sigma_t^2}{2}\left\|\nabla\log q_t^2\right\|^2\\
&\quad - \int dx\; p_t\left[\frac{\sigma_t^2}{2}\Delta\log q_t^1 + \frac{\sigma_t^2}{2}\left\|\nabla\log q_t^1\right\|^2 + \frac{\sigma_t^2}{2}\Delta\log q_t^2 + \frac{\sigma_t^2}{2}\left\|\nabla\log q_t^2\right\|^2\right]\\
&= \frac{\sigma_t^2}{2}\Delta\log p_t + \frac{\sigma_t^2}{2}\|\nabla\log p_t\|^2 - \sigma_t^2\left\langle\nabla\log q_t^1, \nabla\log q_t^2\right\rangle\\
&\quad - \int dx\; p_t\big[-\sigma_t^2\left\langle\nabla\log q_t^1, \nabla\log q_t^2\right\rangle\big]\,.
\end{aligned} \tag{129}$$

Thus, we have

$$\frac{\partial}{\partial t}p_t(x) = \frac{\sigma_t^2}{2}\Delta p_t(x) + p_t(x)\big(g_t(x) - \mathbb{E}_{p_t(x)}g_t(x)\big)\,, \tag{130}$$

$$g_t(x) = -\sigma_t^2\left\langle\nabla\log q_t^1(x), \nabla\log q_t^2(x)\right\rangle\,, \tag{131}$$

which can be simulated as

$$dx_t = \sigma_t dW_t\,, \tag{132}$$
$$dw_t = \left[-\sigma_t^2\langle\nabla\log q_t^1(x_t), \nabla\log q_t^2(x_t)\rangle\right]dt\,. \tag{133}$$

□

Proposition D.9 (Product of Re-weightings). *Consider marginals $q_t^{1,2}(x)$ generated by two different diffusion equations*

$$\frac{\partial q_t^1(x)}{\partial t} = \Big(g_t^1(x) - \mathbb{E}_{q_t^1} g_t^1(x)\Big)q_t^1(x)\,,\quad \frac{\partial q_t^2(x)}{\partial t} = \Big(g_t^2(x) - \mathbb{E}_{q_t^2} g_t^2(x)\Big)q_t^2(x)\,. \tag{134}$$

The product of densities $p_t(x) \propto q^1(x)q^2(x)$ satisfies the following PDE

$$\frac{\partial}{\partial t}p_t(x) = p_t(x)\big(g_t(x) - \mathbb{E}_{p_t(x)} g_t(x)\big)\,, \tag{135}$$
$$g_t(x) = g_t^1(x) + g_t^2(x)\,, \tag{136}$$

Proof. We want to find the partial derivative of the annealed density

$$p_t(x) = \frac{q_t^1(x)q_t^2(x)}{\int dx\; q_t^1(x)q_t^2(x)}\,,\quad \frac{\partial}{\partial t}p_t(x) = ? \tag{137}$$

By the straightforward calculations we have

$$\frac{\partial}{\partial t}\log p_t = \frac{\partial}{\partial t}\log q_t^1 + \frac{\partial}{\partial t}\log q_t^2 - \int dx\; p_t\left[\frac{\partial}{\partial t}\log q_t^1 + \frac{\partial}{\partial t}\log q_t^2\right] \tag{138}$$
$$= \Big(g_t^1(x) - \mathbb{E}_{q_t^1} g_t^1(x)\Big) + \Big(g_t^2(x) - \mathbb{E}_{q_t^2} g_t^2(x)\Big) - \tag{139}$$
$$- \int dx\; p_t\Big[\Big(g_t^1(x) - \mathbb{E}_{q_t^1} g_t^1(x)\Big) + \Big(g_t^2(x) - \mathbb{E}_{q_t^2} g_t^2(x)\Big)\Big] \tag{140}$$
$$= g_t^1(x) + g_t^2(x) - \int dx\; p_t\big[g_t^1(x) + g_t^2(x)\big]\,. \tag{141}$$

Thus, we have

$$\frac{\partial}{\partial t}p_t(x) = p_t(x)\big(g_t(x) - \mathbb{E}_{p_t(x)} g_t(x)\big)\,, \tag{142}$$
$$g_t(x) = g_t^1(x) + g_t^2(x)\,, \tag{143}$$

which can be simulated as

$$dx_t = 0\,, \tag{144}$$
$$dw_t = g_t^1(x_t) + g_t^2(x_t)\,. \tag{145}$$

□

E Proofs of Propositions

Proposition E.1 (Annealed SDE). *Consider the SDE*

$$dx_t = \big(-f_t(x_t) + \sigma_t^2\nabla\log q_t(x_t)\big)dt + \sigma_t dW_t\,, \tag{146}$$

then the samples from the annealed marginals $p_{t,\beta}(x) \propto q_t(x)^\beta$ can be obtained via the following family of SDEs

$$dx_t = \big(-f_t(x_t) + (\beta + (1-\beta)a)\sigma_t^2\nabla\log q_t(x_t)\big)dt + \sqrt{\frac{\sigma_t^2(\beta + (1-\beta)2a)}{\beta}}dW_t\,, \tag{147}$$

$$dw_t = \left[(\beta-1)\langle\nabla, f_t(x_t)\rangle + \frac{1}{2}\sigma_t^2\beta(\beta-1)\|\nabla\log q_t(x_t)\|^2\right]dt\,, \tag{148}$$

where the parameter $a \in [0, 1/2]$.

Proof. For the following SDE

$$dx_t = \big(-f_t(x_t) + \sigma_t^2 \nabla \log q_t(x_t)\big)dt + \sigma_t dW_t\,, \tag{149}$$

let's consider everything but the drift f_t. Thus, we can write the following PDE

$$\frac{\partial q_t}{\partial t} = \big\langle \nabla, q_t\big[(1-a)\sigma_t^2 \nabla \log q_t(x_t) + a\sigma_t^2 \nabla \log q_t(x_t)\big]\big\rangle + (1-b)\frac{\sigma_t^2}{2}\Delta q_t + b\frac{\sigma_t^2}{2}\Delta q_t\,. \tag{150}$$

We apply Prop. D.2, Prop. D.1, Prop. D.4, Prop. D.3 (rules from Table 1) to the corresponding terms of the PDE above. Hence, the formulas for the weights are

$$\begin{aligned} g_t(x) = {} & (1-a)\sigma_t^2\beta(\beta-1)\|\nabla \log q_t(x)\|^2 - a\sigma_t^2(\beta-1)\Delta \log q_t(x) \\ & + (\beta-1)\frac{(1-b)\sigma_t^2}{2}\Delta \log q_t(x_t) - \beta(\beta-1)\frac{b\sigma_t^2}{2}\|\nabla \log q_t(x_t)\|^2\,. \end{aligned} \tag{151}$$

Let's cancel out the term with the Laplacians, hence, we have $2a = 1-b$ (hence, $a \in [0, 1/2]$) and

$$g_t(x) = (1-a-b/2)\sigma_t^2\beta(\beta-1)\|\nabla \log q_t(x)\|^2 = \frac{1}{2}\sigma_t^2\beta(\beta-1)\|\nabla \log q_t(x)\|^2\,. \tag{152}$$

The PDE for the density is

$$\begin{aligned} \frac{\partial p_{t,\beta}}{\partial t} = {} & -\big\langle \nabla, p_{t,\beta}\big(-f_t + (\beta(1-a)+a)\sigma_t^2\nabla \log q_t\big)\big\rangle \\ & + \left(\frac{1-b}{\beta} + b\right)\frac{\sigma_t^2}{2}\Delta p_{t,\beta} + p_{t,\beta}\big(g_t - \mathbb{E}_{p_{t,\beta}} g_t\big) \qquad (153) \\ = {} & -\big\langle \nabla, p_{t,\beta}\big(-f_t + (\beta+(1-\beta)a)\sigma_t^2\nabla \log q_t\big)\big\rangle \\ & + \frac{\beta+(1-\beta)2a}{\beta}\frac{\sigma_t^2}{2}\Delta p_{t,\beta} + p_{t,\beta}\big(g_t - \mathbb{E}_{p_{t,\beta}} g_t\big) \qquad (154) \end{aligned}$$

This corresponds to the following family of SDEs ($a \in [0, 1/2]$)

$$dx_t = \big(-f_t(x_t) + (\beta+(1-\beta)a)\sigma_t^2\nabla \log q_t(x_t)\big)dt + \sqrt{\frac{\sigma_t^2(\beta+(1-\beta)2a)}{\beta}}dW_t\,, \tag{155}$$

$$dw_t = \left[(\beta-1)\big\langle \nabla, f_t(x_t)\big\rangle + \frac{1}{2}\sigma_t^2\beta(\beta-1)\|\nabla \log q_t(x_t)\|^2\right]dt\,. \tag{156}$$

□

Proposition E.2 (Product of Experts)**.** *Consider two PDEs corresponding to the following SDEs*

$$dx_t = (-f_t(x_t) + \sigma_t^2\nabla \log q_t^{1,2}(x_t))dt + \sigma_t dW_t\,, \tag{157}$$

which marginals we denote as $q_t^1(x_t)$ and $q_t^2(x_t)$. The following family of SDEs (for all $a \in [0, 1/2]$) corresponds to the product of the marginals $p_{t,\beta}(x) \propto (q_t^1(x)q_t^2(x))^\beta$

$$\begin{aligned} dx_t = {} & \big(-f_t(x_t) + \sigma_t^2(\beta+(1-\beta)a)(\nabla \log q_t^1(x_t) + \nabla \log q_t^2(x_t))\big)dt \\ & + \sqrt{\frac{\sigma_t^2(\beta+(1-\beta)2a)}{\beta}}dW_t\,, \qquad (158) \\ dw_t = {} & \Big[\beta\sigma_t^2\big\langle \nabla \log q_t^1(x_t), \nabla \log q_t^2(x_t)\big\rangle + \beta(\beta-1)\frac{\sigma_t^2}{2}\big\|\nabla \log q_t^1(x_t) + \nabla \log q_t^2(x_t)\big\|^2 \\ & + (2\beta-1)\big\langle \nabla, f_t(x_t)\big\rangle\Big]dt\,. \qquad (159) \end{aligned}$$

Proof. First, according to Table 1, we have the following PDE for the product density $p_t(x) \propto q_t^1(x)q_t^2(x)$ is

$$\frac{\partial p_t(x)}{\partial t} = -\left\langle\nabla, p_t(x)\big(-2f_t(x)+\sigma_t^2(\nabla\log q_t^1(x)+\nabla\log q_t^2(x))\big)\right\rangle + \frac{\sigma_t^2}{2}\Delta p_t(x) + \tag{160}$$

$$+\, p_t(x)(g_t(x) - \mathbb{E}_{p_t} g_t(x))\,, \tag{161}$$

$$\begin{aligned} g_t(x) &= \big\langle\nabla\log q_t^1(x), -f_t(x)+\sigma_t^2\nabla\log q_t^2(x)\big\rangle + \big\langle\nabla\log q_t^2(x), -f_t(x)+\sigma_t^2\nabla\log q_t^1(x)\big\rangle \\ &\quad - \sigma_t^2\big\langle\nabla\log q_t^1(x), \nabla\log q_t^2(x)\big\rangle \\ &= \sigma_t^2\big\langle\nabla\log q_t^1(x), \nabla\log q_t^2(x)\big\rangle - \big\langle f_t(x), \nabla\log q_t^1(x)+\nabla\log q_t^2(x)\big\rangle\,. \end{aligned} \tag{162}$$

Now, combining Prop. E.1 and Prop. D.5, for the annealed density $p_{t,\beta} \propto p_t(x)^\beta$ we have

$$\begin{aligned} \frac{\partial p_{t,\beta}(x)}{\partial t} &= -\left\langle\nabla, p_{t,\beta}(x)\big(-2f_t(x)+\sigma_t^2(\beta+(1-\beta)a)(\nabla\log q_t^1(x)+\nabla\log q_t^2(x))\big)\right\rangle \\ &\quad + \frac{\beta+(1-\beta)2a}{\beta}\frac{\sigma_t^2}{2}\Delta p_{t,\beta}(x) + p_{t,\beta}(x)\big(g_t(x) - \mathbb{E}_{p_{t,\beta}} g_t(x)\big)\,, \end{aligned} \tag{163}$$

$$\begin{aligned} g_t(x) &= \beta\sigma_t^2\big\langle\nabla\log q_t^1(x), \nabla\log q_t^2(x)\big\rangle - \beta\big\langle f_t(x), \nabla\log q_t^1(x)+\nabla\log q_t^2(x)\big\rangle \\ &\quad + (\beta-1)\big\langle\nabla, 2f_t(x)\big\rangle + \beta(\beta-1)\frac{\sigma_t^2}{2}\big\|\nabla\log q_t^1(x)+\nabla\log q_t^2(x)\big\|^2\,. \end{aligned} \tag{164}$$

The last step is interpreting $\big\langle\nabla, p_{t,\beta}(x)f_t(x)\big\rangle$ as the weight term, i.e.

$$\begin{aligned} \frac{\partial p_{t,\beta}(x)}{\partial t} &= -\left\langle\nabla, p_{t,\beta}(x)\big(-f_t(x)+\sigma_t^2(\beta+(1-\beta)a)(\nabla\log q_t^1(x)+\nabla\log q_t^2(x))\big)\right\rangle \\ &\quad + \frac{\beta+(1-\beta)2a}{\beta}\frac{\sigma_t^2}{2}\Delta p_{t,\beta}(x) + p_{t,\beta}(x)\big(g_t(x) - \mathbb{E}_{p_{t,\beta}} g_t(x)\big)\,, \end{aligned} \tag{165}$$

$$g_t(x) = \beta\sigma_t^2\big\langle\nabla\log q_t^1(x), \nabla\log q_t^2(x)\big\rangle + \beta(\beta-1)\frac{\sigma_t^2}{2}\big\|\nabla\log q_t^1(x)+\nabla\log q_t^2(x)\big\|^2 + \tag{166}$$

$$+\, (2\beta-1)\big\langle\nabla, f_t(x)\big\rangle\,. \tag{167}$$

Thus, we get the following family of SDEs (for all $a \in [0, 1/2]$)

$$dx_t = \big(-f_t(x_t)+\sigma_t^2(\beta+(1-\beta)a)(\nabla\log q_t^1(x_t)+\nabla\log q_t^2(x_t))\big)dt + \sqrt{\frac{\sigma_t^2(\beta+(1-\beta)2a)}{\beta}}dW_t\,, \tag{168}$$

$$\begin{aligned} dw_t = \Big[&\beta\sigma_t^2\big\langle\nabla\log q_t^1(x_t), \nabla\log q_t^2(x_t)\big\rangle \\ &+ \beta(\beta-1)\frac{\sigma_t^2}{2}\big\|\nabla\log q_t^1(x_t)+\nabla\log q_t^2(x_t)\big\|^2 + (2\beta-1)\big\langle\nabla, f_t(x_t)\big\rangle\Big]dt\,. \end{aligned} \tag{169}$$

□

Proposition E.3 (Classifier-free Guidance). *Consider two PDEs corresponding to the following SDEs*

$$dx_t = (-f_t(x_t)+\sigma_t^2\nabla\log q_t^{1,2}(x_t))dt + \sigma_t dW_t\,, \tag{170}$$

which marginals we denote as $q_t^1(x_t)$ and $q_t^2(x_t)$. The SDE corresponding to the geometric average of the marginals $p_{t,\beta}(x) \propto q_t^1(x)^{1-\beta}q_t^2(x)^\beta$ is

$$dx_t = \big(-f_t(x_t)+\sigma_t^2((1-\beta)\nabla\log q_t^1(x_t)+\beta\nabla\log q_t^2(x_t))\big)dt + \sigma_t dW_t\,, \tag{171}$$

$$dw_t = \frac{1}{2}\sigma_t^2\beta(\beta-1)\big\|\nabla\log q_t^1(x_t)-\nabla\log q_t^2(x_t)\big\|^2\,. \tag{172}$$

Proof. First, according to Prop. E.1, we perform annealing $p^1_{t,1-\beta}(x) \propto q^1_t(x)^{1-\beta}$ and $p^2_{t,\beta}(x) \propto q^2_t(x)^\beta$, i.e.

$$\begin{aligned}\frac{\partial p^1_{t,1-\beta}(x)}{\partial t} =& -\big\langle \nabla, p^1_{t,1-\beta}(x)\big(-f_t(x) + \sigma_t^2(1-\beta-a_1)\nabla \log q^1_t(x)\big)\big\rangle \\ &+ \frac{1-\beta-2a_1}{1-\beta}\frac{\sigma_t^2}{2}\Delta p^1_{t,1-\beta}(x) + p^1_{t,1-\beta}(x)\Big(g_t(x) - \mathbb{E}_{p^1_{t,1-\beta}} g_t(x)\Big)\,, \quad (173)\\ g_t(x) =& -\beta\big\langle \nabla, f_t(x)\big\rangle + \frac{1}{2}\sigma_t^2\beta(\beta-1)\big\|\nabla \log q^1_t(x_t)\big\|^2\,, \quad (174)\end{aligned}$$

and

$$\begin{aligned}\frac{\partial p^2_{t,\beta}(x)}{\partial t} =& -\big\langle \nabla, p^2_{t,\beta}(x)\big(-f_t(x) + \sigma_t^2(\beta+(1-\beta)a_2)\nabla \log q^2_t(x)\big)\big\rangle \\ &+ \frac{\beta(1-\beta)2a_2}{\beta}\frac{\sigma_t^2}{2}\Delta p^2_{t,\beta}(x) + p^2_{t,\beta}(x)\Big(g_t(x) - \mathbb{E}_{p^2_{t,\beta}} g_t(x)\Big)\,, \quad (175)\\ g_t(x) =& (\beta-1)\big\langle \nabla, f_t(x)\big\rangle + \frac{1}{2}\sigma_t^2\beta(\beta-1)\big\|\nabla \log q^2_t(x_t)\big\|^2\,, \quad (176)\end{aligned}$$

Now, according to Table 1, for the product density $p_{t,\beta} \propto p^1_{t,1-\beta}(x)p^2_{t,\beta}(x)$. However, first, we have to match the diffusion coefficient

$$\frac{1-\beta-2a_1}{1-\beta} = \frac{\beta+(1-\beta)2a_2}{\beta} \implies (1-2a_1)\beta - \beta^2 = \beta - \beta^2 + (1-\beta)^2 2a_2 \quad (177)$$

$$a_1\beta + (1-\beta)^2 a_2 = 0 \implies a_2 \coloneqq a\,,\;\; a_1 = \frac{-a(1-\beta)^2}{\beta}\,. \quad (178)$$

However, we see that the only possible solution that have $a_1 \in [0, 1/2]$ and $a_2 \in [0, 1/2]$ for positive β is $a_1 = a_2 = 0$. Thus, we have

$$\begin{aligned}\frac{\partial p_{t,\beta}(x)}{\partial t} =& -\big\langle \nabla, p_{t,\beta}(x)\big(-2f_t(x) + \sigma_t^2(1-\beta)\nabla \log q^1_t(x) + \beta\nabla \log q^2_t(x))\big)\big\rangle + \frac{\sigma_t^2}{2}\Delta p_{t,\beta}(x) \\ &+ p_{t,\beta}(x)\big(g_t(x) - \mathbb{E}_{p_{t,\beta}} g_t(x)\big)\,, \quad (179)\\ g_t(x) =& -\beta\big\langle \nabla, f_t(x)\big\rangle + \frac{1}{2}\sigma_t^2\beta(\beta-1)\big\|\nabla \log q^1_t(x)\big\|^2 \\ &+ (\beta-1)\big\langle \nabla, f_t(x)\big\rangle + \frac{1}{2}\sigma_t^2\beta(\beta-1)\big\|\nabla \log q^2_t(x)\big\|^2 \\ &+ (1-\beta)\big\langle \nabla \log q^1_t(x), -f_t(x) + \sigma_t^2\beta\nabla \log q^2_t(x)\big\rangle \\ &+ \beta\big\langle \nabla \log q^2_t(x), -f_t(x) + \sigma_t^2(1-\beta)\nabla \log q^1_t(x)\big\rangle \\ &- \sigma_t^2\beta(1-\beta)\big\langle \nabla \log q^1_t(x), \nabla \log q^2_t(x)\big\rangle \\ =& \frac{1}{2}\sigma_t^2\beta(\beta-1)\big\|\nabla \log q^1_t(x) - \nabla \log q^2_t(x)\big\|^2 \\ &- \big\langle \nabla, f_t(x)\big\rangle - \big\langle (1-\beta)\nabla \log q^1_t(x) + \beta\nabla \log q^2_t(x), f_t(x)\big\rangle\,. \quad (180)\end{aligned}$$

Finally, we re-interpret $\big\langle \nabla, p_{t,\beta}(x)f_t(x)\big\rangle$ as the weighting term, and get

$$\begin{aligned}\frac{\partial p_{t,\beta}(x)}{\partial t} =& -\big\langle \nabla, p_{t,\beta}(x)\big(-f_t(x) + \sigma_t^2((1-\beta)\nabla \log q^1_t(x) + \beta\nabla \log q^2_t(x))\big)\big\rangle + \frac{\sigma_t^2}{2}\Delta p_{t,\beta}(x) \\ &+ p_{t,\beta}(x)\big(g_t(x) - \mathbb{E}_{p_{t,\beta}} g_t(x)\big)\,, \quad (181)\\ g_t(x) =& \frac{1}{2}\sigma_t^2\beta(\beta-1)\big\|\nabla \log q^1_t(x) - \nabla \log q^2_t(x)\big\|^2\,. \quad (182)\end{aligned}$$

Thus, we have

$$\begin{aligned}dx_t &= \big(-f_t(x_t) + \sigma_t^2((1-\beta)\nabla \log q^1_t(x_t) + \beta\nabla \log q^2_t(x_t))\big)dt + \sigma_t dW_t\,, \quad (183)\\ dw_t &= \frac{1}{2}\sigma_t^2\beta(\beta-1)\big\|\nabla \log q^1_t(x_t) - \nabla \log q^2_t(x_t)\big\|^2\,. \quad (184)\end{aligned}$$

□

Proposition E.4 (PoE + CFG). *Consider two PDEs corresponding to the following SDEs*

$$dx_t = (-f_t(x_t) + \sigma_t^2 \nabla \log q_t(x_t))dt + \sigma_t dW_t\,, \tag{185}$$

$$dx_t = (-f_t(x_t) + \sigma_t^2 \nabla \log q_t^{1,2}(x_t))dt + \sigma_t dW_t\,, \tag{186}$$

with corresponding marginals $q_t(x_t)$, $q_t^1(x_t)$ *and* $q_t^2(x_t)$. *The SDE corresponding to the product of the marginals* $p_{t,\beta}(x) \propto q_t(x)^{2(1-\beta)}(q_t^1(x)q_t^2(x))^\beta$ *is*

$$dx_t = \big(-f_t(x_t) + \sigma_t^2(v_t^1(x_t) + v_t^2(x_t))\big)dt + \sigma_t dW_t\,, \tag{187}$$

$$\begin{aligned} dw_t = {}& \frac{1}{2}\sigma_t^2\beta(\beta-1)\Big(\big\|\nabla \log q_t(x_t) - \nabla \log q_t^1(x_t)\big\|^2 + \big\|\nabla \log q_t(x_t) - \nabla \log q_t^2(x_t)\big\|^2\Big) \\ & + \sigma_t^2\langle v_t^1(x_t), v_t^2(x_t)\rangle + \langle \nabla, f_t(x_t)\rangle\,, \end{aligned} \tag{188}$$

where we denote $v_t^{1,2}(x) = (1-\beta)\nabla \log q_t(x) + \beta \nabla \log q_t^{1,2}(x)$.

Proof. Using Prop. E.3, we start from the SDEs simulating the product $q_t(x)^{(1-\beta)}q_t^1(x)^\beta$ and $q_t(x)^{(1-\beta)}q_t^2(x)^\beta$, i.e.

$$dx_t = \Big(-f_t(x_t) + \sigma_t^2\big(\underbrace{(1-\beta)\nabla \log q_t(x_t) + \beta \nabla \log q_t^1(x_t)}_{v_t^1(x_t)}\big)\Big)dt + \sigma_t dW_t\,, \tag{189}$$

$$dw_t = \frac{1}{2}\sigma_t^2\beta(\beta-1)\big\|\nabla \log q_t(x_t) - \nabla \log q_t^1(x_t)\big\|^2\,, \tag{190}$$

$$dx_t = \Big(-f_t(x_t) + \sigma_t^2\big(\underbrace{(1-\beta)\nabla \log q_t(x_t) + \beta \nabla \log q_t^2(x_t)}_{v_t^2(x_t)}\big)\Big)dt + \sigma_t dW_t\,, \tag{191}$$

$$dw_t = \frac{1}{2}\sigma_t^2\beta(\beta-1)\big\|\nabla \log q_t(x_t) - \nabla \log q_t^2(x_t)\big\|^2\,. \tag{192}$$

Then we consider the product of these SDEs, i.e.

$$\begin{aligned} \frac{\partial p_{t,\beta}(x)}{\partial t} = {}& -\big\langle \nabla, p_{t,\beta}(x)\big(-2f_t(x) + \sigma_t^2(v_t^1(x) + v_t^2(x))\big)\big\rangle + \frac{\sigma_t^2}{2}\Delta p_{t,\beta}(x) \\ & + p_{t,\beta}(x)\big(g_t(x) - \mathbb{E}_{p_{t,\beta}} g_t(x)\big)\,, \end{aligned} \tag{193}$$

$$g_t(x) = \frac{1}{2}\sigma_t^2\beta(\beta-1)\Big(\big\|\nabla \log q_t(x) - \nabla \log q_t^1(x)\big\|^2 + \big\|\nabla \log q_t(x) - \nabla \log q_t^2(x)\big\|^2\Big) + \tag{194}$$

$$+ \big\langle v_t^1(x), -f_t(x) + \sigma_t^2 v_t^2(x)\big\rangle + \big\langle v_t^2(x), -f_t(x) + \sigma_t^2 v_t^1(x)\big\rangle - \sigma_t^2\langle v_t^1(x), v_t^2(x)\rangle \tag{195}$$

$$\begin{aligned} = {}& \frac{1}{2}\sigma_t^2\beta(\beta-1)\Big(\big\|\nabla \log q_t(x) - \nabla \log q_t^1(x)\big\|^2 + \big\|\nabla \log q_t(x) - \nabla \log q_t^2(x)\big\|^2\Big) \\ & + \sigma_t^2\langle v_t^1(x), v_t^2(x)\rangle - \langle f_t(x), v_t^1(x) + v_t^2(x)\rangle\,. \end{aligned} \tag{196}$$

Re-interpreting $\langle \nabla, p_{t,\beta}(x)f_t(x)\rangle$, we get

$$\begin{aligned} \frac{\partial p_{t,\beta}(x)}{\partial t} = {}& -\big\langle \nabla, p_{t,\beta}(x)\big(-f_t(x) + \sigma_t^2(v_t^1(x) + v_t^2(x))\big)\big\rangle \\ & + \frac{\sigma_t^2}{2}\Delta p_{t,\beta}(x) + p_{t,\beta}(x)\big(g_t(x) - \mathbb{E}_{p_{t,\beta}} g_t(x)\big)\,, \end{aligned} \tag{197}$$

$$\begin{aligned} g_t(x) = {}& \frac{1}{2}\sigma_t^2\beta(\beta-1)\Big(\big\|\nabla \log q_t(x) - \nabla \log q_t^1(x)\big\|^2 + \big\|\nabla \log q_t(x) - \nabla \log q_t^2(x)\big\|^2\Big) \\ & + \sigma_t^2\langle v_t^1(x), v_t^2(x)\rangle + \langle \nabla, f_t(x)\rangle\,, \end{aligned} \tag{198}$$

which corresponds to

$$dx_t = \big(-f_t(x_t) + \sigma_t^2(v_t^1(x_t) + v_t^2(x_t))\big)dt + \sigma_t dW_t\,, \tag{199}$$

$$\begin{aligned} dw_t = {}& \frac{1}{2}\sigma_t^2\beta(\beta-1)\Big(\big\|\nabla \log q_t(x_t) - \nabla \log q_t^1(x_t)\big\|^2 + \big\|\nabla \log q_t(x_t) - \nabla \log q_t^2(x_t)\big\|^2\Big) \\ & + \sigma_t^2\langle v_t^1(x_t), v_t^2(x_t)\rangle + \langle \nabla, f_t(x_t)\rangle\,. \end{aligned} \tag{200}$$

□

Proposition E.5 (Reward-tilted SDE). *Consider the following SDE*

$$dx_t = v_t(x)dt + \sigma_t dW_t\,, \tag{201}$$

which samples from the marginals $q_t(x)$. *The samples from the marginals* $p_t(x) \propto q_t(x)\exp(\beta_t r(x))$ *can be simulated according to the following SDE*

$$dx_t = v_t(x_t)dt + \sigma_t dW_t\,, \tag{202}$$

$$dw_t = \left[\left\langle \beta_t \nabla r(x_t), v_t(x_t) - \sigma_t^2 \nabla \log q_t(x_t) - \frac{\sigma_t^2}{2}\beta_t \nabla r(x_t)\right\rangle - \beta_t \frac{\sigma_t^2}{2}\Delta r(x_t) + \frac{\partial \beta_t}{\partial t} r(x_t)\right] dt\,. \tag{203}$$

For the reverse SDE, it is

$$dx_t = (-f_t(x_t) + \sigma_t^2 \nabla \log q_t(x_t))dt + \sigma_t dW_t\,, \tag{204}$$

$$dw_t = \left[\left\langle \beta_t \nabla r(x_t), -f_t(x_t) - \frac{\sigma_t^2}{2}\beta_t \nabla r(x_t)\right\rangle - \beta_t \frac{\sigma_t^2}{2}\Delta r(x_t) + \frac{\partial \beta_t}{\partial t} r(x_t)\right] dt \tag{205}$$

Proof. First, consider the density $q_t(x)$ that follows the PDE

$$\frac{\partial q_t(x)}{\partial t} = -\langle \nabla, q_t(x)v_t(x)\rangle + \frac{\sigma_t^2}{2}\Delta q_t(x)\,. \tag{206}$$

We want to find the PDE for the reward-tilted density

$$p_t(x) = \frac{q_t(x)\exp(\beta_t r(x))}{\int dx\; q_t(x)\exp(\beta_t r(x))}\,. \tag{207}$$

Straightforwardly, we get

$$\frac{\partial}{\partial t}\log p_t(x) = \frac{\partial}{\partial t}\log q_t(x) + \frac{\partial \beta_t}{\partial t} r(x) - \int dx\; p_t(x)\left[\frac{\partial}{\partial t}\log q_t(x) + \frac{\partial \beta_t}{\partial t} r(x)\right] \tag{208}$$

For the first term, we have

$$\frac{\partial}{\partial t}\log q_t(x) = -\langle \nabla, v_t(x)\rangle - \langle \nabla \log q_t(x), v_t(x)\rangle + \frac{\sigma_t^2}{2}\Delta \log q_t(x) + \frac{\sigma_t^2}{2}\|\nabla \log q_t(x)\|^2 \tag{209}$$

$$\begin{aligned} &= -\langle \nabla, v_t(x)\rangle - \langle \nabla \log p_t(x), v_t(x)\rangle + \frac{\sigma_t^2}{2}\Delta \log p_t(x) + \frac{\sigma_t^2}{2}\|\nabla \log p_t(x)\|^2 \\ &\quad + \left\langle \beta_t \nabla r(x), v_t(x) - \sigma_t^2 \nabla \log q_t(x) - \frac{\sigma_t^2}{2}\beta_t \nabla r(x)\right\rangle - \beta_t \frac{\sigma_t^2}{2}\Delta r(x)\,. \end{aligned} \tag{210}$$

Thus, we have

$$\frac{\partial p_t(x)}{\partial t} = -\langle \nabla, p_t(x)v_t(x)\rangle + \frac{\sigma_t^2}{2}\Delta p_t(x) + p_t(x)\big(g_t(x) - \mathbb{E}_{p_t(x)} g_t(x)\big) \tag{211}$$

$$g_t(x) = \left\langle \beta_t \nabla r(x), v_t(x) - \sigma_t^2 \nabla \log q_t(x) - \frac{\sigma_t^2}{2}\beta_t \nabla r(x)\right\rangle - \beta_t \frac{\sigma_t^2}{2}\Delta r(x) + \frac{\partial \beta_t}{\partial t} r(x)\,. \tag{212}$$

This can be simulated as

$$dx_t = v_t(x_t)dt + \sigma_t dW_t\,, \tag{213}$$

$$dw_t = \left[\left\langle \beta_t \nabla r(x_t), v_t(x_t) - \sigma_t^2 \nabla \log q_t(x_t) - \frac{\sigma_t^2}{2}\beta_t \nabla r(x_t)\right\rangle - \beta_t \frac{\sigma_t^2}{2}\Delta r(x_t) + \frac{\partial \beta_t}{\partial t} r(x_t)\right] dt \tag{214}$$

□

F Additional Experimental Details and Results

F.1 Sampling Metrics

We use a number of metrics to asses the quality of generated samples. These metrics capture different aspects of the distribution.

Energy-$\mathcal{W}_{1/2}$ The Energy-$\mathcal{W}_1$ and Energy-$\mathcal{W}_2$ measures the deviation in the energy value distribution of samples from the reference distribution and the generated distribution. We find this metric is useful to assess the overall fit of a model, although it cannot assess whether a sampler drops modes well. A model that has a reasonably small Energy Wasserstein distance may still have missed a mode of a similar energy value.

Maximum Mean Discrepancy (MMD) We use a radial-basis function MMD with multiple scales to assess distribution fit. This measures how well the reference distribution matches the generated distribution locally.

Total Variation distance For low dimensional sampling problems, it is useful to consider the total variation distance between empirical distributions that are discretized into a grid. This measures fit in terms of density, ignoring the underlying metric, and is less sensitive to global reweighting of modes.

1-Wasserstein and 2-Wasserstein distances ($\mathcal{W}_1$ / $\mathcal{W}_2$) On 40 GMM we also measure the 1-Wasserstein and 2-Wasserstein distances between the generated and reference distributions with respect to the Euclidean metric. We note that while this is possible to measure in the LJ-13 case, it is not as useful as particles in the LJ-13 setting are SE(3) equivariant, and therefore the Euclidean distance is not a suitable ground metric.

F.2 Mixture of 40 Gaussians

The mixture of 40 Gaussians setting is a 2D energy function with 40 randomly initialized modes with equal standard deviation. This serves as a useful experimental setting where we are able to calculate true densities and scores efficiently without modelling error.

F.2.1 Additional Results

We include quantitative results for the tractable GMM example in Sec. 5.1, where we start at temperature $T = 3$ and anneal to target temperature $T = 1/3$. We used a geometric noise schedule with $\sigma_{\min} = 0.01$ and $\sigma_{\max} = 500$. We sample 10k samples with 1000 integration steps, with $dt = 0.001$. We observe that Target Score sampling ($a = 0$) from Eq. (21) with systematic resampling performs best in more metrics. We also use this example as an ablation study for the impact of the resampling scheme, where we find that systematic resampling appears to outperform the birth-death exponential clocks implementation of the jump process resampling. See App. A and App. C.2.

On ground truth q_t^β A subtle point to note is that q_t^β is not a mixture of $|\pi|$ Gaussians, but rather $|\pi|^\beta$ Gaussians for integer β. This means that we are restricted to small integer β. We use $\beta = 3$ for all experiments in the 40 Gaussians setting.

Table 6: Mixture of 40 Gaussians. Sampling from an annealed distribution with inverse temperature $\beta = 3$. Metrics are calculated over 5 runs with 10k samples.

SDE Type	FKC	Energy-$\mathcal{W}_2$	MMD	Total Var	$\mathcal{W}_1$	$\mathcal{W}_2$
Target Score	✗	0.943 ± 0.026	0.020 ± 0.001	0.487 ± 0.007	11.304 ± 0.296	15.671 ± 0.269
Tempered Noise	✗	1.032 ± 0.012	0.058 ± 0.001	0.638 ± 0.002	16.051 ± 0.123	19.627 ± 0.101
Target Score	✓ BDC	1.064 ± 0.369	0.010 ± 0.004	0.402 ± 0.029	7.797 ± 3.990	12.451 ± 5.417
Tempered Noise	✓ BDC	1.228 ± 0.401	0.056 ± 0.029	0.572 ± 0.055	12.598 ± 4.155	17.679 ± 4.178
Target Score	✓ systematic	1.098 ± 0.418	$\mathbf{0.007 \pm 0.005}$	$\mathbf{0.372 \pm 0.020}$	$\mathbf{6.256 \pm 3.960}$	$\mathbf{11.265 \pm 5.629}$
Tempered Noise	✓ systematic	$\mathbf{0.926 \pm 0.248}$	0.027 ± 0.011	0.512 ± 0.017	9.974 ± 1.229	14.045 ± 1.308

F.3 LJ-13 Sampling Task

The Lennard-Jones Potential. The Lennard-Jones (LJ) potential is an intermolecular potential, modelling interactions of non-bonding particles. This system is studied to evaluate the performance of various neural samplers. The energy for the system is based on the interatomic distance between the particles is given by:

$$\mathcal{E}^{\mathrm{LJ}}(x) = \frac{\varepsilon}{2\tau} \sum_{ij} \left(\left(\frac{r_m}{d_{ij}} \right)^6 - \left(\frac{r_m}{d_{ij}} \right)^{12} \right) \tag{215}$$

where we denote the Euclidean distance between two particles i and j by $d_{ij} = \|x_i - x_j\|_2$ and r_m, τ, ϵ and c are physical constants. As in Köhler et al. (2020), we also add a harmonic potential to the energy so that $\mathcal{E}^{LJ-system} = \mathcal{E}^{\mathrm{LJ}}(x) + c\mathcal{E}^{\mathrm{osc}}(x)$ The harmonic potential is given by:

$$\mathcal{E}^{\mathrm{osc}}(x) = \frac{1}{2} \sum_i ||x_i - x_{\mathrm{COM}}||^2 \tag{216}$$

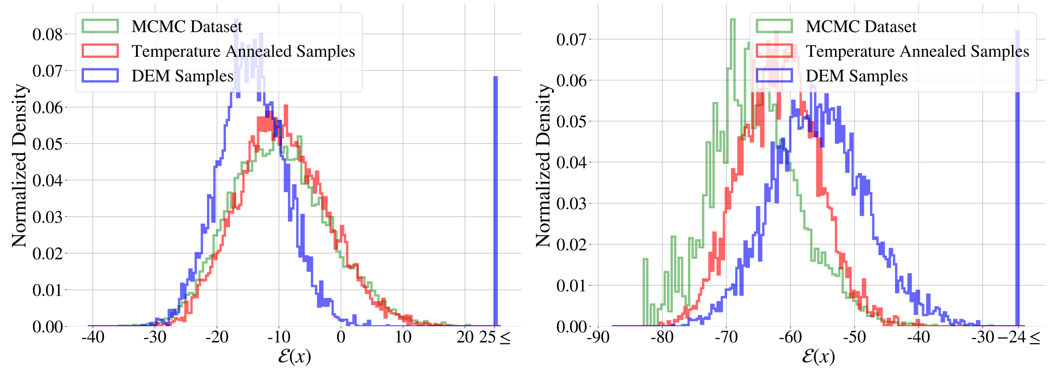

Figure 5: Comparison between the energy distribution of the MCMC dataset, samples generated using a DEM model trained at the target temperature, and samples generated using temperature annealing from a model trained at starting distribution $T = 2$. **Left:** the target temperature is 1.5 and temperature annealed samples correspond to tempered noise SDE + FKC and **Right**: the target temperature is 0.8 and temperature annealed samples correspond to target score SDE + FKC.

Table 7: Additional results for LJ-13 at different target temperatures. The model is trained at starting temperature 2.0 and metrics are computed over 3 runs.

Target Temp.	SDE Type	FKC	distance-$\mathcal{W}_2$	Energy-$\mathcal{W}_1$	Energy-$\mathcal{W}_2$
0.9 (β=2.2)	Target Score	✗	0.861 ± 0.014	13.560 ± 0.064	13.662 ± 0.068
		✓	0.861 ± 0.021	4.296 ± 0.217	4.342 ± 0.195
	Tempered Noise	✗	**0.853** ± **0.018**	5.314 ± 0.047	5.350 ± 0.049
		✓	0.863 ± 0.011	**3.948** ± **0.235**	**4.104** ± **0.253**
1.0 (β=2.0)	Target Score	✗	0.796 ± 0.003	12.865 ± 0.077	12.938 ± 0.080
		✓	0.777 ± 0.010	4.009 ± 0.324	4.034 ± 0.342
	Tempered Noise	✗	**0.771** ± **0.009**	4.859 ± 0.085	4.919 ± 0.074
		✓	0.781 ± 0.021	**2.587** ± **0.089**	**2.822** ± **0.103**
1.2 (β=1.67)	Target Score	✗	0.590 ± 0.008	10.224 ± 0.102	10.248 ± 0.098
		✓	0.551 ± 0.002	3.358 ± 0.024	3.363 ± 0.026
	Tempered Noise	✗	0.547 ± 0.005	4.042 ± 0.058	4.092 ± 0.057
		✓	**0.547** ± **0.007**	**0.956** ± **0.223**	**1.154** ± **0.208**

where x_{COM} refers to the center of mass of the system. We set $r_m = 1$, $\tau = 1$, $\varepsilon = 2.0$ and $c = 1.0$.

Training details. All DEM models are trained for 166 epochs on 4 NVIDIA A100 80GB GPUs. For all models, the best checkpoint with the lowest energy-$\mathcal{W}_2$ is used for inference. The model is an EGNN with the same architecture as in Akhound-Sadegh et al. (2024). Similar to Akhound-Sadegh et al. (2024), we use a geometric noise schedule for all experiments. We set $\sigma_{\min} = 0.01$ and $\sigma_{\max} = 4.0$. We clip the score to a maximum norm of 1000 (per particle). For sampling, we use 1000 integration steps with $dt = 0.001$. For inference with FKC, we assume a Gaussian distribution at time $t_{\mathrm{start}} = 0.99$ and start integration with the annealed SDE and resampling at that time. We found that this helps significantly to reduce the variance of the results over different runs. For visualizations in Fig. 5, we selected the best run for all methods for consistency.

In line with previous work, we find the DEM scores are noisy at high times, based on the score of the energy. This can be seen from the score estimator in DEM, which depends on the average gradient direction from a normal distribution sampled around x_t. The variance of this estimate grows with both time and gradient of the energy. This makes DEM style objective significantly easier to train on smooth energies, as quantified by norm of the score of the energy.

Sampling Reference distributions To generate reference distributions from the Lennard-Jones-13 potential we use Pyro Bingham et al. (2018) and a No-U-Turn sampler Hoffman & Gelman (2011) with default arguments. We use 20k warmup steps and collect 20k samples from the 10 chains for each temperature.

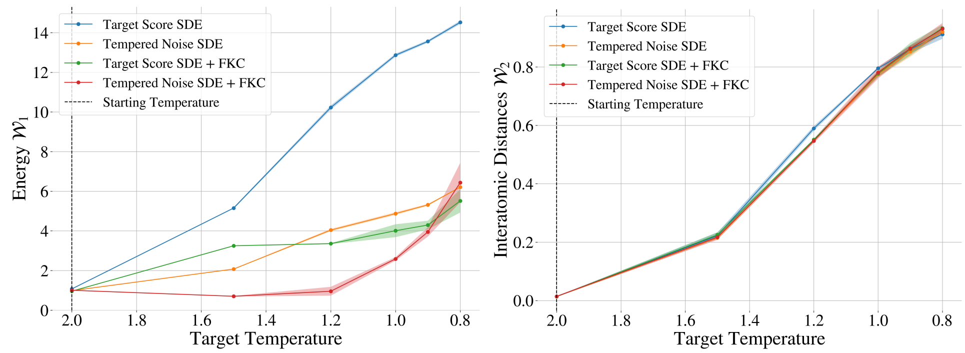

Figure 6: **Left:** 1-Wasserstein between energy distributions and **Right:** 2-Wasserstein between distributions of interatomic distances of MCMC samples from the annealed distribution and generated samples.

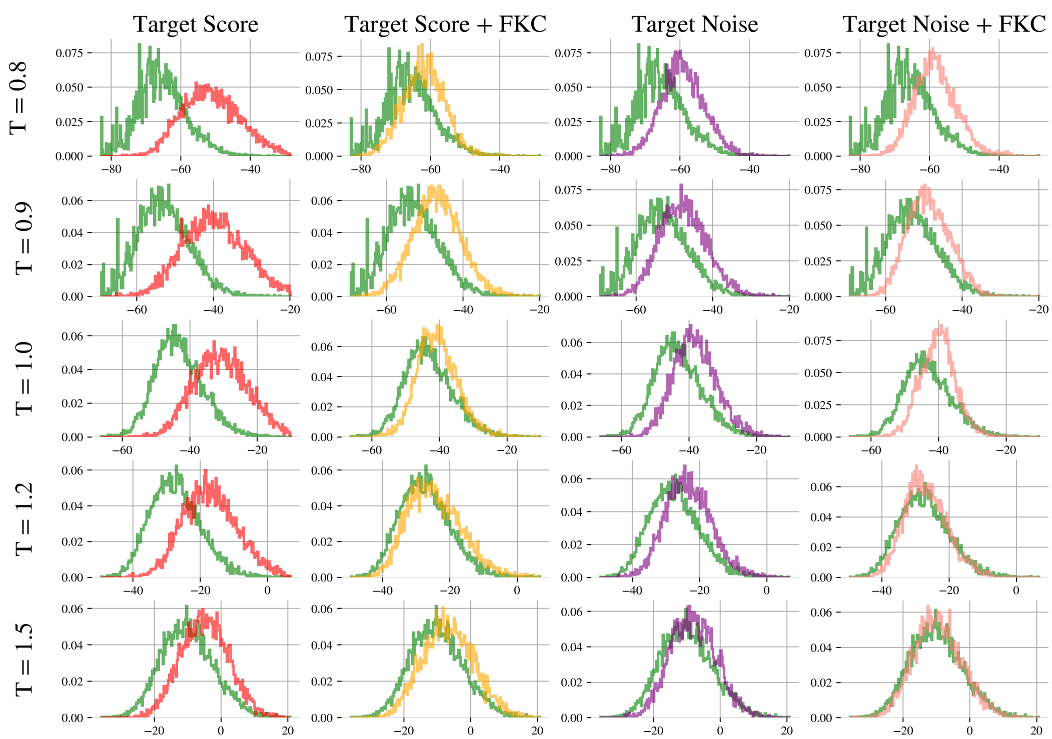

Figure 7: Energy distributions of samples generated with temperature annealing compared to the MCMC samples (in green), at different target temperatures. The starting temperature is $T = 2.0$.

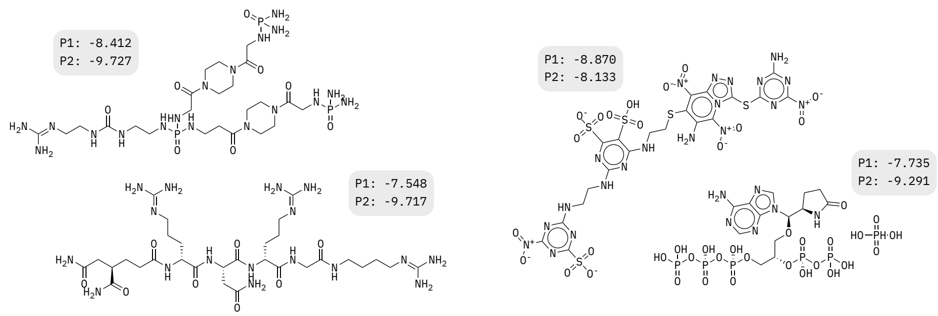

Figure 8: Molecules with best docking scores for binding to ATPA1 ($\mathtt{P}_1$) and CPT2 ($\mathtt{P}_2$) from PoE with FKC (left) and without (right).

Table 8: Multi-property molecule generation results. For a set of two target properties ($\mathtt{P}_1$ and $\mathtt{P}_2$), we take the set of the top-10 best performing molecules as the molecules with the highest $\mathtt{P}_1$*$\mathtt{P}_2$ scores. We report the average properties of the top-10 molecules over five runs and the top-1 molecule overall. We also report the diversity, validity & uniqueness, and quality of all generated molecules, where quality is the percent of molecules that are valid, unique, have a QED ≥ 0.6 and SA < 0.4. For $\beta = 1$, target score and tempering noise match (Prop. 3.3).

$\mathtt{P}_1$ $\mathtt{P}_2$	SDE Type	β	FKC	$\mathtt{P}_1$ top-10 (↑)	$\mathtt{P}_2$ top-10 (↑)	($\mathtt{P}_1$, $\mathtt{P}_2$) top-1 (↑)	Div. (↑)	Val. & Uniq. (↑)	Qual. (↑)
JNK3 GSK3β	Target Score	0.5	✗	$0.212_{\pm 0.016}$	$0.356_{\pm 0.046}$	$(0.500, 0.580)$	$\mathbf{0.910_{\pm 0.000}}$	$0.713_{\pm 0.027}$	$0.127_{\pm 0.015}$
	Tempered Noise		✗	$\mathbf{0.225_{\pm 0.028}}$	$\mathbf{0.385_{\pm 0.042}}$	$(\mathbf{0.440}, \mathbf{0.690})$	$0.909_{\pm 0.001}$	$\mathbf{0.723_{\pm 0.016}}$	$\mathbf{0.134_{\pm 0.006}}$
	—	1.0	✗	$0.289_{\pm 0.022}$	$0.429_{\pm 0.018}$	$(0.470, 0.580)$	$\mathbf{0.898_{\pm 0.002}}$	$\mathbf{0.811_{\pm 0.008}}$	$\mathbf{0.205_{\pm 0.011}}$
	—		✓	$\mathbf{0.342_{\pm 0.029}}$	$\mathbf{0.442_{\pm 0.051}}$	$(\mathbf{0.600}, \mathbf{0.650})$	$0.897_{\pm 0.002}$	$0.804_{\pm 0.015}$	$\mathbf{0.205_{\pm 0.015}}$
	Target Score	1.5	✗	$0.336_{\pm 0.031}$	$0.484_{\pm 0.052}$	$(0.480, 0.780)$	$\mathbf{0.886_{\pm 0.003}}$	$0.816_{\pm 0.013}$	$0.336_{\pm 0.022}$
	Target Score		✓	$0.351_{\pm 0.0340}$	$0.447_{\pm 0.026}$	$(\mathbf{0.590}, \mathbf{0.780})$	$\mathbf{0.886_{\pm 0.003}}$	$0.823_{\pm 0.024}$	$0.356_{\pm 0.037}$
	Tempered Noise		✗	$0.341_{\pm 0.039}$	$0.468_{\pm 0.041}$	$(0.590, 0.560)$	$0.881_{\pm 0.002}$	$0.813_{\pm 0.025}$	$0.352_{\pm 0.012}$
	Tempered Noise		✓	$\mathbf{0.342_{\pm 0.012}}$	$\mathbf{0.502_{\pm 0.034}}$	$(0.500, 0.720)$	$0.882_{\pm 0.002}$	$\mathbf{0.832_{\pm 0.021}}$	$\mathbf{0.360_{\pm 0.021}}$
JNK3 DRD2	Target Score	0.5	✗	$\mathbf{0.090_{\pm 0.018}}$	$\mathbf{0.434_{\pm 0.065}}$	$(0.150, 0.472)$	$\mathbf{0.915_{\pm 0.001}}$	$0.671_{\pm 0.022}$	$0.228_{\pm 0.011}$
	Tempered Score		✗	$0.066_{\pm 0.015}$	$0.571_{\pm 0.187}$	$(\mathbf{0.110}, \mathbf{0.943})$	$0.914_{\pm 0.002}$	$\mathbf{0.678_{\pm 0.0187}}$	$\mathbf{0.236_{\pm 0.020}}$
	—	1.0	✗	$0.087_{\pm 0.028}$	$0.624_{\pm 0.094}$	$(0.100, 0.978)$	$\mathbf{0.903_{\pm 0.001}}$	$0.675_{\pm 0.022}$	$0.241_{\pm 0.010}$
	—		✓	$\mathbf{0.094_{\pm 0.024}}$	$\mathbf{0.635_{\pm 0.067}}$	$(\mathbf{0.413}, \mathbf{0.550})$	$0.899_{\pm 0.002}$	$\mathbf{0.686_{\pm 0.025}}$	$\mathbf{0.263_{\pm 0.023}}$
	Target Score	1.5	✗	$0.136_{\pm 0.046}$	$0.582_{\pm 0.067}$	$(\mathbf{0.490}, \mathbf{0.640})$	$\mathbf{0.886_{\pm 0.003}}$	$0.639_{\pm 0.019}$	$0.241_{\pm 0.017}$
	Target Score		✓	$0.102_{\pm 0.031}$	$0.620_{\pm 0.148}$	$(0.320, 0.541)$	$0.885_{\pm 0.006}$	$0.659_{\pm 0.022}$	$\mathbf{0.274_{\pm 0.028}}$
	Tempered Noise		✗	$0.132_{\pm 0.032}$	$0.550_{\pm 0.036}$	$(0.280, 0.469)$	$0.884_{\pm 0.001}$	$0.650_{\pm 0.021}$	$0.258_{\pm 0.020}$
	Tempered Noise		✓	$\mathbf{0.141_{\pm 0.020}}$	$\mathbf{0.617_{\pm 0.040}}$	$(0.360, 0.655)$	$0.884_{\pm 0.005}$	$\mathbf{0.661_{\pm 0.018}}$	$0.252_{\pm 0.014}$
GSK3β DRD2	Target Score	0.5	✗	$0.146_{\pm 0.034}$	$0.528_{\pm 0.077}$	$(0.051, 0.908)$	$\mathbf{0.914_{\pm 0.001}}$	$\mathbf{0.709_{\pm 0.021}}$	$\mathbf{0.203_{\pm 0.015}}$
	Tempered Score		✗	$\mathbf{0.162_{\pm 0.025}}$	$\mathbf{0.543_{\pm 0.063}}$	$(\mathbf{0.430}, \mathbf{0.965})$	$\mathbf{0.914_{\pm 0.001}}$	$0.697_{\pm 0.013}$	$0.198_{\pm 0.017}$
	—	1.0	✗	$0.202_{\pm 0.023}$	$0.620_{\pm 0.057}$	$(\mathbf{0.660}, \mathbf{0.726})$	$\mathbf{0.908_{\pm 0.002}}$	$0.773_{\pm 0.021}$	$0.238_{\pm 0.021}$
	—		✓	$\mathbf{0.190_{\pm 0.022}}$	$\mathbf{0.666_{\pm 0.093}}$	$(0.240, 0.986)$	$0.907_{\pm 0.002}$	$\mathbf{0.784_{\pm 0.010}}$	$\mathbf{0.254_{\pm 0.019}}$
	Target Score	1.5	✗	$0.240_{\pm 0.030}$	$0.636_{\pm 0.066}$	$(0.350, 0.804)$	$\mathbf{0.894_{\pm 0.002}}$	$0.759_{\pm 0.015}$	$0.290_{\pm 0.016}$
	Target Score		✓	$0.222_{\pm 0.036}$	$0.584_{\pm 0.068}$	$(0.630, 0.580)$	$0.891_{\pm 0.003}$	$0.740_{\pm 0.027}$	$0.283_{\pm 0.020}$
	Tempered Score		✗	$0.228_{\pm 0.016}$	$0.649_{\pm 0.084}$	$(0.550, 0.655)$	$0.884_{\pm 0.002}$	$\mathbf{0.774_{\pm 0.015}}$	$0.303_{\pm 0.012}$
	Tempered Score		✓	$\mathbf{0.266_{\pm 0.061}}$	$\mathbf{0.638_{\pm 0.036}}$	$(\mathbf{0.520}, \mathbf{0.796})$	$0.885_{\pm 0.002}$	$\mathbf{0.774_{\pm 0.017}}$	$\mathbf{0.307_{\pm 0.012}}$

F.4 Molecule generation

Visualizing top-performing molecules We showcase the molecules with the best docking scores from Table 4 in App. F.4.

Metrics In addition to reporting the top-performing molecules, we report the percent of molecules that are valid *and* unique, as well as their diversity (evaluated using Tanimoto distance on Morgan fingerprints (Rogers & Hahn, 2010)) and quality, which is the set of unique and valid molecules that also have a quantitative estimate of drug-likeness (QED) ≥ 0.6. This metric was taken from Lee et al. (2025).

Inference process In practice, we find that the FKC weights have a large variance during molecule generation. This is problematic, as a large number of samples are thrown away. Furthermore, we noted that the score was not always well-conditioned. To ameliorate this, we divided the weights by a set temperature term ($T = 100$) to reduce their variance before resampling, clipped the top 20% to account for any score instabilities, and did early-stopping (only resampled for 70% of the timesteps).

Molecule generation metrics for different SDE types and temperatures In Table 8, we show an ablation over different types of SDEs and β , with and without FKC.

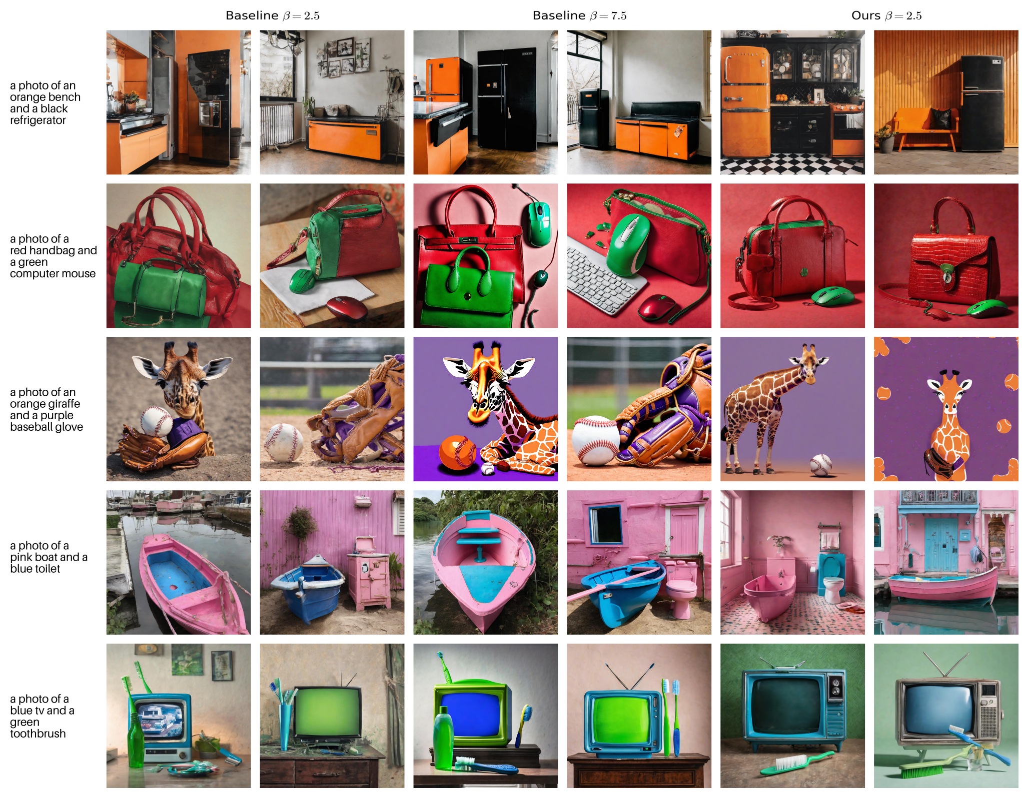

Figure 9: Samples from SDXL

F.5 Additional images for SDXL

We show addition images generated by our method and vanilla SDXL in Fig. 9.