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

Marta Skreta ^{*1 2} Tara Akhound-Sadegh ^{*3 4} Viktor Ohanesian ^{*5} Roberto Bondesan ⁵ Alán Aspuru-Guzik ^{1 2}
Arnaud Doucet ⁶ Rob Brekelmans ² Alexander Tong ^{†7 4 8} Kirill Neklyudov ^{†7 4 9}

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. Our code is available at <https://github.com/martaskrt/fkc-diffusion>.

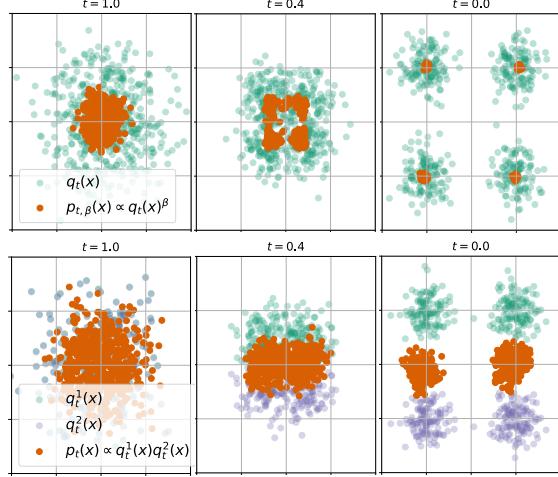


Figure 1. FEYNMAN-KAC CORRECTOR Inference for annealed $p_{t,\beta}(x) \propto q_t(x)^{\beta=10}$ and product $p_t(x) \propto q_t^1(x)q_t^2(x)$ densities.

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.

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 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

^{*}Equal contribution, [†]Equal senior-authorship ¹University of Toronto ²Vector Institute ³McGill University ⁴Mila - Quebec AI Institute ⁵Imperial College London ⁶Google DeepMind ⁷Université de Montréal ⁸Current AT affiliation: Duke University ⁹Institut Courtois. Correspondence to: AT <ayt14@duke.edu>, KN <k.neklyudov@gmail.com>.

Proceedings of the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

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; Thornton et al., 2025). While methods for sampling from mixtures or equiprobable regions of diffusion models have been proposed (Skreta et al., 2025), general solutions to accurately sample 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 for constructing 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).
- For the problem of sampling from an unnormalized density, we demonstrate that FKC allows for sampling from a variety of temperatures without retraining (Sec. 5.2). 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 a lower temperature.
- For pretrained diffusion models we demonstrate that adding FKC terms enhances compositional generation of molecules with multiple properties (Sec. 5.3) and classifier-free guidance for image generation (Sec. 5.1).

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:

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

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 $d\bar{W}_\tau$ 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 ($t = 1 - \tau$),

$$dx_t = (-f_t(x_t) + \sigma_t^2 \nabla \log p_t(x_t))dt + \sigma_t dW_t, \quad (2)$$

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

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

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 ,

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

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

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

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 ,

$$dw_t = \bar{g}_t(x_t)dt \implies \frac{\partial p_t^w(x)}{\partial t} = \bar{g}_t(x)p_t^w(x), \quad (6)$$

where $\bar{g}_t(x) = g_t(x) - \int g_t(x)p_t^w(x)dx$

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

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

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

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

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

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 can account for this difference by sampling

$$i \sim \text{Categorical} \left\{ \frac{\exp(w_T^k)}{\sum_{j=1}^K \exp(w_T^j)} \right\}_{k=1}^K, \quad (9)$$

and returning $x_T^{(i)}$ as an approximate sample from p_T . We discuss more refined resampling techniques in Sec. 4. For estimating the expectation of test functions ϕ , we account for the weights by reweighting a collection of K particles, i.e.,

$$\mathbb{E}_{p_T}[\phi(x)] \approx \sum_{k=1}^K \frac{\exp(w_T^k)}{\sum_j \exp(w_T^j)} \phi(x_T^k). \quad (10)$$

For justification of the validity of this weighting scheme for Feynman-Kac PDEs, see App. A. The expression in Eq. (10) corresponds to Self-Normalized Importance Sampling (SNIS) estimation, which converges to exact expectation estimators when $K \rightarrow \infty$ (e.g. Naesseth et al. (2019)).

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. For instance, it is well-known that the diffusion equation (4) can be simulated using an ODE (Song et al., 2021).

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

$$\frac{\partial p_t(x)}{\partial t} = \frac{\sigma_t^2}{2} \Delta p_t(x) = -\left\langle \nabla, p_t(x) \left(-\frac{\sigma_t^2}{2} \nabla \log p_t(x) \right) \right\rangle \implies dx_t = -\frac{\sigma_t^2}{2} \nabla \log p_t(x_t)dt. \quad (11)$$

The reweighting equation adds an extra dimension to the interplay between different simulation schemes.

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),

$$\frac{\partial p_t(x)}{\partial t} = -\langle \nabla, p_t(x)v_t(x) \rangle = \left(\frac{-1}{p_t(x)} \langle \nabla, p_t(x)v_t(x) \rangle \right) p_t(x) \implies dw_t = (-\langle \nabla, v_t(x_t) \rangle - \langle \nabla \log p_t(x_t), v_t(x_t) \rangle) dt \quad (12)$$

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

$$\frac{\partial p_t(x)}{\partial t} = \frac{\sigma_t^2}{2} \Delta p_t(x) = \frac{\sigma_t^2}{2} p_t(x) (\Delta \log p_t(x) + \|\nabla \log p_t(x)\|^2) \implies dw_t = \frac{\sigma_t^2}{2} (\Delta \log p_t(x_t) + \|\nabla \log p_t(x_t)\|^2) dt \quad (13)$$

In particular, using Eqs. (12) and (13) 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.

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

$$\frac{\partial q_t^i}{\partial t} = -\langle \nabla, q_t^i(-f_t + \sigma_t^2 \nabla \log q_t^i) \rangle + \frac{\sigma_t^2}{2} \Delta q_t^i, \quad (14a)$$

$$dx_t = (-f_t(x_t) + \sigma_t^2 \nabla \log q_t^i(x_t))dt + \sigma_t dW_t, \quad (14b)$$

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:

$$\begin{aligned} \text{Annealed: } & p_{t,\beta}^{\text{anneal}}(x) = \frac{1}{Z_t(\beta)} q_t(x)^\beta \\ \text{Product: } & p_t^{\text{prod}}(x) = \frac{1}{Z_t} q_t^1(x) q_t^2(x) \\ \text{Geometric Avg: } & p_{t,\beta}^{\text{geo}}(x) = \frac{1}{Z_t(\beta)} q_t^1(x)^{1-\beta} q_t^2(x)^\beta. \end{aligned} \quad (15)$$

A common heuristic for sampling from the distributions in the form of Eq. (15) is to simulate according to the score function of the target density. For example, in classifier-free guidance (Ho & Salimans, 2021) we use the score of the ge-

ometric 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

$$dx_t = (-f_t(x_t) + \sigma_t^2 \nabla \log p_{t,\beta}^{\text{geo}}(x_t))dt + \sigma_t dW_t. \quad (16)$$

However, despite the similarity to Eq. (2), this heuristic does not sample from the prescribed marginals (including the final distribution), 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

$$\frac{\partial p_{t,\beta}^{\text{geo}}(x)}{\partial t} = \frac{\partial}{\partial t} \frac{1}{Z_t(\beta)} q_t^1(x)^{(1-\beta)} q_t^2(x)^\beta. \quad (17)$$

Expanding and using our expressions for the Fokker-Planck equation of q_t^i in (14), we proceed to locate terms corresponding to the simulation of an SDE with the drift $v_t(x_t) = -f_t(x_t) + \sigma_t^2 \nabla \log p_{t,\beta}^{\text{geo}}(x_t)$. Collecting all remaining terms of PDE (17) into weights $\bar{g}_t(x_t)$ 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 Sec. 4

$$\frac{\partial p_{t,\beta}^{\text{geo}}}{\partial t} = -\left\langle \nabla, p_{t,\beta}^{\text{geo}} v_t \right\rangle + \frac{\sigma_t^2}{2} \Delta p_{t,\beta}^{\text{geo}} + p_{t,\beta}^{\text{geo}} \bar{g}_t. \quad (18)$$

Conversion Rules To facilitate the construction of 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.

Computational Considerations Our recipe above can yield many different weighted PDEs for a given sequence of target distributions. In practice, we would like our simulation scheme to closely approximate the intermediate targets distributions to limit the need for correction. On the other hand, for computational efficiency, we hope to obtain weights which avoid expensive divergence $\langle \nabla, v_t(x) \rangle$ or Laplacian terms $\langle \nabla, \nabla \log q_t^i(x_t) \rangle$. Remarkably, for linear drift functions $f_t(x)$ commonly used in diffusion models (Song et al., 2021), we find that simulating according to the common heuristic in Eq. (16) yields a Feynman-Kac PDE whose weights can be estimated with no additional overhead. We focus on these schemes in our examples.

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 ,

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

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., 2024a).

For our purposes, we will view CFG as 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 (14). The weighted SDE corresponding to the geometric average of the marginals $p_{t,\beta}^{\text{geo}}(x) \propto q_t^1(x)^{1-\beta} q_t^2(x)^\beta$ is*

$$\begin{aligned} dx_t &= \sigma_t^2 ((1 - \beta) \nabla \log q_t^1(x_t) + \beta \nabla \log q_t^2(x_t)) dt \\ &\quad - f_t(x_t) dt + \sigma_t dW_t, \\ dw_t &= \frac{\sigma_t^2}{2} \beta (\beta - 1) \|\nabla \log q_t^1(x_t) - \nabla \log q_t^2(x_t)\|^2 dt. \end{aligned} \quad (19)$$

In Prop. D.3, we provide a more general formulation of this proposition outlining a continuous family of weighted SDEs sampling from the geometric average $p_{t,\beta}^{\text{geo}}(x) \propto q_t^1(x)^{1-\beta} q_t^2(x)^\beta$. As a further example, we combine CFG with a product of experts in Prop. D.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

$$p_{t,\beta}^{\text{anneal}}(x) = q_t(x)^\beta / Z_t(\beta). \quad (20)$$

For $\beta > 1$, this can be used to generate samples from modes or high-probability regions of given models (Karczewski et al., 2025), while in Sec. 5.2 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 (14). Sampling from the annealed marginals $p_{t,\beta}^{\text{anneal}}(x) \propto q_t(x)^\beta$, $\beta > 0$ can be performed by simulating the following weighted SDE*

$$\begin{aligned} dx_t &= (-f_t(x_t) + \eta \sigma_t^2 \nabla \log q_t(x_t)) dt + \zeta \sigma_t dW_t, \\ dw_t &= (\beta - 1) \left(\langle \nabla, f_t(x_t) \rangle + \frac{\sigma_t^2}{2} \beta \|\nabla \log q_t(x_t)\|^2 \right) dt, \end{aligned}$$

with the coefficients (for $(\beta + (1 - \beta)2a)/\beta \geq 0$)

$$\eta = \beta + (1 - \beta)a, \quad \zeta = \sqrt{(\beta + (1 - \beta)2a)/\beta}. \quad (21)$$

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

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

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.

See Prop. D.1 for proof, and note that linear drifts $f_t(x)$ will lead to constant divergence terms which cancel upon reweighting in (9) and (10). 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,

$$dx_t = (-f_t(x_t) + \beta \sigma_t^2 \nabla \log q_t(x_t))dt + \sigma_t dW_t. \quad (22)$$

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

$$dx_t = (-f_t(x_t) + \frac{\beta + 1}{2} \sigma_t^2 \nabla \log q_t(x_t))dt + \frac{\sigma_t}{\sqrt{\beta}} dW_t. \quad (23)$$

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

$$p_t^{\text{prod}}(x) = q_t^1(x)q_t^2(x)/Z_t. \quad (24)$$

For conditional generative models, the product of densities can describe samples satisfying several conditions. For example, in image generation, we could use $q(x | \text{"horse"})q(x | \text{"a sandy beach"})$ to generate images of “a horse on a sandy beach” (Du et al., 2023). In Sec. 5.3, we demonstrate that the PoE target can be used to improve molecule generations which satisfy multiple

conditions simultaneously.

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

$$\nabla \log p_t^{\text{prod}}(x) = \nabla \log q_t^1(x_t) + \nabla \log q_t^2(x_t), \quad (25)$$

In the following proposition, we further combine these rules with the annealing procedure to present the weighted SDE that samples from the marginals $p_{t,\beta}^{\text{prod}}(x) \propto (q_t^1(x)q_t^2(x))^\beta$, with $\beta > 0$ is

$$\begin{aligned} dx_t &= \sigma_t^2 \eta (\nabla \log q_t^1(x_t) + \nabla \log q_t^2(x_t))dt \\ &\quad - f_t(x_t)dt + \zeta \sigma_t dW_t, \end{aligned} \quad (26)$$

$$\begin{aligned} dw_t &= \beta(\beta - 1) \frac{\sigma_t^2}{2} \|\nabla \log q_t^1(x_t) + \nabla \log q_t^2(x_t)\|^2 dt \\ &\quad + \beta \sigma_t^2 \langle \nabla \log q_t^1(x_t), \nabla \log q_t^2(x_t) \rangle dt \\ &\quad + (2\beta - 1) \langle \nabla, f_t(x_t) \rangle dt, \end{aligned} \quad (27)$$

with the coefficients (for $(\beta + (1 - \beta)2a)/\beta \geq 0$)

$$\eta = \beta + (1 - \beta)a, \quad \zeta = \sqrt{(\beta + (1 - \beta)2a)/\beta}. \quad (28)$$

See proof in Prop. D.2. Again, note that for linear drifts, the divergence term $\langle \nabla, f_t(x) \rangle$ is constant and can be ignored. Further, for $\beta = 1$, the first term in the weight evolution vanishes to leave only the inner product of score vectors. Similarly to Eqs. (22) and (23) for annealing, we have the *target score* SDE ($a = 0, \eta = \beta, \zeta = 1$) and the *tempered noise* SDE ($a = 1/2, \eta = (\beta + 1)/2, \zeta = 1/\sqrt{\beta}$).

More generally, we derive the weighted SDE that samples from $p_{t,\beta}(x) \propto \prod_i q_t^i(x)^{\beta_i}$, i.e. the weighted product of marginal densities $q_t^i(x)$ for arbitrary number of diffusion models (see Prop. D.5).

3.5. Reward-tilted Target Density

Finally, our framework can easily incorporate a reward function $r(x)$ defined on the state-space at inference time. Namely, we assume that the function $\exp(\beta_t r(x))$ is normalizable and consider the reward-tilted density $p_t^{\text{reward}}(x) \propto q_t(x) \exp(\beta_t r(x))$. Despite its similarity to the product of densities, this case is different as we do not assume $\exp(\beta_t r(x))$ changes according to the diffusion process.

Proposition 3.4 (Reward-tilted Target + FKC). *Consider a diffusion model $q_t(x)$ defined via (14). Sampling from the reward-tilted marginals $p_t^{\text{reward}}(x) \propto q_t(x) \exp(\beta_t r(x))$ is performed by the following weighted SDE*

$$dx_t = \sigma_t^2 (\nabla \log q_t(x_t) + \frac{\beta_t}{2} \nabla r(x_t)) dt - f_t(x_t) dt + \sigma_t dW_t, \quad (29)$$

$$dw_t = \frac{\partial \beta_t}{\partial t} r(x_t) dt - \langle \beta_t \nabla r(x_t), f_t(x_t) \rangle dt + \left\langle \beta_t \nabla r(x_t), \frac{\sigma_t^2}{2} \nabla \log q_t(x_t) \right\rangle dt. \quad (30)$$

See proof in Prop. D.6. Here, the weights increase when the vector field of the diffusion models aligns with the gradient of the reward function.

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 (10), 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), App. A), we can leverage SMC techniques which reweight particles along our trajectories.

In practice, we find that resampling only over an ‘active interval’ $t \in [t_{\min}, t_{\max}]$ is 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 might 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 two terms can intuitively be seen to measure the inflow and outflow of probability due to jumps.

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

Proposition 4.1. *For a given g_t in Eq. (6), define the jump process rate and transition as*

$$\lambda_t(x) = (g_t(x) - \mathbb{E}_{p_t}[g_t])^- \quad (31a)$$

$$J_t(y|x) = \frac{(g_t(y) - \mathbb{E}_{p_t}[g_t])^+ p_t(y)}{\int (g_t(z) - \mathbb{E}_{p_t}[g_t])^+ p_t(z) dz} \quad (31b)$$

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

$$\frac{\partial p_t^{\text{jump}}(x)}{\partial t} = \frac{\partial p_t^w(x)}{\partial t} = p_t(x) (g_t(x) - \mathbb{E}_{p_t}[g_t])^- \quad (32)$$

which matches Eq. (6).

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

$$x_{t+dt} = \begin{cases} x_t & \text{w.p. } 1 - \lambda_t(x_t)dt + o(dt) \\ \sim J_t(y|x_t) & \text{w.p. } \lambda_t(x_t)dt + o(dt). \end{cases} \quad (33)$$

We expect this process to improve the sample population in efficient fashion, 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. (33) directly, one can also adopt an implementation based on birth-death ‘exponential clocks’ (BDC, Del Moral (2013, Ch. 5.3-4), see App. B.3).

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

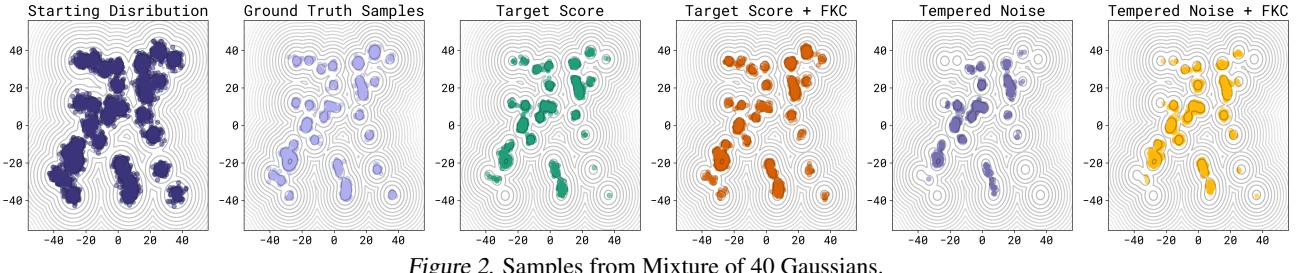


Figure 2. Samples from Mixture of 40 Gaussians.



Figure 3. Samples from EDM2+CFG (top), EDM2+FKC (bottom).

5. Empirical Study

In 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. While we show results for BDC sampling in App. F.2 Table A1, we proceed with systematic resampling throughout the remainder of our experiments.

5.1. Image Generation with EDM2

In this section, we study the effect of FKC resampling for image generation in RGB pixel space, using CFG with an EDM2-XS model trained on ImageNet-512 (Karras et al., 2024b). In particular, we test whether resampling to more closely match the intermediate geometric average distributions translates to improvement in two downstream image quality metrics: CLIP Score (Radford et al., 2021) and ImageReward (Xu et al., 2024). CLIP Score measures the cosine similarity between the image and text prompt embeddings; ImageReward assigns a score that reflects human preferences (aesthetic quality and prompt adherence).

For a fixed simulation scheme, we compare the effect of adding FKC resampling (✓) versus the standard baseline without resampling (✗). We report results across various simulation parameters, namely the number of sampling steps N and churn parameter γ (which controls how the SDE integration scheme adds noise). For FKC, we additionally sweep over the batch size or number of particles K , whereas $K = 1$ corresponds to the no-resampling baseline (✗). To calculate metrics on a single image for FKC, we resample from among K particles according to the weights since the last resampling step. Note that we often observe that final-step images from a single batch with FKC are nearly identical visually due to weight degeneracy.

In Table 2, we compare the quantitative performance of our

Table 2. Comparison of EDM2+FKC (✓) with EDM2+CFG (✗) for image generation using EDM2. We sweep over noise level (γ) and steps (N). For all metrics, we report CLIP and ImageReward (IR) scores averaged over 10,000 images.

FKC	γ	N	CLIP (\uparrow)	IR (\uparrow)	FKC	γ	N	CLIP (\uparrow)	IR (\uparrow)
✗	10	32	28.74	-0.25	✗	40	16	28.67	-0.30
✓	10	32	28.97	0.03	✓	40	16	29.12	-0.01
✗	40	32	28.75	-0.24	✗	40	32	28.75	-0.24
✓	40	32	29.00	0.04	✓	40	32	29.14	0.05
✗	80	32	28.75	-0.24	✗	40	64	28.81	-0.19
✓	80	32	28.99	0.04	✓	40	64	29.12	0.07

FKC resampling against vanilla CFG. We find that adding FKC (✓) improves performance in both ImageReward and CLIP score, indicating both higher prompt adherence and aesthetically better images. While this comes at the cost of extra computation due to $K > 1$, we find that FKC demonstrates benefits even for $K = 2$, with $K = 8$ performing the best (Table A3). Qualitative results in Fig. 3 further support the finding that FKC can improve image quality.

In App. F.5, we provide an additional analysis on using FKC with latent diffusion image models.

5.2. Samplers from the Boltzmann Density

As described in the 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 these settings, 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 $\beta = T_L/T_S$ 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 T_L . 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 A1.

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

Table 3. 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	✖ 0.189 \pm 0.002	14.730 \pm 0.029	15.556 \pm 0.045	
	Tempered Noise	✓ 0.048 \pm 0.019	6.252 \pm 2.710	6.356 \pm 2.673	
	DEM	—	0.103 \pm 0.001	9.794 \pm 0.100	9.804 \pm 0.101
1.5 $(\beta = 1.33)$	Target Score	✖ 0.168 \pm 0.009	5.340 \pm 0.054	6.210 \pm 0.254	
	Tempered Noise	✓ 0.083 \pm 0.003	3.366 \pm 0.083	3.386 \pm 0.090	
	DEM	—	0.095 \pm 0.006	2.154 \pm 0.048	3.920 \pm 0.258

Matching (DEM) model (Akhound-Sadegh et al., 2024) at this base temperature and perform temperature-annealed inference to lower temperatures. In Table 3 and A2 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 (App. F.3). We find that tempered noise+FKC performs best at higher temperatures. However, at lower temperatures, the target score SDE+FKC performs best. Both methods outperform DEM directly trained at the lower temperature for temperatures $T_S \in [2.0, 0.8]$ (Fig. 4). We find DEM is qualitatively easier to learn at higher temperatures requiring much less tuning compared to lower temperatures (Fig. A1). This makes the train-then-anneal approach attractive in this setting. For extended results and discussion see App. F.

5.3. Multi-Target Structure-Based Drug Design

We apply FKC to the setting of structure-based drug design (SBDD), where the goal is to design molecules (or ligands) using the three-dimensional structure of a biological target—typically a protein—as a guide (Anderson, 2003). The ligands are then evaluated based on how well they fit into the protein’s binding site. We focus on dual-target drug design, where a molecule should interact with two proteins simultaneously. Dual-target drug design has become increasingly investigated for targeting complex disease pathways such as in various cancers and neurodegeneration (Ramsay et al., 2018), as well as for diminishing drug resistance mechanisms (Yang et al., 2024).

We investigate the performance of PoE using both target score and tempered noise SDEs at various β , with (✓) and without (✖) FKC. Ligand performance is determined by docking scores to each protein target using AutoDock Vina (Eberhardt et al., 2021). We evaluate 100 protein pairs and average our results over tasks. We sampled 5 molecule sizes from the original training set from Guan et al. (2023): {15, 19, 23, 27, 35} generating 32 molecules per size. We

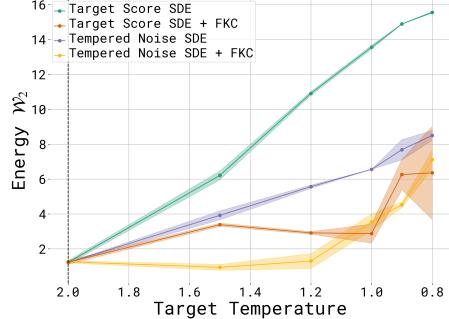


Figure 4. 2-Wasserstein between energy distributions of MCMC samples from the annealed target distribution and our methods at different temperatures. Note the training temperature $T_L = 2$.

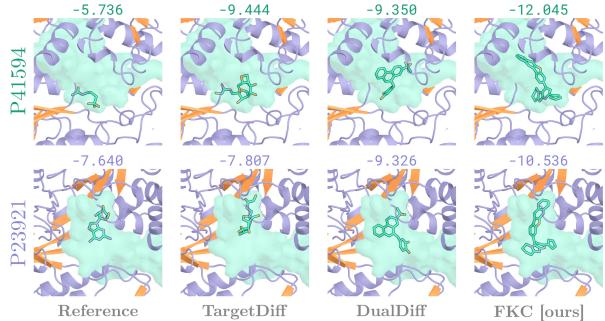


Figure 5. Molecules generated from our method (target score SDE with $\beta = 2.0$ and FKC resampling) and baselines in the binding pockets of two proteins: GRM5 (top row, UniProt ID P41594) and RRM1 (bottom row, UniProt ID P23921). Docking scores for each molecule and target are above each image; lower docking scores are better. Here, we display molecules with the best docking scores that have a QED ≥ 0.4 ; more generations are in App. F.6. The binding pocket is shaded in light green.

showcase our best results in Table 4 and the full ablation in App. F.6. We evaluate the generated molecules on their docking scores to a protein pair, P_1 and P_2 . We report the average of docking score products for each target, as well as the average maximum docking score for a pair. Lower docking scores are better, and so lower maximum docking scores indicate the molecule is better at binding to *both* targets. We compute the percentage of molecules that have better docking scores than known binders, as well as the number of valid and unique molecules generated, their diversity, their drug-likeness (QED (Bickerton et al., 2012)), and their synthetic accessibility (SA (Ertl & Schuffenhauer, 2009)).

We find that the target noise SDE at $\beta > 0.5$ generates molecules with better average docking scores for each of the target proteins compared with both baselines DualDiff (Zhou et al., 2024) and TargetDiff (Guan et al., 2023). When we incorporate FKC, the average docking scores improve further. In Fig. A7, we observe a positive correlation between the FKC weights and docking scores. There is a slight sacrifice in terms of diversity and uniqueness when resampling with FKC, although this is a common trade-off for an increase in quality. Notably, our method achieves

Table 4. Docking scores of generated ligands for 100 protein target pairs (P_1, P_2). We generate 32 ligands for 5 molecule lengths for each protein pair using the Target Score SDE. Lower docking scores are better. Values are reported as averages over all generated molecules in each run. "Better than ref." is the percentage of ligands with better docking scores than known reference molecules for *both* targets (the mean docking score for the reference molecules is -7.915 ± 2.841). We also report the diversity, validity & uniqueness, SA score, and QED. ¹TargetDiff from Guan et al. (2023), ²DualDiff from Zhou et al. (2024).

	$(P_1 * P_2) (\uparrow)$	$\max(P_1, P_2) (\downarrow)$	$P_1 (\downarrow)$	$P_2 (\downarrow)$	Better than ref. (\uparrow)	Div. (\uparrow)	Val. & Uniq. (\uparrow)	SA (\downarrow)	QED (\uparrow)	
P_1 only ¹	59.355 ± 30.169	-6.961 ± 2.774	-8.090 ± 1.783	-7.213 ± 2.746	0.321 ± 0.371	0.886 ± 0.013	0.918 ± 0.107	0.588 ± 0.086	0.531 ± 0.150	
β	F_{FKC}	$(P_1 * P_2) (\uparrow)$	$\max(P_1, P_2) (\downarrow)$	$P_1 (\downarrow)$	$P_2 (\downarrow)$	Better than ref. (\uparrow)	Div. (\uparrow)	Val. & Uniq. (\uparrow)	SA (\downarrow)	QED (\uparrow)
0.5	✖ ²	64.554 ± 28.225	-7.030 ± 2.556	-7.950 ± 2.212	-8.028 ± 2.154	0.306 ± 0.346	0.883 ± 0.012	0.943 ± 0.124	0.609 ± 0.084	0.575 ± 0.134
	✓	66.380 ± 35.747	-6.966 ± 3.291	-8.089 ± 2.832	-8.098 ± 2.638	0.341 ± 0.377	0.870 ± 0.021	0.951 ± 0.096	0.596 ± 0.094	0.587 ± 0.129
1.0	✖	68.851 ± 30.153	-7.256 ± 2.622	-8.206 ± 2.385	-8.287 ± 2.123	0.363 ± 0.375	0.880 ± 0.013	0.964 ± 0.100	0.611 ± 0.090	0.589 ± 0.126
	✓	76.036 ± 33.835	-7.649 ± 2.605	-8.658 ± 2.347	-8.660 ± 2.349	0.434 ± 0.416	0.844 ± 0.029	0.939 ± 0.106	0.627 ± 0.095	0.591 ± 0.128
2.0	✖	71.186 ± 30.799	-7.421 ± 2.497	-8.365 ± 2.336	-8.401 ± 2.051	0.383 ± 0.389	0.877 ± 0.015	0.961 ± 0.115	0.642 ± 0.086	0.594 ± 0.124
	✓	77.271 ± 34.268	-7.720 ± 2.562	-8.682 ± 2.488	-8.735 ± 2.187	0.450 ± 0.438	0.806 ± 0.048	0.862 ± 0.174	0.641 ± 0.112	0.592 ± 0.146

the lowest maximum docking score, meaning that generated ligands are able to better bind to both proteins (on average across tasks). Our method also generates the highest fraction of molecules that are better than known binders (reference molecules), which could motivate using our model in *de novo* drug design settings (the mean docking score of reference molecules is -7.915 ± 2.841). We visualize ligands for a sample target pair in Fig. 5 and Fig. A6.

In App. F.7, we further investigate the utility of PoE in generating molecule SMILES using a latent diffusion model, and show that FKC resampling improves generation for small molecules satisfying multiple functional properties.

6. Related Work

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

For compositional generation, Du et al. (2023) learn an energy-based score function and use the energy within MCMC procedures. Thornton et al. (2025) improve training of the energy-based score function by distilling an unconditional score model, where the resulting energy can be used for SMC resampling from annealed or product densities.

Within the context of diffusion samplers from Boltzmann densities, Phillips et al. (2024) consider SMC for energy-based score parameterizations. Chen et al. (2025b); 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., 2025; Máté & Fleuret, 2023; Tian et al., 2024; Fan et al., 2024; Maurais & Marzouk, 2024; Vargas et al., 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. Bradley & Nakkiran (2024) interpret standard CFG SDE simulation (19) as a predictor-corrector where the corrector targets a different guidance or geometric mixture weight $\beta' = \frac{1}{2}(1 + \beta)$. Our resampling correctors are instead tailored to the original guidance weight β .

Finally, SMC methods have recently been extended to discrete diffusion models (Singhal et al., 2025; Li et al., 2024; Uehara et al., 2025; Lee et al., 2025a), where the approach of Lee et al. (2025a) is analogous to FKC for discrete settings.

7. Conclusion

In this work, we proposed FEYNMAN-KAC CORRECTORS, an array of tools allowing for 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 and 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 (Prop. D.6) and for a time-dependent annealing schedule β_t (Prop. C.6), where the log-density terms needed for weights can be estimated using methods from Skreta et al. (2025).

Impact Statement

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

Acknowledgments

This project was partially sponsored by Google through the Google & Mila projects program. The authors acknowledge funding from UNIQUE, CIFAR, NSERC, Intel, and Samsung. The research was enabled in part by computational resources provided by the Digital Research Alliance of Canada (<https://alliancecan.ca>), Mila (<https://mila.quebec>), the Acceleration Consortium (<https://acceleration.utoronto.ca/>), and NVIDIA. KN was supported by IVADO and Institut Courtois. MS thanks Ella Rajaonson for assistance with docking visualizations, as well as Austin Cheng and Cher-Tian Ser for providing feedback on molecule generation.

References

- Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, pp. 1–3, 2024.
- Akhound-Sadegh, T., Rector-Brooks, J., Bose, J., Mittal, S., Lemos, P., Liu, C.-H., Sendera, M., Ravanbakhsh, S., Gidel, G., Bengio, Y., et al. Iterated denoising energy matching for sampling from Boltzmann densities. In *International Conference on Machine Learning*, 2024.
- Albergo, M. S. and Vanden-Eijnden, E. Nets: A non-equilibrium transport sampler. *arXiv preprint arXiv:2410.02711*, 2024.
- Anderson, A. C. The process of structure-based drug design. *Chemistry & Biology*, 10(9):787–797, 2003.
- Angeli, L. *Interacting particle approximations of Feynman-Kac measures for continuous-time jump processes*. PhD thesis, University of Warwick, 2020.
- Angeli, L., Grosskinsky, S., Johansen, A. M., and Pizzoferrato, A. Rare event simulation for stochastic dynamics in continuous time. *Journal of Statistical Physics*, 176(5): 1185–1210, 2019.
- Arbel, M., Matthews, A., and Doucet, A. Annealed flow transport Monte Carlo. In *International Conference on Machine Learning*, 2021.
- Bickerton, G. R., Paolini, G. V., Besnard, J., Muresan, S., and Hopkins, A. L. Quantifying the chemical beauty of drugs. *Nature Chemistry*, 4(2):90–98, 2012.
- Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P. A., Horsfall, P., and Goodman, N. D. Pyro: Deep universal probabilistic programming. *J. Mach. Learn. Res.*, 20: 28:1–28:6, 2019.
- Bradley, A. and Nakkiran, P. Classifier-free guidance is a predictor-corrector. *arXiv preprint arXiv:2408.09000*, 2024.
- Cardoso, G. V., El Idrissi, Y. J., Le Corff, S., and Moulines, E. Monte Carlo guided diffusion for Bayesian linear inverse problems. In *International Conference on Learning Representations*, 2024.
- Chang, J. and Ye, J. C. Ldmol: Text-conditioned molecule diffusion model leveraging chemically informative latent space. *arXiv preprint arXiv:2405.17829*, 2024.
- Chemseddine, J., Wald, C., Duong, R., and Steidl, G. Neural sampling from Boltzmann densities: Fisher-Rao curves in the Wasserstein geometry. In *International Conference on Learning Representations*, 2025.
- Chen, H., Ren, Y., Min, M. R., Ying, L., and Izzo, Z. Solving inverse problems via diffusion-based priors: An approximation-free ensemble sampling approach. *arXiv preprint arXiv:2506.03979*, 2025a.
- Chen, J., Richter, L., Berner, J., Blessing, D., Neumann, G., and Anandkumar, A. Sequential controlled Langevin diffusions. In *International Conference on Machine Learning*, 2025b.
- Chizat, L., Peyré, G., Schmitzer, B., and Vialard, F.-X. An interpolating distance between optimal transport and Fisher–Rao metrics. *Foundations of Computational Mathematics*, 18:1–44, 2018.
- Crooks, G. E. *Excursions in Statistical Dynamics*. University of California, Berkeley, 1999.
- Davis, M. H. 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.
- De Bortoli, V., Hutchinson, M., Wirnsberger, P., and Doucet, A. Target score matching. *arXiv preprint arXiv:2402.08667*, 2024.
- Del Moral, P. *Mean Field Simulation for Monte Carlo Integration*. Chapman and Hall, CRC press, 2013.

- Dou, Z. and Song, Y. Diffusion posterior sampling for linear inverse problem solving: A filtering perspective. In *International Conference on Learning Representations*, 2024.
- Douc, R. and Cappé, O. Comparison of resampling schemes for particle filtering. In *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, pp. 64–69, 2005.
- Du, Y. and Kaelbling, L. Compositional generative modeling: A single model is not all you need. 2024.
- Du, Y., Durkan, C., Strudel, R., Tenenbaum, J. B., Dieleman, S., Fergus, R., Sohl-Dickstein, J., Doucet, A., and Grathwohl, W. S. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and MCMC. In *International Conference on Machine Learning*, 2023.
- Eberhardt, J., Santos-Martins, D., Tillack, A. F., and Forli, S. 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.
- Ertl, P. and Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics*, 1:1–11, 2009.
- Ethier, S. N. and Kurtz, T. G. *Markov Processes: Characterization and Convergence*. John Wiley & Sons, 2009.
- Fan, M., Zhou, R., Tian, C., and Qian, X. Path-guided particle-based sampling. In *International Conference on Machine Learning*, 2024.
- Gardiner, C. *Stochastic Methods*. Springer, 2009.
- Ghosh, D., Hajishirzi, H., and Schmidt, L. Geneval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, 2023.
- Guan, J., Qian, W. W., Peng, X., Su, Y., Peng, J., and Ma, J. 3d equivariant diffusion for target-aware molecule generation and affinity prediction. *arXiv preprint arXiv:2303.03543*, 2023.
- Ho, J. and Salimans, T. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.
- Hoffman, M. D. and Gelman, A. The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 2014.
- Holderrieth, P., Havasi, M., Yim, J., Shaul, N., Gat, I., Jaakkola, T., Karrer, B., Chen, R. T., and Lipman, Y. Generator matching: Generative modeling with arbitrary Markov processes. In *International Conference on Learning Representations*, 2025.
- Huang, K., Fu, T., Gao, W., Zhao, Y., Roohani, Y., Leskovec, J., Coley, C. W., Xiao, C., Sun, J., and Zitnik, M. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *Proceedings of Neural Information Processing Systems, NeurIPS Datasets and Benchmarks*, 2021.
- Jarzynski, C. Equilibrium free-energy differences from nonequilibrium measurements: A master-equation approach. *Physical Review E*, 56(5):5018, 1997.
- Karczewski, R., Heinonen, M., and Garg, V. Diffusion models as cartoonists! the curious case of high density regions. In *International Conference on Learning Representations*, 2025.
- Karras, T., Aittala, M., Kynkänniemi, T., Lehtinen, J., Aila, T., and Laine, S. Guiding a diffusion model with a bad version of itself. In *Neural Information Processing Systems*, 2024a.
- Karras, T., Aittala, M., Lehtinen, J., Hellsten, J., Aila, T., and Laine, S. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24174–24184, 2024b.
- Kim, S., Kim, M., and Park, D. Test-time alignment of diffusion models without reward over-optimization. In *International Conference on Learning Representations*, 2025.
- Köhler, J., Klein, L., and Noé, F. Equivariant flows: exact likelihood generative learning for symmetric densities. In *International Conference on Machine Learning*, 2020.
- Kondratyev, S., Monsaingeon, L., and Vorotnikov, D. A new optimal transport distance on the space of finite Radon measures. *Adv. Differential Equations*, 2016.
- Lee, C. K., Jeha, P., Frellsen, J., Lio, P., Albergo, M. S., and Vargas, F. Debiasing guidance for discrete diffusion with sequential Monte Carlo. *arXiv preprint arXiv:2502.06079*, 2025a.
- Lee, S., Kreis, K., Veccham, S. P., Liu, M., Reidenbach, D., Peng, Y., Paliwal, S., Nie, W., and Vahdat, A. Genmol: A drug discovery generalist with discrete diffusion. 2025b.
- Lelièvre, T., Rousset, M., and Stoltz, G. *Free Energy Computations: A Mathematical Perspective*. World Scientific, 2010.
- Li, X., Zhao, Y., Wang, C., Scalia, G., Eraslan, G., Nair, S., Biancalani, T., Ji, S., Regev, A., Levine, S., et al.

- Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. *arXiv preprint arXiv:2408.08252*, 2024.
- Liero, M., Mielke, A., and Savaré, G. Optimal entropy-transport problems and a new Hellinger–Kantorovich distance between positive measures. *Inventiones Mathematicae*, 211(3):969–1117, 2018.
- Liu, N., Li, S., Du, Y., Torralba, A., and Tenenbaum, J. B. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, pp. 423–439. Springer, 2022.
- Lu, Y., Lu, J., and Nolen, J. Accelerating Langevin sampling with birth-death. *arXiv preprint arXiv:1905.09863*, 2019.
- Máté, B. and Fleuret, F. Learning interpolations between Boltzmann densities. *Transactions on Machine Learning Research*, 2023.
- Maurais, A. and Marzouk, Y. Sampling in unit time with kernel Fisher-Rao flow. In *Forty-first International Conference on Machine Learning*, 2024.
- Midgley, L. I., Stumper, V., Simm, G. N., Schölkopf, B., and Hernández-Lobato, J. M. Flow annealed importance sampling bootstrap. *International Conference on Learning Representations*, 2023.
- Naesseth, C. A., Lindsten, F., Schön, T. B., et al. Elements of sequential Monte Carlo. *Foundations and Trends® in Machine Learning*, 12(3):307–392, 2019.
- Neal, R. M. Annealed importance sampling. *Statistics and Computing*, 11:125–139, 2001.
- OuYang, R., Qiang, B., and Hernández-Lobato, J. M. BNEM: A Boltzmann sampler based on bootstrapped noised energy matching. *arXiv preprint arXiv:2409.09787*, 2024.
- Phillips, A., Dau, H.-D., Hutchinson, M. J., De Bortoli, V., Deligiannidis, G., and Doucet, A. Particle denoising diffusion sampler. In *International Conference on Machine Learning*, 2024.
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.
- Ramsay, R. R., Popovic-Nikolic, M. R., Nikolic, K., Uliassi, E., and Bolognesi, M. L. A perspective on multi-target drug discovery and design for complex diseases. *Clinical and Translational Medicine*, 7:1–14, 2018.
- Richter, L. and Berner, J. Improved sampling via learned diffusions. In *International Conference on Learning Representations*, 2024.
- Rogers, D. and Hahn, M. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010.
- Rousset, M. On the control of an interacting particle estimation of Schrödinger ground states. *SIAM Journal on Mathematical Analysis*, 38(3):824–844, 2006.
- Rousset, M. and Stoltz, G. Equilibrium sampling from nonequilibrium dynamics. *Journal of Statistical Physics*, 123:1251–1272, 2006.
- Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, 2022.
- Sahoo, S. S., Arriola, M., Gokaslan, A., Marroquin, E. M., Rush, A. M., Schiff, Y., Chiu, J. T., and Kuleshov, V. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, 2024.
- Salimans, T. and Ho, J. Should EBMs model the energy or the score? In *Energy Based Models Workshop-ICLR 2021*, 2021.
- Singhal, R., Horvitz, Z., Teehan, R., Ren, M., Yu, Z., McKown, K., and Ranganath, R. A general framework for inference-time scaling and steering of diffusion models. *arXiv preprint arXiv:2501.06848*, 2025.
- Skreta, M., Atanackovic, L., Bose, A. J., Tong, A., and Neklyudov, K. The superposition of diffusion models using the Itô density estimator. In *International Conference on Learning Representations*, 2025.
- Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.
- Thornton, J., Béthune, L., ZHANG, R., Bradley, A., Nakki- ran, P., and Zhai, S. Controlled generation with distilled diffusion energy models and sequential Monte Carlo. In *International Conference on Artificial Intelligence and Statistics*, 2025.
- Tian, Y., Panda, N., and Lin, Y. T. Liouville flow importance sampler. *International Conference on Machine Learning*, 2024.

- Uehara, M., Zhao, Y., Biancalani, T., and Levine, S. Understanding reinforcement learning-based fine-tuning of diffusion models: A tutorial and review. *arXiv preprint arXiv:2407.13734*, 2024.
- Uehara, M., Zhao, Y., Wang, C., Li, X., Regev, A., Levine, S., and Biancalani, T. Inference-time alignment in diffusion models with reward-guided generation: Tutorial and review. *arXiv preprint arXiv:2501.09685*, 2025.
- Vaikuntanathan, S. and Jarzynski, C. Escorted free energy simulations: Improving convergence by reducing dissipation. *Physical Review Letters*, 100(19):190601, 2008.
- Vaikuntanathan, S. and Jarzynski, C. Escorted free energy simulations. *The Journal of Chemical Physics*, 134(5), 2011.
- Vargas, F., Grathwohl, W. S., and Doucet, A. Denoising diffusion samplers. In *International Conference on Learning Representations*, 2023.
- Vargas, F., Padhy, S., Blessing, D., and Nusken, N. Transport meets variational inference: Controlled Monte Carlo diffusions. In *International Conference on Learning Representations*, 2024.
- Wang, H., Skreta, M., Ser, C.-T., Gao, W., Kong, L., Strieth-Kalthoff, F., Duan, C., Zhuang, Y., Yu, Y., Zhu, Y., et al. Efficient evolutionary search over chemical space with large language models. In *International Conference on Learning Representations*, 2025.
- Woo, D. and Ahn, S. Iterated energy-based flow matching for sampling from Boltzmann densities. *arXiv preprint arXiv:2408.16249*, 2024.
- Wu, L., Trippe, B., Naesseth, C., Blei, D., and Cunningham, J. P. Practical and asymptotically exact conditional sampling in diffusion models. In *Advances in Neural Information Processing Systems*, 2024.
- Xu, J., Liu, X., Wu, Y., Tong, Y., Li, Q., Ding, M., Tang, J., and Dong, Y. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36, 2024.
- Yang, Y., Mou, Y., Wan, L.-X., Zhu, S., Wang, G., Gao, H., and Liu, B. Rethinking therapeutic strategies of dual-target drugs: An update on pharmacological small-molecule compounds in cancer. *Medicinal Research Reviews*, 44(6):2600–2623, 2024.
- Zhang, Q. and Chen, Y. Path integral sampler: A stochastic control approach for sampling. In *International Conference on Learning Representations*, 2022.
- Zhou, X., Guan, J., Zhang, Y., Peng, X., Wang, L., and Ma, J. Reprogramming pretrained target-specific diffusion models for dual-target drug design. In *Neural Information Processing Systems*, 2024.

A. Expectation Estimation under Feynman-Kac PDEs

We proceed in two steps, first finding a Kolmogorov backward equation corresponding to evolution under a weighted Feynman-Kac SDE. We then use this identity to derive the expectation estimator in Eq. (10). Throughout, we consider the evolution of density p_t defined via the following Feynman-Kac PDE,

$$\frac{\partial}{\partial t} p_t(x_t) = -\langle \nabla, p_t(x_t) v_t(x_t) \rangle + \frac{\sigma_t^2}{2} \Delta p_t(x_t) + p_t(x_t) \left(g_t(x_t) - \int g_t(x_t) p_t(x_t) dx_t \right) \quad (34)$$

Our proof follows similar derivations as in Lelièvre et al. (2010, Prop 4.1, Ch. 4.1.4.3) (see also (Vaikuntanathan & Jarzynski, 2008; 2011) and references therein), where the authors are interested in sampling from a sequence of unnormalized distributions \tilde{p}_t specified via a time-varying energy or Hamiltonian. The proofs often rely on Langevin dynamics that leave p_t invariant. We adopt a similar proof technique, but focus directly on simulation with arbitrary v_t, g_t derived via our methods in Sec. 3.

Proposition A.1. *For a bounded test function $\phi : \mathcal{X} \rightarrow \mathbb{R}$ and p_t satisfying Eq. (34), we have*

$$\mathbb{E}_{p_T(x_T)}[\phi(x_T)] = \frac{1}{Z_T} \mathbb{E}\left[e^{\int_0^T g_s(x_s) ds} \phi(x_T)\right] \quad (35)$$

where $dx_t = v_t(x_t) dt + \sigma_t dW_t$, $x_0 \sim p_0$

where Z_T is a normalization constant independent of x . Eq. (35) which suggests that the self-normalized importance sampling approximation in Eq. (10) is consistent as $K \rightarrow \infty$.

Proof. The proof proceeds in three steps, delineated with bold paragraph headers. We first derive the backward Kolmogorov equation for appropriate functions, then specify the evolution of the Feynman-Kac PDE for the unnormalized density, before combining these results to prove the result in Prop. A.1.

Backward PDE: For a given test function $\phi(x)$, consider defining the following function

$$\Phi_T(x, t) = \mathbb{E}\left[e^{\int_t^T g_s(x_s) ds} \phi(x_T) \mid x_t = x\right], \quad \Phi_T(x, T) = \phi(x) \quad (36)$$

where expectations are taken under the evolution of the SDE $dx_t = v_t(x_t) dt + \sigma_t dW_t$.

In particular, for $\tau > t$, we have

$$\Phi_T(x, t) = \mathbb{E}\left[e^{\int_t^\tau g_s(x_s) ds} e^{\int_\tau^T g_s(x_s) ds} \phi(x_T) \mid x_t = x\right] = \mathbb{E}\left[e^{\int_t^\tau g_s(x_s) ds} \Phi_T(x_\tau, \tau) \mid x_t = x\right] \quad (37)$$

We will leverage this identity to derive a PDE which $\Phi_T(x, t)$ must satisfy. Note, to link $\Phi_T(x, t)$ and (the expected value of) $\Phi_T(x_\tau, \tau)$, we should account for the weights $e^{\int_t^\tau g_s(x_s) ds}$. Thus, we apply Ito's product rule and Ito's lemma to capture how $e^{\int_t^\tau g_s(x_s) ds} \Phi_T(x_\tau, \tau)$ evolves with τ ,

$$d\left(e^{\int_t^\tau g_s(x_s) ds} \Phi_T(x_\tau, \tau)\right) = e^{\int_t^\tau g_s(x_s) ds} d\Phi_T(x_\tau, \tau) + \Phi_T(x_\tau, \tau) de^{\int_t^\tau g_s(x_s) ds} + d\langle \Phi_T(x_\tau, \tau), e^{\int_t^\tau g_s(x_s) ds} \rangle \quad (38)$$

In the final term, $e^{\int_t^\tau g_s(x_s) ds}$ is non-stochastic and, assuming it has finite variation, the term $d\langle \Phi_T(x, t), e^{\int_t^\tau g_s(x_s) ds} \rangle$ vanishes. We can use Ito's lemma to expand $d\Phi_T(x_\tau, \tau)$ and simple differentiation for $de^{\int_t^\tau g_s(x_s) ds}$,

$$\begin{aligned} d\left(e^{\int_t^\tau g_s(x_s) ds} \Phi_T(x_\tau, \tau)\right) &= e^{\int_t^\tau g_s(x_s) ds} \left(\frac{\partial \Phi_T(x_\tau, \tau)}{\partial \tau} + \langle v_\tau(x_\tau), \nabla \Phi_T(x_\tau, \tau) \rangle + \frac{\sigma_\tau^2}{2} \Delta \Phi_T(x_\tau, \tau) \right) d\tau \\ &\quad + e^{\int_t^\tau g_s(x_s) ds} \sigma_\tau \langle \nabla \Phi_T(x_\tau, \tau), dW_t \rangle + \Phi_T(x_\tau, \tau) e^{\int_t^\tau g_s(x_s) ds} (g_\tau(x_\tau)) d\tau \end{aligned} \quad (39)$$

$$\begin{aligned} &= e^{\int_t^\tau g_s(x_s) ds} \left(\frac{\partial \Phi_T(x_\tau, \tau)}{\partial t} + \langle v_t(x), \nabla \Phi_T(x_\tau, \tau) \rangle + \frac{\sigma_t^2}{2} \Delta \Phi_T(x_\tau, \tau) + \Phi_T(x_\tau, \tau) g_t(x) \right) dt \\ &\quad + e^{\int_t^\tau g_s(x_s) ds} \sigma_t \langle \nabla \Phi_T(x_\tau, \tau), dW_t \rangle \end{aligned} \quad (40)$$

Integrating Eq. (40) $\tau = t$ to $\tau = T$ and taking expectations under the simulated process from initial point $x_t = x$, the

stochastic term vanishes and we obtain

$$\begin{aligned} & \mathbb{E}\left[e^{\int_t^T g_s(x_s)ds} \Phi_T(x_T, T) \mid x_t = x\right] - \mathbb{E}\left[e^{\int_t^t g_s(x_s)ds} \Phi_T(x, t) \mid x_t = x\right] \\ &= \mathbb{E}\left[\int_{\tau=t}^T e^{\int_t^\tau g_s(x_s)ds} \left(\frac{\partial \Phi_T(x_\tau, \tau)}{\partial \tau} + \langle v_\tau(x), \nabla \Phi_T(x_\tau, \tau) \rangle + \frac{\sigma_\tau^2}{2} \Delta \Phi_T(x_\tau, \tau) + \Phi_T(x_\tau, \tau) g_\tau(x)\right) d\tau\right] \end{aligned} \quad (41)$$

Finally, we simplify the first line in Eq. (41). Considering the definition and endpoint condition in Eq. (36), we have

$$\begin{aligned} & \mathbb{E}\left[e^{\int_t^T g_s(x_s)ds} \Phi_T(x_T, T) \mid x_t = x\right] - \mathbb{E}\left[e^{\int_t^t g_s(x_s)ds} \Phi_T(x, t) \mid x_t = x\right] \\ &= \mathbb{E}\left[e^{\int_t^T g_s(x_s)ds} \phi(x_T) \mid x_t = x\right] - \Phi_T(x, t) = 0 \end{aligned} \quad (42)$$

by definition in Eq. (36). Since $e^{\int_t^\tau g_s(x_s)ds} > 0$, this implies that the integrand in the second line of Eq. (41) should be zero for any τ . Thus, we obtain a backward PDE which is often used directly in the statement of the Feynman-Kac formula,

$$\frac{\partial \Phi_T(x_\tau, \tau)}{\partial \tau} + \langle v_\tau(x), \nabla \Phi_T(x_\tau, \tau) \rangle + \frac{\sigma_\tau^2}{2} \Delta \Phi_T(x_\tau, \tau) + \Phi_T(x_\tau, \tau) g_\tau(x) = 0 \quad (43)$$

Evolution of Unnormalized Density In practice, we cannot exactly calculate $\int g_t(x_t)p_t(x_t)dx_t$, which appears in the reweighting equation in Eq. (6) (or Eq. (46) below) to ensure normalization. Eventually, we will account for normalization using SNIS as in Eq. (10).

For now, consider the evolution of unnormalized density $\tilde{p}_t(x) = p_t(x)Z_t$ for a particular v_t, σ_t, g_t and some normalization constant Z_t . With foresight, we define

$$\frac{\partial}{\partial t} \tilde{p}_t(x_t) = -\langle \nabla, \tilde{p}_t(x_t)v_t(x_t) \rangle + \frac{\sigma_t^2}{2} \Delta \tilde{p}_t(x_t) + \tilde{p}_t(x_t)g_t(x_t) \quad (44)$$

which we justify by noting that only the reweighting term does not preserve normalization. In particular, let

$$\partial_t \log Z_t := \int p_t(x)g_t(x)dx. \quad (45)$$

which seems to be a natural candidate from inspecting a general, reweighting-only evolution $\partial_t p_t^w(x) = p_t^w(x)(g_t(x) - \int p_t^w(x)g_t(x)dx)$, which implies $\partial_t \log p_t^w(x) = g_t(x) - \int p_t^w(x)g_t(x)dx$. Defining terms such that $\partial_t \log p_t^w(x) = \partial_t \log \tilde{p}_t^w(x) - \partial_t \log Z_t$ yields Eq. (45). We finally confirm that the definitions in Eq. (44) and Eq. (45) are consistent with the original Feynman-Kac PDE,

$$\frac{\partial}{\partial t} p_t(x_t) = -\langle \nabla, p_t(x_t)v_t(x_t) \rangle + \frac{\sigma_t^2}{2} \Delta p_t(x_t) + p_t(x_t) \left(g_t(x_t) - \int g_t(x_t)p_t(x_t)dx_t \right) \quad (46)$$

Namely, since $p_t(x_t) = \tilde{p}_t(x_t)Z_t^{-1}$, the definitions in Eq. (44)-(45) should satisfy

$$\frac{\partial}{\partial t} p_t(x_t) = \frac{\partial}{\partial t} (\tilde{p}_t(x_t)Z_t^{-1}) \quad (47a)$$

$$= Z_t^{-1} \frac{\partial}{\partial t} \tilde{p}_t(x_t) + \tilde{p}_t(x_t)Z_t^{-1} \partial_t \log(Z_t^{-1}) \quad (47b)$$

$$= Z_t^{-1} \frac{\partial}{\partial t} \tilde{p}_t(x_t) - \tilde{p}_t(x_t)Z_t^{-1} \partial_t \log Z_t \quad (47c)$$

$$= Z_t^{-1} \left(-\langle \nabla, \tilde{p}_t(x_t)v_t(x_t) \rangle + \frac{\sigma_t^2}{2} \Delta \tilde{p}_t(x_t) + \tilde{p}_t(x_t)g_t(x_t) \right) - \tilde{p}_t(x_t)Z_t^{-1} \int p_t(x_t)g_t(x_t)dx \quad (47d)$$

Noting that $\nabla_{x_t} Z_t = 0$, we can pull Z_t^{-1} inside differential operators to obtain

$$= -\left\langle \nabla, \frac{\tilde{p}_t(x_t)}{Z_t} v_t(x_t) \right\rangle + \frac{\sigma_t^2}{2} \Delta \frac{\tilde{p}_t(x_t)}{Z_t} + \frac{\tilde{p}_t(x_t)}{Z_t} g_t(x_t) - \frac{\tilde{p}_t(x_t)}{Z_t} \int p_t(x_t)g_t(x_t)dx \quad (47e)$$

$$= -\langle \nabla, p_t(x_t)v_t(x_t) \rangle + \frac{\sigma_t^2}{2} \Delta p_t(x_t) + p_t(x_t) \left(g_t(x_t) - \int p_t(x_t)g_t(x_t)dx \right) \quad (47f)$$

as desired.

Expectation Estimation: Now, we use Eq. (43) to write the total derivative of the following integral under the unnormalized density $\tilde{p}_t(x)$,

$$\frac{d}{dt} \left(\int \Phi_T(x, t) \tilde{p}_t(x) dx \right) = \int \left(\frac{\partial \Phi_T(x, t)}{\partial t} \right) \tilde{p}_t(x) dx + \int \Phi_T(x, t) \left(\frac{\partial \tilde{p}_t(x)}{\partial t} \right) dx \quad (48a)$$

Using Eq. (43) and Eq. (44), we have

$$\begin{aligned} &= \int \left(-\langle v_t(x), \nabla \Phi_T(x, t) \rangle - \frac{\sigma_t^2}{2} \Delta \Phi_T(x, t) - \Phi_T(x, t) g_\tau(x) \right) \tilde{p}_t(x) dx \\ &\quad + \int \Phi_T(x, t) \left(-\langle \nabla, \tilde{p}_t(x) v_t(x) \rangle + \frac{\sigma_t^2}{2} \Delta \tilde{p}_t(x) + \tilde{p}_t(x) g_t(x) \right) dx \end{aligned} \quad (48b)$$

Integrating by parts in the second line, we have

$$= \int \left(-\langle v_t(x), \nabla \Phi_T(x, t) \rangle - \frac{\sigma_t^2}{2} \Delta \Phi_T(x, t) - \Phi_T(x, t) g_\tau(x) \right) \tilde{p}_t(x) dx \quad (48c)$$

$$+ \int \left(\langle v_t(x), \nabla \Phi_T(x, t) \rangle + \frac{\sigma_t^2}{2} \Delta \Phi_T(x, t) + \Phi_T(x, t) g_t(x) \right) \tilde{p}_t(x) dx$$

$$= 0 \quad (48d)$$

Integrating on the interval $t = 0$ to $t = T$, we obtain

$$\int \Phi_T(x_T, T) \tilde{p}_T(x_T) dx_T - \int \Phi_T(x_0, 0) \tilde{p}_0(x_0) dx_0 = \int_0^T \frac{d}{dt} \left(\int \Phi_T(x, t) \tilde{p}_t(x) dx \right) dt = 0 \quad (49)$$

Thus, we can set these two quantities equal to each other. Using the identity $\tilde{p}_t(x) = p_t(x) Z_t$ and assuming we initialize simulation with normalized $p_0(x) = \tilde{p}_0(x)$ with $Z_0 = 1$, we can finally use the definitions in Eq. (36) (namely $\Phi_t(x_T, T) = \phi(x_T)$) to write

$$\int \Phi_T(x_0, 0) \tilde{p}_0(x_0) dx_0 = \int \Phi_T(x_T, T) \tilde{p}_T(x_T) dx_T \quad (50)$$

$$Z_0 \int \left(\mathbb{E}[e^{\int_0^T g_s(x_s) ds} \phi(x_T) | x_0] \right) p_0(x_0) dx_0 = Z_T \int \phi(x_T) p_T(x_T) dx_T \quad (51)$$

$$\frac{1}{Z_T} \mathbb{E} \left[e^{\int_0^T g_s(x_s) ds} \phi(x_T) \right] = \mathbb{E}_{p_T(x_T)}[\phi(x_T)] \quad (52)$$

which is the desired identity. In practice, we could estimate $Z_T \approx \frac{1}{K} \sum_{k=1}^K e^{\int_0^T g_s(x_s^{(k)}) ds} = \frac{1}{K} \sum_{k=1}^K e^{w_T^{(k)}}$ and $\mathbb{E}[e^{\int_0^T g_s(x_s) ds} \phi(x_T)] \approx \frac{1}{K} \sum_{k=1}^K e^{w_T^{(k)}} \phi(x_T^{(k)})$, which yields Eq. (10). \square

Note that our choice of upper limit T in Φ_T was arbitrary, suggesting that we could repurpose the same reasoning for estimating expectations at intermediate t from initialization at time 0. This suggests that our samples are properly weighted for estimating expectations and normalization constants Z_t for intermediate p_t (Naesseth et al., 2019).

Similarly, changing the lower limit of integration from $t = 0$ to intermediate t , the analogue of Eq. (51) suggests estimating expectations using $Z_t \mathbb{E}_{p_t}[\Phi_T(x_t, t)] = Z_T \mathbb{E}_{p_T}[\phi(x_T)]$. Given properly-weighted particle approximations of p_t, Z_t , we can continue calculating the appropriate weights along the trajectory to estimate Z_T or terminal expectations under p_T . These arguments can be similarly adapted to justify SMC resampling at intermediate steps, as we do in practice (Sec. 4).

B. Feynman-Kac Processes

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

$$\begin{aligned}
 \textbf{Flow:} \quad & \mathcal{L}_t^{(v)}[\phi](x) = \langle \nabla \phi(x), v_t(x) \rangle & \mathcal{L}_t^{*(v)}[p_t](x) = -\langle \nabla, p_t(x) v_t(x) \rangle \\
 \textbf{Diffusion:} \quad & \mathcal{L}_t^{(\sigma)}[\phi](x) = \frac{\sigma_t^2}{2} \Delta \phi(x) & \mathcal{L}_t^{*(\sigma)}[p_t](x) = \frac{\sigma_t^2}{2} p_t(x) \\
 \textbf{Reweighting:} \quad & \mathcal{L}_t^{(g,p)}[\phi](x) = \phi_t(x) \left(g_t(x) - \int g_t(y) p_t(y) dy \right) & \mathcal{L}_t^{*(g)}[p_t](x) = p_t(x) \left(g_t(x) - \int g_t(y) p_t(y) dy \right)
 \end{aligned} \tag{53}$$

Proof. The proofs for flows and diffusions follow using integration by parts, with proofs found in, for example Holderrieth et al. (2025, Sec. A.5). For the reweighting generator, we have

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

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 . \square

B.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. (2025, A.5.3)) and derive its adjoint generator.

Lemma B.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 $J_t(y|x)$,*

$$\textbf{Jump Process:} \quad \mathcal{J}_t^{(W)}[\phi](x) := \int (\phi(y) - \phi(x)) \lambda_t(x) J_t(y|x) dy \tag{54a}$$

$$\mathcal{J}_t^{*(W)}[p_t](x) = \left(\int \lambda_t(y) J_t(x|y) p_t(y) dy \right) - p_t(x) \lambda_t(x) \tag{54b}$$

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

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

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

Reweighting \rightarrow 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. (53)).

While Del Moral (2013); Angeli (2020) emphasize the freedom of choice in such generators,¹ Sec. 4 of (Angeli et al., 2019)

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

argues for a particular choice to reduce the expected number of resampling events. To define this process, consider the following thresholding operations,

$$(u)^- := \max(0, -u) \quad (u)^+ := \max(0, u), \quad \text{which satisfy: } (u)^+ - (u)^- = u. \quad (55)$$

We can now define the Markov generator using

$$W_t(x, y) = \lambda_t(x) J_t(y|x) \quad \lambda_t(x) := (g_t(x) - \mathbb{E}_{p_t}[g_t])^- \quad J_t(y|x) := \frac{(g_t(y) - \mathbb{E}_{p_t}[g_t])^+ p_t(y)}{\int (g_t(z) - \mathbb{E}_{p_t}[g_t])^+ p_t(z) dz} \quad (56)$$

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 B.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. (56) satisfies $\mathcal{J}_t^{*(W)}[p_t](x) = \mathcal{L}_t^{*(g)}[p_t](x)$. More explicitly, we have*

$$\mathcal{L}_t^{*(g)}[p_t](x) = \mathcal{J}_t^{*(W)}[p_t](x) \quad (57)$$

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

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

$$\mathcal{J}_t^{*(W)}[p_t](x) = \left(\int \lambda_t(y) J_t(x|y) p_t(y) dy \right) - p_t(x) \lambda_t(x) \quad (58a)$$

$$= \left(\int (g_t(y) - \mathbb{E}_{p_t}[g_t])^- \frac{(g_t(x) - \mathbb{E}_{p_t}[g_t])^+ p_t(x)}{\int (g_t(z) - \mathbb{E}_{p_t}[g_t])^+ p_t(z) dz} p_t(y) dy \right) - p_t(x) (g_t(x) - \mathbb{E}_{p_t}[g_t])^- \quad (58b)$$

$$= \left(\int (g_t(y) - \mathbb{E}_{p_t}[g_t])^- p_t(y) dy \right) \left(\frac{(g_t(x) - \mathbb{E}_{p_t}[g_t])^+ p_t(x)}{\int (g_t(z) - \mathbb{E}_{p_t}[g_t])^+ p_t(z) dz} \right) - p_t(x) (g_t(x) - \mathbb{E}_{p_t}[g_t])^- \quad (58c)$$

$$= \left(\frac{\int (g_t(y) - \mathbb{E}_{p_t}[g_t])^- p_t(y) dy}{\int (g_t(z) - \mathbb{E}_{p_t}[g_t])^+ p_t(z) dz} \right) p_t(x) (g_t(x) - \mathbb{E}_{p_t}[g_t])^+ - p_t(x) (g_t(x) - \mathbb{E}_{p_t}[g_t])^- \quad (58d)$$

Using Eq. (55), note that

$$\int (g_t(z) - \mathbb{E}_{p_t}[g_t])^+ p_t(z) dz - \int dp_t(z) (g_t(z) - \mathbb{E}_{p_t}[g_t])^- = \int (g_t(z) - \mathbb{E}_{p_t}[g_t]) p_t(z) dz = 0 \quad (59)$$

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. (58d) is zero.

Case 1 ($\lambda_t(x) = 0 \forall z \in \text{supp}(p_t)$): Note that $\int (g_t(z) - \mathbb{E}_{p_t}[g_t])^- p_t(z) dz = 0$ 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

$$\begin{aligned} \mathcal{L}_t^{*(g)}[p_t](x) &= p_t(x) \left(g_t(x) - \int g_t(x) p_t(x) dx \right) = p_t(x) (\mathbb{E}_{p_t}[g_t] - \mathbb{E}_{p_t}[g_t]) = 0 \\ \mathcal{J}_t^{*(W)}[p_t](x) &= \int 0 \cdot 0 p_t(y) dy - p_t(x) \cdot 0 = 0 \end{aligned} \quad (60)$$

and thus Eq. (57) holds, as desired.

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

In this case, we can conclude using Eq. (59) that $\frac{\int dp_t(z) (g_t(z) - \mathbb{E}_{p_t}[g_t])^-}{\int dp_t(z) (g_t(z) - \mathbb{E}_{p_t}[g_t])^+} = 1$.

Continuing from Eq. (58d)

$$\mathcal{J}_t^{*(W)}[p_t](x) = \left(\frac{\int(g_t(y) - \mathbb{E}_{p_t}[g_t])^- p_t(y) dy}{\int(g_t(z) - \mathbb{E}_{p_t}[g_t])^+ p_t(z) dz} \right) p_t(x) \left(g_t(x) - \mathbb{E}_{p_t}[g_t] \right)^+ - p_t(x) \left(g_t(x) - \mathbb{E}_{p_t}[g_t] \right)^- \quad (61a)$$

$$= p_t(x) \left(\left(g_t(x) - \mathbb{E}_{p_t}[g_t] \right)^+ - \left(g_t(x) - \mathbb{E}_{p_t}[g_t] \right)^- \right) \quad (61b)$$

$$= p_t(x)(g_t(x) - \mathbb{E}_{p_t}[g_t]) \quad (61c)$$

$$= \mathcal{L}_t^{*(g)}[p_t](x) \quad (61d)$$

as desired. Note that, in the second to last line, we used the identity in Eq. (55) that $(u)^+ - (u)^- = u$. \square

B.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

$$\left(g_t(x^{(k)}) - \mathbb{E}_{p_t}[g_t] \right)^- \approx \left(g_t(x^{(k)}) - \frac{1}{K} \sum_{i=1}^K g_t(x^{(i)}) \right)^-, \quad \left(g_t(x^{(k)}) - \mathbb{E}_{p_t}[g_t] \right)^+ \approx \left(g_t(x^{(k)}) - \frac{1}{K} \sum_{i=1}^K g_t(x^{(i)}) \right)^+$$

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. \(2025\)](#)). With rate $\lambda_t(x) = \left(g_t(x) - \mathbb{E}_{p_t}[g_t] \right)^-$, the particle jumps to a new configuration,

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

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 $\lambda_{t_{\text{last}}:s}^{(k)} \geq \tau^{(k)}$, we sample a transition according the empirical approximaton of $J_t(y|x)$ in [Eq. \(62\)](#). We report results using this scheme in [App. F.2 Table A1](#), but found it to underperform relative to systematic resampling in these initial experiments.

C. Proofs for Table 1

C.1. Annealing

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

$$\frac{\partial q_t(x)}{\partial t} = -\langle \nabla, q_t(x)v_t(x) \rangle. \quad (62)$$

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

$$\frac{\partial}{\partial t} p_{t,\beta}(x) = -\langle \nabla, p_{t,\beta}(x)v_t(x) \rangle + p_{t,\beta}(x)[g_t(x) - \mathbb{E}_{p_{t,\beta}} g_t(x)], \quad (63)$$

$$g_t(x) = -(\beta - 1)\langle \nabla, v_t(x) \rangle. \quad (64)$$

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

$$p_{t,\beta}(x) = \frac{q_t(x)^\beta}{\int dx q_t(x)^\beta}, \quad \frac{\partial}{\partial t} p_{t,\beta}(x) = ? \quad (65)$$

By the straightforward calculations we have

$$\frac{\partial}{\partial t} \log p_{t,\beta} = \beta \frac{\partial}{\partial t} \log q_t - \int dx p_{t,\beta} \beta \frac{\partial}{\partial t} \log q_t \quad (66)$$

$$= -\beta \langle \nabla, v_t \rangle - \beta \langle \nabla \log q_t, v_t \rangle - \int dx p_{t,\beta} [-\beta \langle \nabla, v_t \rangle - \beta \langle \nabla \log q_t, v_t \rangle] \quad (67)$$

$$= -\langle \nabla, v_t \rangle - \langle \nabla \log p_{t,\beta}, v_t \rangle + (1-\beta) \langle \nabla, v_t \rangle - \int dx p_{t,\beta} [-\beta \langle \nabla, v_t \rangle - \langle \nabla \log p_{t,\beta}, v_t \rangle] \quad (68)$$

$$= -\langle \nabla, v_t \rangle - \langle \nabla \log p_{t,\beta}, v_t \rangle + (1-\beta) \langle \nabla, v_t \rangle - \int dx p_{t,\beta} [(1-\beta) \langle \nabla, v_t \rangle]. \quad (69)$$

Thus, we have

$$\frac{\partial}{\partial t} p_{t,\beta}(x) = -\langle \nabla, p_{t,\beta}(x) v_t(x) \rangle + p_{t,\beta}(x) [(1-\beta) \langle \nabla, v_t(x) \rangle - \mathbb{E}_{p_{t,\beta}} (1-\beta) \langle \nabla, v_t(x) \rangle], \quad (70)$$

which can be simulated as

$$dx_t = v_t(x_t) dt, \quad (71)$$

$$dw_t = -(\beta-1) \langle \nabla, v_t(x_t) \rangle dt. \quad (72)$$

□

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

$$\frac{\partial q_t(x)}{\partial t} = -\langle \nabla, q_t(x) v_t(x) \rangle. \quad (73)$$

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

$$\frac{\partial}{\partial t} p_{t,\beta}(x) = -\langle \nabla, p_{t,\beta}(x) \beta v_t(x) \rangle + p_{t,\beta}(x) [g_t(x) - \mathbb{E}_{p_{t,\beta}} g_t(x)], \quad (74)$$

$$g_t(x) = (\beta-1) \langle \nabla \log p_{t,\beta}(x), v_t(x) \rangle. \quad (75)$$

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

$$p_{t,\beta}(x) = \frac{q_t(x)^\beta}{\int dx q_t(x)^\beta}, \quad \frac{\partial}{\partial t} p_{t,\beta}(x) = ? \quad (76)$$

By the straightforward calculations we have

$$\frac{\partial}{\partial t} \log p_{t,\beta} = \beta \frac{\partial}{\partial t} \log q_t - \int dx p_{t,\beta} \beta \frac{\partial}{\partial t} \log q_t \quad (77)$$

$$= -\beta \langle \nabla, v_t \rangle - \beta \langle \nabla \log q_t, v_t \rangle - \int dx p_{t,\beta} [-\beta \langle \nabla, v_t \rangle - \beta \langle \nabla \log q_t, v_t \rangle] \quad (78)$$

$$= -\langle \nabla, \beta v_t \rangle - \langle \nabla \log p_{t,\beta}, v_t \rangle - \int dx p_{t,\beta} [-\beta \langle \nabla, v_t \rangle - \langle \nabla \log p_{t,\beta}, v_t \rangle] \quad (79)$$

$$= -\langle \nabla, \beta v_t \rangle - \langle \nabla \log p_{t,\beta}, \beta v_t \rangle - (1-\beta) \langle \nabla \log p_{t,\beta}, v_t \rangle - \int dx p_{t,\beta} [-(1-\beta) \langle \nabla \log p_{t,\beta}, v_t \rangle]. \quad (80)$$

Thus, we have

$$\frac{\partial}{\partial t} p_{t,\beta}(x) = -\langle \nabla, p_{t,\beta}(x) \beta v_t(x) \rangle + p_{t,\beta}(x) [g_t(x) - \mathbb{E}_{p_{t,\beta}} g_t(x)], \quad (81)$$

$$g_t(x) = -(1-\beta) \langle \nabla \log p_{t,\beta}, v_t \rangle, \quad (82)$$

which can be simulated as

$$dx_t = \beta v_t(x_t) dt, \quad (83)$$

$$dw_t = \beta(\beta - 1) \langle \nabla \log q_t(x_t), v_t(x_t) \rangle dt. \quad (84)$$

□

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

$$\frac{\partial q_t(x)}{\partial t} = \frac{\sigma_t^2}{2} \Delta q_t(x). \quad (85)$$

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

$$\frac{\partial}{\partial t} p_{t,\beta}(x) = \frac{\sigma_t^2}{2} \Delta p_{t,\beta}(x) + p_{t,\beta}(x) [g_t(x) - \mathbb{E}_{p_{t,\beta}} g_t(x)], \quad (86)$$

$$g_t(x) = -\beta(\beta - 1) \frac{\sigma_t^2}{2} \|\nabla \log q_t(x)\|^2. \quad (87)$$

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

$$p_{t,\beta}(x) = \frac{q_t(x)^\beta}{\int dx q_t(x)^\beta}, \quad \frac{\partial}{\partial t} p_{t,\beta}(x) = ? \quad (88)$$

By the straightforward calculations we have

$$\frac{\partial}{\partial t} \log p_{t,\beta} = \beta \frac{\partial}{\partial t} \log q_t - \int dx p_{t,\beta} \beta \frac{\partial}{\partial t} \log q_t \quad (89)$$

$$= \beta \frac{\sigma_t^2}{2} \Delta \log q_t + \beta \frac{\sigma_t^2}{2} \|\nabla \log q_t\|^2 - \int dx p_{t,\beta} \left[\beta \frac{\sigma_t^2}{2} \Delta \log q_t + \beta \frac{\sigma_t^2}{2} \|\nabla \log q_t\|^2 \right] \quad (90)$$

$$= \frac{\sigma_t^2}{2} \Delta \log p_{t,\beta} + \frac{\sigma_t^2}{2\beta} \|\nabla \log p_{t,\beta}\|^2 - \int dx p_{t,\beta} \left[\frac{\sigma_t^2}{2} \Delta \log p_{t,\beta} + \frac{\sigma_t^2}{2\beta} \|\nabla \log p_{t,\beta}\|^2 \right] \quad (91)$$

$$= \frac{\sigma_t^2}{2} \Delta \log p_{t,\beta} + \frac{\sigma_t^2}{2} \|\nabla \log p_{t,\beta}\|^2 - \left(1 - \frac{1}{\beta}\right) \frac{\sigma_t^2}{2} \|\nabla \log p_{t,\beta}\|^2 \quad (92)$$

$$- \int dx p_{t,\beta} \left[-\left(1 - \frac{1}{\beta}\right) \frac{\sigma_t^2}{2} \|\nabla \log p_{t,\beta}\|^2 \right]. \quad (93)$$

Thus, we have

$$\frac{\partial}{\partial t} p_{t,\beta}(x) = \frac{\sigma_t^2}{2} \Delta p_{t,\beta}(x) + p_{t,\beta}(x) [g_t(x) - \mathbb{E}_{p_{t,\beta}} g_t(x)], \quad (94)$$

$$g_t(x) = -\beta(\beta - 1) \frac{\sigma_t^2}{2} \|\nabla \log q_t(x)\|^2, \quad (95)$$

which can be simulated (for $\beta > 0$) as

$$dx_t = \sigma_t dW_t, \quad (96)$$

$$dw_t = -\beta(\beta - 1) \frac{\sigma_t^2}{2} \|\nabla \log q_t(x_t)\|^2 dt. \quad (97)$$

□

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

$$\frac{\partial q_t(x)}{\partial t} = \frac{\sigma_t^2}{2} \Delta q_t(x). \quad (98)$$

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

$$\frac{\partial}{\partial t} p_{t,\beta}(x) = \frac{\sigma_t^2}{2\beta} \Delta p_{t,\beta}(x) + p_{t,\beta}(x) [g_t(x) - \mathbb{E}_{p_{t,\beta}} g_t(x)], \quad (99)$$

$$g_t(x) = (\beta - 1) \frac{\sigma_t^2}{2} \Delta \log q_t(x). \quad (100)$$

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

$$p_{t,\beta}(x) = \frac{q_t(x)^\beta}{\int dx q_t(x)^\beta}, \quad \frac{\partial}{\partial t} p_{t,\beta}(x) = ? \quad (101)$$

By the straightforward calculations we have

$$\frac{\partial}{\partial t} \log p_{t,\beta} = \beta \frac{\partial}{\partial t} \log q_t - \int dx p_{t,\beta} \beta \frac{\partial}{\partial t} \log q_t \quad (102)$$

$$= \beta \frac{\sigma_t^2}{2} \Delta \log q_t + \beta \frac{\sigma_t^2}{2} \|\nabla \log q_t\|^2 - \int dx p_{t,\beta} \left[\beta \frac{\sigma_t^2}{2} \Delta \log q_t + \beta \frac{\sigma_t^2}{2} \|\nabla \log q_t\|^2 \right] \quad (103)$$

$$= \frac{\sigma_t^2}{2} \Delta \log p_{t,\beta} + \frac{\sigma_t^2}{2\beta} \|\nabla \log p_{t,\beta}\|^2 - \int dx p_{t,\beta} \left[\frac{\sigma_t^2}{2} \Delta \log p_{t,\beta} + \frac{\sigma_t^2}{2\beta} \|\nabla \log p_{t,\beta}\|^2 \right] \quad (104)$$

$$= \frac{\sigma_t^2}{2\beta} \Delta \log p_{t,\beta} + \frac{\sigma_t^2}{2\beta} \|\nabla \log p_{t,\beta}\|^2 + \left(1 - \frac{1}{\beta}\right) \frac{\sigma_t^2}{2} \Delta \log p_{t,\beta} \quad (105)$$

$$- \int dx p_{t,\beta} \left[\left(1 - \frac{1}{\beta}\right) \frac{\sigma_t^2}{2} \Delta \log p_{t,\beta} \right]. \quad (106)$$

Thus, we have

$$\frac{\partial}{\partial t} p_{t,\beta}(x) = \frac{\sigma_t^2}{2\beta} \Delta p_{t,\beta}(x) + p_{t,\beta}(x) [g_t(x) - \mathbb{E}_{p_{t,\beta}} g_t(x)], \quad (107)$$

$$g_t(x) = (\beta - 1) \frac{\sigma_t^2}{2} \Delta \log q_t(x), \quad (108)$$

which can be simulated (for $\beta > 0$) as

$$dx_t = \frac{\sigma_t}{\sqrt{\beta}} dW_t, \quad (109)$$

$$dw_t = (\beta - 1) \frac{\sigma_t^2}{2} \Delta \log q_t(x_t) dt. \quad (110)$$

□

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

$$\frac{\partial q_t(x)}{\partial t} = q_t(x) (g_t(x) - \mathbb{E}_{q_t(x)} g_t(x)). \quad (111)$$

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

$$\frac{\partial}{\partial t} p_{t,\beta}(x) = p_{t,\beta} [\beta g_t(x) - \mathbb{E}_{p_{t,\beta}} \beta g_t(x)]. \quad (112)$$

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

$$p_{t,\beta}(x) = \frac{q_t(x)^\beta}{\int dx q_t(x)^\beta}, \quad \frac{\partial}{\partial t} p_{t,\beta}(x) = ? \quad (113)$$

By the straightforward calculations we have

$$\frac{\partial}{\partial t} \log p_{t,\beta} = \beta \frac{\partial}{\partial t} \log q_t - \int dx p_{t,\beta} \beta \frac{\partial}{\partial t} \log q_t \quad (114)$$

$$= \beta(g_t(x) - \mathbb{E}_{q_t(x)} g_t(x)) - \int dx p_{t,\beta} [\beta(g_t(x) - \mathbb{E}_{q_t(x)} g_t(x))] \quad (115)$$

$$= \beta g_t(x) - \int dx p_{t,\beta} \beta g_t(x). \quad (116)$$

Thus, we have

$$\frac{\partial}{\partial t} p_{t,\beta}(x) = p_{t,\beta} [\beta g_t(x) - \mathbb{E}_{p_{t,\beta}} \beta g_t(x)], \quad (117)$$

which can be simulated as

$$dx_t = 0, \quad (118)$$

$$dw_t = \beta g_t(x_t). \quad (119)$$

□

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

$$dx_t = v_{t,\beta}(x_t) + \sigma_{t,\beta} dW_t, \quad (120)$$

$$dw_t = g_{t,\beta}(x_t). \quad (121)$$

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

$$dx_t = v_{t,\beta_t}(x_t) + \sigma_{t,\beta_t} dW_t, \quad (122)$$

$$dw_t = g_{t,\beta_t}(x_t) + \frac{\partial \beta_t}{\partial t} \log q_t(x_t), \quad (123)$$

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

$$\frac{\partial}{\partial t} \log p_{t,\beta} = \beta \frac{\partial}{\partial t} \log q_t - \int dx p_{t,\beta} \left[\beta \frac{\partial}{\partial t} \log q_t \right] \quad (124)$$

$$= -\frac{1}{p_{t,\beta}} \langle \nabla, p_{t,\beta} v_{t,\beta} \rangle + \frac{1}{p_{t,\beta}} \frac{\sigma_{t,\beta}^2}{2} \Delta p_{t,\beta} + (g_{t,\beta} - \mathbb{E}_{p_{t,\beta}} g_{t,\beta}). \quad (125)$$

Thus, for the time-dependent β_t , we have

$$\frac{\partial}{\partial t} \log p_{t,\beta_t} = \beta_t \frac{\partial}{\partial t} \log q_t + \frac{\partial \beta_t}{\partial t} \log q_t - \int dx p_{t,\beta_t} \left[\beta_t \frac{\partial}{\partial t} \log q_t + \frac{\partial \beta_t}{\partial t} \log q_t \right] \quad (126)$$

$$= -\frac{1}{p_{t,\beta_t}} \langle \nabla, p_{t,\beta_t} v_{t,\beta_t} \rangle + \frac{1}{p_{t,\beta_t}} \frac{\sigma_{t,\beta_t}^2}{2} \Delta p_{t,\beta_t} + \left[\left(g_{t,\beta_t} + \frac{\partial \beta_t}{\partial t} \log q_t \right) - \mathbb{E}_{p_{t,\beta_t}} \left(g_{t,\beta_t} + \frac{\partial \beta_t}{\partial t} \log q_t \right) \right]. \quad (127)$$

From which we have the statement of the proposition. □

C.2. Product

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

$$\frac{\partial q_t^1(x)}{\partial t} = -\langle \nabla, q_t^1(x)v_t^1(x) \rangle, \quad \frac{\partial q_t^2(x)}{\partial t} = -\langle \nabla, q_t^2(x)v_t^2(x) \rangle. \quad (128)$$

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

$$\frac{\partial}{\partial t} p_t(x) = -\langle \nabla, p_t(x)(v_t^1(x) + v_t^2(x)) \rangle + p_t(x)(g_t(x) - \mathbb{E}_{p_t(x)} g_t(x)), \quad (129)$$

$$g_t(x) = \langle \nabla \log q_t^1(x), v_t^2(x) \rangle + \langle \nabla \log q_t^2(x), v_t^1(x) \rangle. \quad (130)$$

Proof. For the continuity equations

$$\frac{\partial}{\partial t} q_t^{1,2}(x) = -\langle \nabla, q_t^{1,2}(x)v_t^{1,2}(x) \rangle, \quad (131)$$

we want to find the partial derivative of the annealed density

$$p_t(x) = \frac{q_t^1(x)q_t^2(x)}{\int dx q_t^1(x)q_t^2(x)}, \quad \frac{\partial}{\partial t} p_t(x) = ? \quad (132)$$

By the straightforward calculations we have

$$\frac{\partial}{\partial t} \log p_t = \frac{\partial}{\partial t} \log q_t^1 + \frac{\partial}{\partial t} \log q_t^2 - \int dx p_t \left[\frac{\partial}{\partial t} \log q_t^1 + \frac{\partial}{\partial t} \log q_t^2 \right] \quad (133)$$

$$= -\langle \nabla, v_t^1 + v_t^2 \rangle - \langle \nabla \log q_t^1, v_t^1 \rangle - \langle \nabla \log q_t^2, v_t^2 \rangle - \quad (134)$$

$$- \int dx p_t [-\langle \nabla, v_t^1 + v_t^2 \rangle - \langle \nabla \log q_t^1, v_t^1 \rangle - \langle \nabla \log q_t^2, v_t^2 \rangle] \quad (135)$$

$$= -\langle \nabla, v_t^1 + v_t^2 \rangle - \langle \nabla \log p_t, v_t^1 + v_t^2 \rangle + \langle \nabla \log q_t^1, v_t^2 \rangle + \langle \nabla \log q_t^2, v_t^1 \rangle - \quad (136)$$

$$- \int dx p_t [\langle \nabla \log q_t^1, v_t^2 \rangle + \langle \nabla \log q_t^2, v_t^1 \rangle]. \quad (137)$$

Thus, we have

$$\frac{\partial}{\partial t} p_t(x) = -\langle \nabla, p_t(x)(v_t^1(x) + v_t^2(x)) \rangle + p_t(x)(g_t(x) - \mathbb{E}_{p_t(x)} g_t(x)), \quad (138)$$

$$g_t(x) = \langle \nabla \log q_t^1(x), v_t^2(x) \rangle + \langle \nabla \log q_t^2(x), v_t^1(x) \rangle, \quad (139)$$

which can be simulated as

$$dx_t = (v_t^1(x_t) + v_t^2(x_t))dt, \quad (140)$$

$$dw_t = [\langle \nabla \log q_t^1(x_t), v_t^2(x_t) \rangle + \langle \nabla \log q_t^2(x_t), v_t^1(x_t) \rangle]dt. \quad (141)$$

□

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

$$\frac{\partial q_t^1(x)}{\partial t} = \frac{\sigma_t^2}{2} \Delta q_t^1(x), \quad \frac{\partial q_t^2(x)}{\partial t} = \frac{\sigma_t^2}{2} \Delta q_t^2(x). \quad (142)$$

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

$$\frac{\partial}{\partial t} p_t(x) = \frac{\sigma_t^2}{2} \Delta p_t(x) + p_t(x)(g_t(x) - \mathbb{E}_{p_t(x)} g_t(x)), \quad (143)$$

$$g_t(x) = -\sigma_t^2 \langle \nabla \log q_t^1(x), \nabla \log q_t^2(x) \rangle. \quad (144)$$

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

$$p_t(x) = \frac{q_t^1(x)q_t^2(x)}{\int dx q_t^1(x)q_t^2(x)}, \quad \frac{\partial}{\partial t} p_t(x) = ? \quad (145)$$

By the straightforward calculations we have

$$\frac{\partial}{\partial t} \log p_t = \frac{\partial}{\partial t} \log q_t^1 + \frac{\partial}{\partial t} \log q_t^2 - \int dx p_t \left[\frac{\partial}{\partial t} \log q_t^1 + \frac{\partial}{\partial t} \log q_t^2 \right] \quad (146)$$

$$= \frac{\sigma_t^2}{2} \Delta \log q_t^1 + \frac{\sigma_t^2}{2} \|\nabla \log q_t^1\|^2 + \frac{\sigma_t^2}{2} \Delta \log q_t^2 + \frac{\sigma_t^2}{2} \|\nabla \log q_t^2\|^2 - \quad (147)$$

$$- \int dx p_t \left[\frac{\sigma_t^2}{2} \Delta \log q_t^1 + \frac{\sigma_t^2}{2} \|\nabla \log q_t^1\|^2 + \frac{\sigma_t^2}{2} \Delta \log q_t^2 + \frac{\sigma_t^2}{2} \|\nabla \log q_t^2\|^2 \right] \quad (148)$$

$$= \frac{\sigma_t^2}{2} \Delta \log p_t + \frac{\sigma_t^2}{2} \|\nabla \log p_t\|^2 - \sigma_t^2 \langle \nabla \log q_t^1, \nabla \log q_t^2 \rangle - \int dx p_t [-\sigma_t^2 \langle \nabla \log q_t^1, \nabla \log q_t^2 \rangle]. \quad (149)$$

Thus, we have

$$\frac{\partial}{\partial t} p_t(x) = \frac{\sigma_t^2}{2} \Delta p_t(x) + p_t(x) (g_t(x) - \mathbb{E}_{p_t(x)} g_t(x)), \quad (150)$$

$$g_t(x) = -\sigma_t^2 \langle \nabla \log q_t^1(x), \nabla \log q_t^2(x) \rangle, \quad (151)$$

which can be simulated as

$$dx_t = \sigma_t dW_t, \quad (152)$$

$$dw_t = [-\sigma_t^2 \langle \nabla \log q_t^1(x_t), \nabla \log q_t^2(x_t) \rangle] dt. \quad (153)$$

□

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

$$\frac{\partial q_t^1(x)}{\partial t} = (g_t^1(x) - \mathbb{E}_{q_t^1} g_t^1(x)) q_t^1(x), \quad \frac{\partial q_t^2(x)}{\partial t} = (g_t^2(x) - \mathbb{E}_{q_t^2} g_t^2(x)) q_t^2(x). \quad (154)$$

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

$$\frac{\partial}{\partial t} p_t(x) = p_t(x) (g_t(x) - \mathbb{E}_{p_t(x)} g_t(x)), \quad (155)$$

$$g_t(x) = g_t^1(x) + g_t^2(x), \quad (156)$$

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

$$p_t(x) = \frac{q_t^1(x)q_t^2(x)}{\int dx q_t^1(x)q_t^2(x)}, \quad \frac{\partial}{\partial t} p_t(x) = ? \quad (157)$$

By the straightforward calculations we have

$$\frac{\partial}{\partial t} \log p_t = \frac{\partial}{\partial t} \log q_t^1 + \frac{\partial}{\partial t} \log q_t^2 - \int dx p_t \left[\frac{\partial}{\partial t} \log q_t^1 + \frac{\partial}{\partial t} \log q_t^2 \right] \quad (158)$$

$$= (g_t^1(x) - \mathbb{E}_{q_t^1} g_t^1(x)) + (g_t^2(x) - \mathbb{E}_{q_t^2} g_t^2(x)) - \quad (159)$$

$$- \int dx p_t [(g_t^1(x) - \mathbb{E}_{q_t^1} g_t^1(x)) + (g_t^2(x) - \mathbb{E}_{q_t^2} g_t^2(x))] \quad (160)$$

$$= g_t^1(x) + g_t^2(x) - \int dx p_t [g_t^1(x) + g_t^2(x)]. \quad (161)$$

Thus, we have

$$\frac{\partial}{\partial t} p_t(x) = p_t(x) (g_t(x) - \mathbb{E}_{p_t(x)} g_t(x)), \quad (162)$$

$$g_t(x) = g_t^1(x) + g_t^2(x), \quad (163)$$

which can be simulated as

$$dx_t = 0, \quad (164)$$

$$dw_t = g_t^1(x_t) + g_t^2(x_t). \quad (165)$$

□

D. Proofs of Propositions

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

$$dx_t = (-f_t(x_t) + \sigma_t^2 \nabla \log q_t(x_t)) dt + \sigma_t dW_t, \quad (166)$$

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

$$dx_t = (-f_t(x_t) + (\beta + (1-\beta)a) \sigma_t^2 \nabla \log q_t(x_t)) dt + \sqrt{\frac{\sigma_t^2(\beta + (1-\beta)2a)}{\beta}} dW_t, \quad (167)$$

$$dw_t = \left[(\beta - 1) \langle \nabla, f_t(x_t) \rangle + \frac{1}{2} \sigma_t^2 \beta (\beta - 1) \|\nabla \log q_t(x_t)\|^2 \right] dt, \quad (168)$$

where the parameter a satisfies $0 \leq (\beta + (1-\beta)2a)/\beta$.

Proof. For the following SDE

$$dx_t = (-f_t(x_t) + \sigma_t^2 \nabla \log q_t(x_t)) dt + \sigma_t dW_t, \quad (169)$$

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

$$\frac{\partial q_t}{\partial t} = \langle \nabla, q_t [(1-a)\sigma_t^2 \nabla \log q_t(x_t) + a\sigma_t^2 \nabla \log q_t(x_t)] \rangle + (1-b) \frac{\sigma_t^2}{2} \Delta q_t + b \frac{\sigma_t^2}{2} \Delta q_t. \quad (170)$$

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

$$g_t(x) = (1-a)\sigma_t^2 \beta (\beta - 1) \|\nabla \log q_t(x)\|^2 - a\sigma_t^2 (\beta - 1) \Delta \log q_t(x) + \quad (171)$$

$$+ (\beta - 1) \frac{(1-b)\sigma_t^2}{2} \Delta \log q_t(x_t) - \beta (\beta - 1) \frac{b\sigma_t^2}{2} \|\nabla \log q_t(x_t)\|^2. \quad (172)$$

Let's cancel out the term with the Laplacians, hence, we have $2a = 1 - b$ and

$$g_t(x) = (1-a-b/2)\sigma_t^2 \beta (\beta - 1) \|\nabla \log q_t(x)\|^2 = \frac{1}{2} \sigma_t^2 \beta (\beta - 1) \|\nabla \log q_t(x)\|^2. \quad (173)$$

The PDE for the density is

$$\frac{\partial p_{t,\beta}}{\partial t} = - \langle \nabla, p_{t,\beta} (-f_t + (\beta(1-a) + a)\sigma_t^2 \nabla \log q_t) \rangle + \left(\frac{1-b}{\beta} + b \right) \frac{\sigma_t^2}{2} \Delta p_{t,\beta} + p_{t,\beta} (g_t - \mathbb{E}_{p_{t,\beta}} g_t) \quad (174)$$

$$= - \langle \nabla, p_{t,\beta} (-f_t + (\beta + (1-\beta)a)\sigma_t^2 \nabla \log q_t) \rangle + \frac{\beta + (1-\beta)2a}{\beta} \frac{\sigma_t^2}{2} \Delta p_{t,\beta} + p_{t,\beta} (g_t - \mathbb{E}_{p_{t,\beta}} g_t) \quad (175)$$

This corresponds to the following family of SDEs ($0 \leq \beta + (1 - \beta)2a$)

$$dx_t = (-f_t(x_t) + (\beta + (1 - \beta)a)\sigma_t^2 \nabla \log q_t(x_t))dt + \sqrt{\frac{\sigma_t^2(\beta + (1 - \beta)2a)}{\beta}} dW_t, \quad (176)$$

$$dw_t = \left[(\beta - 1)\langle \nabla, f_t(x_t) \rangle + \frac{1}{2}\sigma_t^2\beta(\beta - 1)\|\nabla \log q_t(x_t)\|^2 \right] dt. \quad (177)$$

□

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

$$dx_t = (-f_t(x_t) + \sigma_t^2 \nabla \log q_t^{1,2}(x_t))dt + \sigma_t dW_t, \quad (178)$$

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

$$dx_t = (-f_t(x_t) + \sigma_t^2(\beta + (1 - \beta)a)(\nabla \log q_t^1(x_t) + \nabla \log q_t^2(x_t)))dt + \sqrt{\frac{\sigma_t^2(\beta + (1 - \beta)2a)}{\beta}} dW_t, \quad (179)$$

$$dw_t = \left[\beta\sigma_t^2\langle \nabla \log q_t^1(x_t), \nabla \log q_t^2(x_t) \rangle + \beta(\beta - 1)\frac{\sigma_t^2}{2}\|\nabla \log q_t^1(x_t) + \nabla \log q_t^2(x_t)\|^2 + (2\beta - 1)\langle \nabla, f_t(x_t) \rangle \right] dt. \quad (180)$$

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

$$\frac{\partial p_t(x)}{\partial t} = -\langle \nabla, p_t(x)(-2f_t(x) + \sigma_t^2(\nabla \log q_t^1(x) + \nabla \log q_t^2(x))) \rangle + \frac{\sigma_t^2}{2}\Delta p_t(x) + \quad (181)$$

$$+ p_t(x)(g_t(x) - \mathbb{E}_{p_t} g_t(x)), \quad (182)$$

where

$$g_t(x) = \langle \nabla \log q_t^1(x), -f_t(x) + \sigma_t^2 \nabla \log q_t^2(x) \rangle + \langle \nabla \log q_t^2(x), -f_t(x) + \sigma_t^2 \nabla \log q_t^1(x) \rangle - \quad (183)$$

$$- \sigma_t^2 \langle \nabla \log q_t^1(x), \nabla \log q_t^2(x) \rangle \quad (184)$$

$$= \sigma_t^2 \langle \nabla \log q_t^1(x), \nabla \log q_t^2(x) \rangle - \langle f_t(x), \nabla \log q_t^1(x) + \nabla \log q_t^2(x) \rangle. \quad (185)$$

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

$$\frac{\partial p_{t,\beta}(x)}{\partial t} = -\langle \nabla, p_{t,\beta}(x)(-2f_t(x) + \sigma_t^2(\beta + (1 - \beta)a)(\nabla \log q_t^1(x) + \nabla \log q_t^2(x))) \rangle + \quad (186)$$

$$+ \frac{\beta + (1 - \beta)2a}{\beta}\frac{\sigma_t^2}{2}\Delta p_{t,\beta}(x) + p_{t,\beta}(x)(g_t(x) - \mathbb{E}_{p_{t,\beta}} g_t(x)), \quad (187)$$

$$g_t(x) = \beta\sigma_t^2 \langle \nabla \log q_t^1(x), \nabla \log q_t^2(x) \rangle - \beta \langle f_t(x), \nabla \log q_t^1(x) + \nabla \log q_t^2(x) \rangle + \quad (188)$$

$$+ (\beta - 1)\langle \nabla, 2f_t(x) \rangle + \beta(\beta - 1)\frac{\sigma_t^2}{2}\|\nabla \log q_t^1(x) + \nabla \log q_t^2(x)\|^2. \quad (189)$$

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

$$\frac{\partial p_{t,\beta}(x)}{\partial t} = -\langle \nabla, p_{t,\beta}(x)(-f_t(x) + \sigma_t^2(\beta + (1 - \beta)a)(\nabla \log q_t^1(x) + \nabla \log q_t^2(x))) \rangle + \quad (190)$$

$$+ \frac{\beta + (1 - \beta)2a}{\beta}\frac{\sigma_t^2}{2}\Delta p_{t,\beta}(x) + p_{t,\beta}(x)(g_t(x) - \mathbb{E}_{p_{t,\beta}} g_t(x)), \quad (191)$$

$$g_t(x) = \beta\sigma_t^2 \langle \nabla \log q_t^1(x), \nabla \log q_t^2(x) \rangle + \beta(\beta - 1)\frac{\sigma_t^2}{2}\|\nabla \log q_t^1(x) + \nabla \log q_t^2(x)\|^2 + \quad (192)$$

$$+ (2\beta - 1)\langle \nabla, f_t(x) \rangle. \quad (193)$$

Thus, we get the following family of SDEs (for $0 \leq (\beta + (1 - \beta)2a)/\beta$)

$$dx_t = (-f_t(x_t) + \sigma_t^2(\beta + (1 - \beta)a)(\nabla \log q_t^1(x_t) + \nabla \log q_t^2(x_t)))dt + \sqrt{\frac{\sigma_t^2(\beta + (1 - \beta)2a)}{\beta}}dW_t, \quad (194)$$

$$dw_t = \left[\beta\sigma_t^2 \langle \nabla \log q_t^1(x_t), \nabla \log q_t^2(x_t) \rangle + \beta(\beta - 1)\frac{\sigma_t^2}{2} \|\nabla \log q_t^1(x_t) + \nabla \log q_t^2(x_t)\|^2 + (2\beta - 1)\langle \nabla, f_t(x_t) \rangle \right] dt. \quad (195)$$

□

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

$$dx_t = (-f_t(x_t) + \sigma_t^2 \nabla \log q_t^{1,2}(x_t))dt + \sigma_t dW_t, \quad (196)$$

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 $p_{t,\beta}(x) \propto q_t^1(x)^{1-\beta} q_t^2(x)^\beta$, for $(\beta + 2a(1 - \beta))/\beta \geq 0$, is

$$dx_t = \left(-f_t(x_t) + \sigma_t^2 \left(1 + \frac{a(1 - \beta)}{\beta} \right) ((1 - \beta) \nabla \log q_t^1(x_t) + \beta \nabla \log q_t^2(x_t)) \right) dt + \sigma_t \sqrt{1 + \frac{2a(1 - \beta)}{\beta}} dW_t, \quad (197)$$

$$dw_t = \frac{1}{2}\sigma_t^2 \beta(\beta - 1) \|\nabla \log q_t^1(x_t) - \nabla \log q_t^2(x_t)\|^2 + 2a\sigma_t^2(\beta - 1)^2 \langle \nabla \log q_t^1(x_t), \nabla \log q_t^2(x_t) \rangle. \quad (198)$$

Proof. First, according to Prop. D.1, we perform annealing $p_{t,1-\beta}^1(x) \propto q_t^1(x)^{1-\beta}$ and $p_{t,\beta}^2(x) \propto q_t^2(x)^\beta$, i.e.

$$\frac{\partial p_{t,1-\beta}^1(x)}{\partial t} = -\langle \nabla, p_{t,1-\beta}^1(x)(-f_t(x) + \sigma_t^2(1 - \beta + \beta a_1) \nabla \log q_t^1(x)) \rangle + \frac{1 - \beta + 2\beta a_1}{1 - \beta} \frac{\sigma_t^2}{2} \Delta p_{t,1-\beta}^1(x) + \quad (199)$$

$$+ p_{t,1-\beta}^1(x) \left(g_t(x) - \mathbb{E}_{p_{t,1-\beta}^1} g_t(x) \right), \quad (200)$$

$$g_t(x) = -\beta \langle \nabla, f_t(x) \rangle + \frac{1}{2}\sigma_t^2 \beta(\beta - 1) \|\nabla \log q_t^1(x_t)\|^2, \quad (201)$$

and

$$\frac{\partial p_{t,\beta}^2(x)}{\partial t} = -\langle \nabla, p_{t,\beta}^2(x)(-f_t(x) + \sigma_t^2(\beta + (1 - \beta)a_2) \nabla \log q_t^2(x)) \rangle + \frac{\beta + (1 - \beta)2a_2}{\beta} \frac{\sigma_t^2}{2} \Delta p_{t,\beta}^2(x) + \quad (202)$$

$$+ p_{t,\beta}^2(x) \left(g_t(x) - \mathbb{E}_{p_{t,\beta}^2} g_t(x) \right), \quad (203)$$

$$g_t(x) = (\beta - 1) \langle \nabla, f_t(x) \rangle + \frac{1}{2}\sigma_t^2 \beta(\beta - 1) \|\nabla \log q_t^2(x_t)\|^2, \quad (204)$$

Now, we would like to match the diffusion coefficient (to directly apply Prop. C.8 for diffusion equations in the product case, and avoid the evaluation of additional Laplacian terms in the weights).

$$\frac{1 - \beta + 2\beta a_1}{1 - \beta} = \frac{\beta + (1 - \beta)2a_2}{\beta} \implies \beta - \beta^2 + 2\beta^2 a_1 = \beta - \beta^2 + (1 - \beta)^2 2a_2 \quad (205)$$

$$\beta^2 a_1 = (1 - \beta)^2 a_2 \implies a_2 := a, \quad a_1 = \frac{a(1 - \beta)^2}{\beta^2}. \quad (206)$$

Now, according to [Table 1](#), for the product density $p_{t,\beta} \propto p_{t,1-\beta}^1(x)p_{t,\beta}^2(x)$, we have

$$\frac{\partial p_{t,\beta}(x)}{\partial t} = -\left\langle \nabla, p_{t,\beta}(x) \left(-2f_t(x) + \sigma_t^2(\beta + (1-\beta)a) \left(\frac{1-\beta}{\beta} \nabla \log q_t^1(x) + \nabla \log q_t^2(x) \right) \right) \right\rangle + \quad (207)$$

$$+ \frac{\beta + (1-\beta)2a}{\beta} \frac{\sigma_t^2}{2} \Delta p_{t,\beta}(x) + p_{t,\beta}(x)(g_t(x) - \mathbb{E}_{p_{t,\beta}} g_t(x)), \quad (208)$$

$$g_t(x) = \underbrace{-\beta \langle \nabla, f_t(x) \rangle + \frac{1}{2} \sigma_t^2 \beta (\beta - 1) \|\nabla \log q_t^1(x)\|^2}_{Eq. (201)} + \quad (209)$$

$$+ \underbrace{(\beta - 1) \langle \nabla, f_t(x) \rangle + \frac{1}{2} \sigma_t^2 \beta (\beta - 1) \|\nabla \log q_t^2(x)\|^2}_{Eq. (204)} + \quad (210)$$

$$+ (1-\beta) \langle \nabla \log q_t^1(x), -f_t(x) + \sigma_t^2(\beta + (1-\beta)a) \nabla \log q_t^2(x) \rangle + \quad (211)$$

$$+ \beta \left\langle \nabla \log q_t^2(x), -f_t(x) + \sigma_t^2 \frac{(1-\beta)}{\beta} (\beta + (1-\beta)a) \nabla \log q_t^1(x) \right\rangle - \quad (212)$$

$$- \sigma_t^2 \beta (1-\beta) \langle \nabla \log q_t^1(x), \nabla \log q_t^2(x) \rangle \quad (213)$$

where the terms in the last three lines arise from the conversion rules from the product in [Table 1](#). Finally, we obtain

$$g_t(x) = \frac{1}{2} \sigma_t^2 \beta (\beta - 1) \left(\|\nabla \log q_t^1(x) - \nabla \log q_t^2(x)\|^2 - 4 \frac{a(1-\beta)}{\beta} \langle \nabla \log q_t^1(x), \nabla \log q_t^2(x) \rangle \right) - \quad (214)$$

$$- \langle \nabla, f_t(x) \rangle - \langle (1-\beta) \nabla \log q_t^1(x) + \beta \nabla \log q_t^2(x), f_t(x) \rangle. \quad (215)$$

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

$$\frac{\partial p_{t,\beta}(x)}{\partial t} = -\left\langle \nabla, p_{t,\beta}(x) \left(-f_t(x) + \sigma_t^2 \left(1 + \frac{a(1-\beta)}{\beta} \right) ((1-\beta) \nabla \log q_t^1(x) + \beta \nabla \log q_t^2(x)) \right) \right\rangle + \quad (216)$$

$$+ \frac{\sigma_t^2}{2} \left(1 + \frac{2a(1-\beta)}{\beta} \right) \Delta p_{t,\beta}(x) + p_{t,\beta}(x)(g_t(x) - \mathbb{E}_{p_{t,\beta}} g_t(x)), \quad (217)$$

$$g_t(x) = \frac{1}{2} \sigma_t^2 \beta (\beta - 1) \|\nabla \log q_t^1(x) - \nabla \log q_t^2(x)\|^2 + 2a\sigma_t^2(\beta - 1)^2 \langle \nabla \log q_t^1(x), \nabla \log q_t^2(x) \rangle. \quad (218)$$

Thus, for $(\beta + 2a(1-\beta))/\beta \geq 0$, we have

$$dx_t = \left(-f_t(x_t) + \sigma_t^2 \left(1 + \frac{a(1-\beta)}{\beta} \right) ((1-\beta) \nabla \log q_t^1(x_t) + \beta \nabla \log q_t^2(x_t)) \right) dt + \sigma_t \sqrt{1 + \frac{2a(1-\beta)}{\beta}} dW_t, \quad (219)$$

$$dw_t = \frac{1}{2} \sigma_t^2 \beta (\beta - 1) \|\nabla \log q_t^1(x_t) - \nabla \log q_t^2(x_t)\|^2 + 2a\sigma_t^2(\beta - 1)^2 \langle \nabla \log q_t^1(x_t), \nabla \log q_t^2(x_t) \rangle. \quad (220)$$

□

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

$$dx_t = (-f_t(x_t) + \sigma_t^2 \nabla \log q_t(x_t)) dt + \sigma_t dW_t, \quad (221)$$

$$dx_t = (-f_t(x_t) + \sigma_t^2 \nabla \log q_t^{1,2}(x_t)) dt + \sigma_t dW_t, \quad (222)$$

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

$$dx_t = (-f_t(x_t) + \sigma_t^2(v_t^1(x_t) + v_t^2(x_t))) dt + \sigma_t dW_t, \quad (223)$$

$$dw_t = \frac{1}{2} \sigma_t^2 \beta(\beta - 1) \left(\|\nabla \log q_t(x_t) - \nabla \log q_t^1(x_t)\|^2 + \|\nabla \log q_t(x_t) - \nabla \log q_t^2(x_t)\|^2 \right) + \quad (224)$$

$$+ \sigma_t^2 \langle v_t^1(x_t), v_t^2(x_t) \rangle + \langle \nabla, f_t(x_t) \rangle, \quad (225)$$

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. D.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.

$$dx_t = \left(-f_t(x_t) + \sigma_t^2 \underbrace{((1 - \beta) \nabla \log q_t(x_t) + \beta \nabla \log q_t^1(x_t))}_{v_t^1(x_t)} \right) dt + \sigma_t dW_t, \quad (226)$$

$$dw_t = \frac{1}{2} \sigma_t^2 \beta(\beta - 1) \|\nabla \log q_t(x_t) - \nabla \log q_t^1(x_t)\|^2, \quad (227)$$

$$dx_t = \left(-f_t(x_t) + \sigma_t^2 \underbrace{((1 - \beta) \nabla \log q_t(x_t) + \beta \nabla \log q_t^2(x_t))}_{v_t^2(x_t)} \right) dt + \sigma_t dW_t, \quad (228)$$

$$dw_t = \frac{1}{2} \sigma_t^2 \beta(\beta - 1) \|\nabla \log q_t(x_t) - \nabla \log q_t^2(x_t)\|^2. \quad (229)$$

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

$$\frac{\partial p_{t,\beta}(x)}{\partial t} = -\langle \nabla, p_{t,\beta}(x) (-2f_t(x) + \sigma_t^2(v_t^1(x) + v_t^2(x))) \rangle + \frac{\sigma_t^2}{2} \Delta p_{t,\beta}(x) + p_{t,\beta}(x)(g_t(x) - \mathbb{E}_{p_{t,\beta}} g_t(x)), \quad (230)$$

$$g_t(x) = \frac{1}{2} \sigma_t^2 \beta(\beta - 1) \left(\|\nabla \log q_t(x) - \nabla \log q_t^1(x)\|^2 + \|\nabla \log q_t(x) - \nabla \log q_t^2(x)\|^2 \right) + \quad (231)$$

$$+ \langle v_t^1(x), -f_t(x) + \sigma_t^2 v_t^2(x) \rangle + \langle v_t^2(x), -f_t(x) + \sigma_t^2 v_t^1(x) \rangle - \sigma_t^2 \langle v_t^1(x), v_t^2(x) \rangle \quad (232)$$

$$= \frac{1}{2} \sigma_t^2 \beta(\beta - 1) \left(\|\nabla \log q_t(x) - \nabla \log q_t^1(x)\|^2 + \|\nabla \log q_t(x) - \nabla \log q_t^2(x)\|^2 \right) + \quad (233)$$

$$+ \sigma_t^2 \langle v_t^1(x), v_t^2(x) \rangle - \langle f_t(x), v_t^1(x) + v_t^2(x) \rangle. \quad (234)$$

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

$$\frac{\partial p_{t,\beta}(x)}{\partial t} = -\langle \nabla, p_{t,\beta}(x) (-f_t(x) + \sigma_t^2(v_t^1(x) + v_t^2(x))) \rangle + \frac{\sigma_t^2}{2} \Delta p_{t,\beta}(x) + p_{t,\beta}(x)(g_t(x) - \mathbb{E}_{p_{t,\beta}} g_t(x)), \quad (235)$$

$$g_t(x) = \frac{1}{2} \sigma_t^2 \beta(\beta - 1) \left(\|\nabla \log q_t(x) - \nabla \log q_t^1(x)\|^2 + \|\nabla \log q_t(x) - \nabla \log q_t^2(x)\|^2 \right) + \quad (236)$$

$$+ \sigma_t^2 \langle v_t^1(x), v_t^2(x) \rangle + \langle \nabla, f_t(x) \rangle, \quad (237)$$

which corresponds to

$$dx_t = (-f_t(x_t) + \sigma_t^2(v_t^1(x_t) + v_t^2(x_t))) dt + \sigma_t dW_t, \quad (238)$$

$$dw_t = \frac{1}{2} \sigma_t^2 \beta(\beta - 1) \left(\|\nabla \log q_t(x_t) - \nabla \log q_t^1(x_t)\|^2 + \|\nabla \log q_t(x_t) - \nabla \log q_t^2(x_t)\|^2 \right) + \quad (239)$$

$$+ \sigma_t^2 \langle v_t^1(x_t), v_t^2(x_t) \rangle + \langle \nabla, f_t(x_t) \rangle. \quad (240)$$

□

Proposition D.5 (Target Score Product SDE). *Consider two PDEs corresponding to the following SDEs*

$$dx_t = (-f_t(x_t) + \sigma_t^2 \nabla \log q_t^i(x_t))dt + \sigma_t dW_t, \quad (241)$$

with corresponding marginals $q_t^i(x_t)$. The SDE corresponding to the product of the marginals $p_{t,\beta}(x) \propto \prod_i q_t^i(x)^{\beta_i}$, can be simulated as follows

$$dx_t = \left(-f_t(x_t) + \sigma_t^2 \sum_i \beta_i \nabla \log q_t^i(x) \right) dt + \sigma_t dW_t, \quad (242)$$

$$dw_t = \left[\left(\sum_i \beta_i - 1 \right) \langle \nabla, f_t(x_t) \rangle + \frac{\sigma_t^2}{2} \left\| \sum_i \beta_i \nabla \log q_t^i(x_t) \right\|^2 - \frac{\sigma_t^2}{2} \sum_i \beta_i \|\nabla \log q_t^i(x_t)\|^2 \right] dt. \quad (243)$$

Proof. The target log-density is defined as

$$\log p_t(x) = \sum_i \beta_i \log q_t^i(x) - \log Z_t. \quad (244)$$

The time-derivative of the target log-density is

$$\frac{\partial}{\partial t} \log p_t(x) = \sum_i \beta_i \frac{\partial}{\partial t} \log q_t^i(x) - \mathbb{E}_{p_t(x)} \sum_i \beta_i \frac{\partial}{\partial t} \log q_t^i(x). \quad (245)$$

We focus on the first term, and write for all the marginals their corresponding PDEs

$$\frac{\partial}{\partial t} \log q_t^i(x) = -\langle \nabla, -f_t(x) + \sigma_t^2 \nabla \log q_t^i(x) \rangle - \langle \nabla \log q_t^i(x), -f_t(x) + \sigma_t^2 \nabla \log q_t^i(x) \rangle + \quad (246)$$

$$+ \frac{\sigma_t^2}{2} \Delta \log q_t^i(x) + \frac{\sigma_t^2}{2} \|\nabla \log q_t^i(x)\|^2 \quad (247)$$

$$= -\langle \nabla, -f_t(x) + \sigma_t^2 \nabla \log q_t^i(x) \rangle - \langle \nabla \log q_t^i(x), -f_t(x) \rangle + \frac{\sigma_t^2}{2} \Delta \log q_t^i(x) - \frac{\sigma_t^2}{2} \|\nabla \log q_t^i(x)\|^2.$$

$$\sum_i \beta_i \frac{\partial}{\partial t} \log q_t^i(x) = -\left\langle \nabla, -\sum_i \beta_i f_t(x) + \sigma_t^2 \nabla \log p_t(x) \right\rangle - \langle \nabla \log p_t(x), -f_t(x) \rangle + \quad (248)$$

$$+ \frac{\sigma_t^2}{2} \Delta \log p_t(x) - \frac{\sigma_t^2}{2} \sum_i \beta_i \|\nabla \log q_t^i(x)\|^2 \quad (249)$$

$$= -\langle \nabla, -f_t(x) + \sigma_t^2 \nabla \log p_t(x) \rangle - \langle \nabla \log p_t(x), -f_t(x) + \sigma_t^2 \nabla \log p_t(x) \rangle + \quad (250)$$

$$+ \frac{\sigma_t^2}{2} \Delta \log p_t(x) + \frac{\sigma_t^2}{2} \|\nabla \log p_t(x)\|^2 + \quad (251)$$

$$+ \left(\sum_i \beta_i - 1 \right) \langle \nabla, f_t(x) \rangle + \frac{\sigma_t^2}{2} \|\nabla \log p_t(x)\|^2 - \frac{\sigma_t^2}{2} \sum_i \beta_i \|\nabla \log q_t^i(x)\|^2. \quad (252)$$

Writing down the PDE $p_t(x)$, we get

$$\frac{\partial p_t(x)}{\partial t} = -\langle \nabla, p_t(x) (-f_t(x) + \sigma_t^2 \nabla \log p_t(x)) \rangle + \frac{\sigma_t^2}{2} \Delta p_t(x) + p_t(x) (g_t(x) - \mathbb{E}_{p_t(x)} g_t(x)), \quad (253)$$

$$g_t(x) = \left(\sum_i \beta_i - 1 \right) \langle \nabla, f_t(x) \rangle + \frac{\sigma_t^2}{2} \|\nabla \log p_t(x)\|^2 - \frac{\sigma_t^2}{2} \sum_i \beta_i \|\nabla \log q_t^i(x)\|^2, \quad (254)$$

which ends the proof.

In particular, for the target $p_t(x) \propto q_t(x)^{(1-\beta)}(q_t^1(x)q_t^2(x))^{\beta/2}$, we have

$$dx_t = \left(-f_t(x_t) + \sigma_t^2 \left((1-\beta)\nabla \log q_t(x) + \frac{\beta}{2}\nabla \log q_t^1(x) + \frac{\beta}{2}\nabla \log q_t^2(x) \right) \right) dt + \sigma_t dW_t, \quad (255)$$

$$dw_t = \frac{\sigma_t^2}{2} \left[\left\| (1-\beta)\nabla \log q_t(x) + \frac{\beta}{2}\nabla \log q_t^1(x) + \frac{\beta}{2}\nabla \log q_t^2(x) \right\|^2 - \right. \quad (256)$$

$$\left. - \left((1-\beta)\|\nabla \log q_t(x_t)\|^2 + \frac{\beta}{2}\|\nabla \log q_t^1(x_t)\|^2 + \frac{\beta}{2}\|\nabla \log q_t^2(x_t)\|^2 \right) \right] dt. \quad (257)$$

□

Proposition D.6 (Reward-tilted SDE). *Consider the following SDE*

$$dx_t = v_t(x)dt + \sigma_t dW_t, \quad (258)$$

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

$$dx_t = (v_t(x_t) + a\nabla r(x_t))dt + \sigma_t dW_t, \quad (259)$$

$$dw_t = \left[\left\langle \nabla r(x_t), \beta_t(v_t(x_t) - \sigma_t^2 \nabla \log q_t(x_t) - \frac{\sigma_t^2}{2} \beta_t \nabla r(x_t)) + a(\nabla \log q_t(x_t) + \beta_t \nabla r(x_t)) \right\rangle + \right. \quad (260)$$

$$\left. + \left(a - \beta_t \frac{\sigma_t^2}{2} \right) \Delta r(x_t) + \frac{\partial \beta_t}{\partial t} r(x_t) \right] dt. \quad (261)$$

For the reverse-time SDE with drift $v_t(x_t) = -f_t(x_t) + \sigma_t^2 \nabla \log q_t(x_t)$ corresponding to the original diffusion generative model and $a = \beta_t \sigma_t^2 / 2$, we obtain the following weighted SDE

$$dx_t = (-f_t(x_t) + \sigma_t^2 \nabla \log q_t(x_t) + \beta_t \frac{\sigma_t^2}{2} \nabla r(x_t))dt + \sigma_t dW_t, \quad (262)$$

$$dw_t = \left[\frac{\partial \beta_t}{\partial t} r(x) + \left\langle \beta_t \nabla r(x), \frac{\sigma_t^2}{2} \nabla \log q_t(x) - f_t(x) \right\rangle \right] dt \quad (263)$$

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

$$\frac{\partial q_t(x)}{\partial t} = -\langle \nabla, q_t(x)v_t(x) \rangle + \frac{\sigma_t^2}{2} \Delta q_t(x). \quad (264)$$

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

$$p_t(x) = \frac{q_t(x) \exp(\beta_t r(x))}{\int dx q_t(x) \exp(\beta_t r(x))}. \quad (265)$$

Straightforwardly, we get

$$\frac{\partial}{\partial t} \log p_t(x) = \frac{\partial}{\partial t} \log q_t(x) + \frac{\partial \beta_t}{\partial t} r(x) - \int dx p_t(x) \left[\frac{\partial}{\partial t} \log q_t(x) + \frac{\partial \beta_t}{\partial t} r(x) \right] \quad (266)$$

For the first term, we have

$$\frac{\partial}{\partial t} \log q_t(x) = -\langle \nabla, v_t(x) \rangle - \langle \nabla \log q_t(x), v_t(x) \rangle + \frac{\sigma_t^2}{2} \Delta \log q_t(x) + \frac{\sigma_t^2}{2} \|\nabla \log q_t(x)\|^2 \quad (267)$$

$$= -\langle \nabla, v_t(x) \rangle - \langle \nabla \log p_t(x), v_t(x) \rangle + \frac{\sigma_t^2}{2} \Delta \log p_t(x) + \frac{\sigma_t^2}{2} \|\nabla \log p_t(x)\|^2 + \quad (268)$$

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

Thus, we have

$$\frac{\partial p_t(x)}{\partial t} = -\langle \nabla, p_t(x)v_t(x) \rangle + \frac{\sigma_t^2}{2} \Delta p_t(x) + p_t(x)(g_t(x) - \mathbb{E}_{p_t(x)} g_t(x)) \quad (269)$$

$$g_t(x) = \left\langle \beta_t \nabla r(x), v_t(x) - \sigma_t^2 \nabla \log q_t(x) - \frac{\sigma_t^2}{2} \beta_t \nabla r(x) \right\rangle - \beta_t \frac{\sigma_t^2}{2} \Delta r(x) + \frac{\partial \beta_t}{\partial t} r(x). \quad (270)$$

Furthermore, we can add the gradient of the reward as additional drift term $a \nabla r(x)$, i.e.

$$\frac{\partial p_t(x)}{\partial t} = -\langle \nabla, p_t(x)(v_t(x) + a \nabla r(x)) \rangle + \frac{\sigma_t^2}{2} \Delta p_t(x) + p_t(x)(g_t(x) - \mathbb{E}_{p_t(x)} g_t(x)) \quad (271)$$

$$\begin{aligned} g_t(x) &= a \Delta r(x) + a \langle \nabla \log p_t(x), \nabla r(x) \rangle - \beta_t \frac{\sigma_t^2}{2} \Delta r(x) + \frac{\partial \beta_t}{\partial t} r(x) + \\ &\quad + \left\langle \beta_t \nabla r(x), v_t(x) - \sigma_t^2 \nabla \log q_t(x) - \frac{\sigma_t^2}{2} \beta_t \nabla r(x) \right\rangle. \end{aligned} \quad (272)$$

Taking $v_t(x) = -f_t(x) + \sigma_t^2 \nabla \log q_t(x)$ and $a = \beta_t \sigma_t^2 / 2$, we have

$$\frac{\partial p_t(x)}{\partial t} = -\left\langle \nabla, p_t(x)(-f_t(x) + \sigma_t^2 \nabla \log q_t(x) + \beta_t \frac{\sigma_t^2}{2} \nabla r(x)) \right\rangle + \frac{\sigma_t^2}{2} \Delta p_t(x) + p_t(x)(g_t(x) - \mathbb{E}_{p_t(x)} g_t(x)) \quad (273)$$

$$\begin{aligned} g_t(x) &= \left\langle \beta_t \nabla r(x), \frac{\sigma_t^2}{2} \nabla \log p_t(x) \right\rangle + \frac{\partial \beta_t}{\partial t} r(x) + \left\langle \beta_t \nabla r(x), -f_t(x) - \frac{\sigma_t^2}{2} \beta_t \nabla r(x) \right\rangle \\ &= \frac{\partial \beta_t}{\partial t} r(x) + \left\langle \beta_t \nabla r(x), \frac{\sigma_t^2}{2} \nabla \log q_t(x) - f_t(x) \right\rangle. \end{aligned} \quad (274)$$

This can be simulated as

$$dx_t = (-f_t(x_t) + \sigma_t^2 \nabla \log q_t(x_t) + \beta_t \frac{\sigma_t^2}{2} \nabla r(x_t))dt + \sigma_t dW_t, \quad (275)$$

$$dw_t = \left[\frac{\partial \beta_t}{\partial t} r(x) + \left\langle \beta_t \nabla r(x), \frac{\sigma_t^2}{2} \nabla \log q_t(x) - f_t(x) \right\rangle \right] dt \quad (276)$$

□

E. Additional Related Work

Amortized Sampling Recently, there has been renewed interest 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; 2024); Richter & Berner (2024); Akhound-Sadegh et al. (2024); Albergo & Vanden-Eijnden (2024); De 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 approach can be applied 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., 2016; 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. (2025) 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.

F. Additional Experimental Details and Results

F.1. Sampling Metrics

We use a number of metrics to assess the quality of generated samples. These metrics capture different aspects of the distribution. Before computing metrics we filter out samples with energy > 100 . This only affects non-resampled metrics

and prevents numerical. We find this filters out no samples for DEM or with FKC, and filters less than 3% of samples with target score SDE or tempered noise SDE sampling. We justify this as it is easy to set these filters for generated samples of very poor quality.

Distance- \mathcal{W}_2 For the LJ-13 task we compute the 2-Wasserstein distance between pairwise distance histograms. For this metric we take all pairwise distances for all samples and flatten them into a single distribution. For a sample of 10,000 points this leads to distributions of size 700,000 as there are 70 pairwise distances for a 13 particle system. This is useful metric as it is equivariant and measures the global fidelity of the generated samples. It however is not useful for assessing fine grained details of the generated samples. For that we turn to the Energy- $\mathcal{W}_{1/2}$ distances.

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.

We note that for the LJ-13 task we exclude samples with energy > 100 for all methods and metrics. In practice this only affects Target Score and Tempered Noise SDEs without FKC and DEM trained at lower temperatures. This excludes roughly 2-3% of samples for those models, which helps these baselines.

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. 5.2, where we start at temperature $T_L = 3$ and anneal to target temperature $T_S = 1/3$. 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. (22) 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 Sec. 4 and App. B.2.

On ground truth q_t^β A subtle point to note is that $q_t^{T_L}$ is not a mixture of $|\pi|$ Gaussians, but rather $|\pi|^{T_L}$ Gaussians for integer T_L . This means that we are restricted to small integer T_L . We use $T_L = 3$ for all experiments in the 40 Gaussians setting. Note that we reserve $\beta = T_S/T_L$ for the ratio of learning and sampling/target temperatures.

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:

$$\mathcal{E}^{\text{LJ}}(x) = \frac{\varepsilon}{2\tau} \sum_{ij} \left(\left(\frac{r_m}{d_{ij}} \right)^6 - \left(\frac{r_m}{d_{ij}} \right)^{12} \right) \quad (277)$$

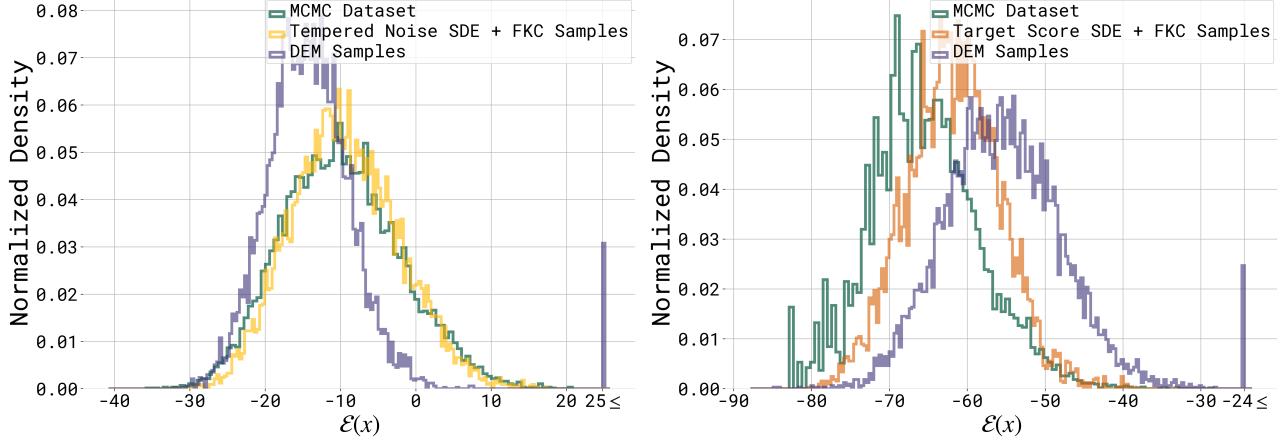


Figure A1. Comparison between the energy distribution of the MCMC dataset, samples generated using a DEM model trained at the target temperature, and samples generated using temperature annealing from a model trained at starting distribution $T = 2$. **Left:** the target temperature is 1.5 and **Right:** the target temperature is 0.8.

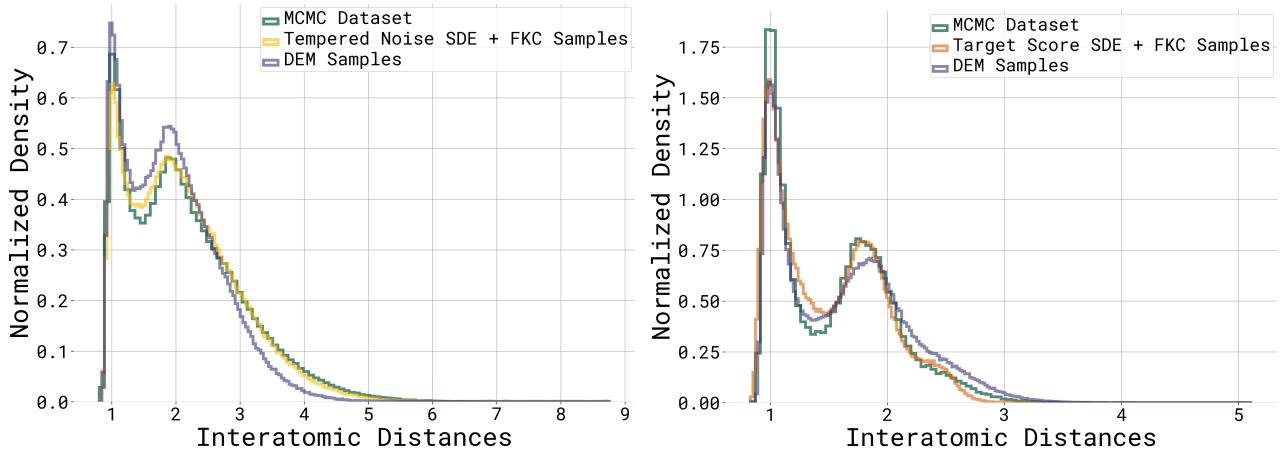


Figure A2. Comparison between the distribution of the interatomic distances of the particles in the MCMC dataset, samples generated using a DEM model trained at the target temperature, and samples generated using temperature annealing from a model trained at starting distribution $T = 2$. **Left:** the target temperature is 1.5 and **Right:** the target temperature is 0.8.

Table A1. 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	0.007 ± 0.005	0.372 ± 0.020	6.256 ± 3.960	11.265 ± 5.629
Tempered Noise	✓ systematic	0.926 ± 0.248	0.027 ± 0.011	0.512 ± 0.017	9.974 ± 1.229	14.045 ± 1.308

Table A2. Additional results for LJ-13 at different target temperatures. The model is trained at starting temperature $T_L = 2.0$ and metrics are computed over 3 runs. DEM is run for one seed only as the standard-deviation over seeds is negligible.

Target Temp.	SDE Type	FKC	distance- \mathcal{W}_2	Energy- \mathcal{W}_1	Energy- \mathcal{W}_2
0.9 ($\beta=2.2$)	Target Score	✗	0.215 ± 0.001	13.886 ± 0.040	14.893 ± 0.012
		✓	0.042 ± 0.009	6.218 ± 0.896	6.259 ± 0.873
	Tempered Noise	✗	0.110 ± 0.016	5.633 ± 0.090	7.682 ± 0.585
		✓	0.042 ± 0.004	4.384 ± 0.135	4.530 ± 0.167
	DEM	—	0.168 ± —	14.516 ± —	14.606 ± —
1.0 ($\beta=2.0$)	Target Score	✗	0.221 ± 0.001	12.915 ± 0.054	13.558 ± 0.112
		✓	0.039 ± 0.008	2.629 ± 0.665	2.876 ± 0.548
	Tempered Noise	✗	0.094 ± 0.002	5.215 ± 0.095	6.560 ± 0.000
		✓	0.053 ± 0.008	3.205 ± 0.462	3.538 ± 0.468
	DEM	—	0.127 ± —	1.352 ± —	2.050 ± —
1.2 ($\beta=1.67$)	Target Score	✗	0.234 ± 0.004	10.414 ± 0.036	10.910 ± 0.110
		✓	0.026 ± 0.001	2.831 ± 0.155	2.915 ± 0.074
	Tempered Noise	✗	0.098 ± 0.002	4.258 ± 0.069	5.564 ± 0.095
		✓	0.076 ± 0.006	1.017 ± 0.494	1.300 ± 0.433
	DEM	—	0.143 ± —	9.669 ± —	9.736 ± —

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 $\mathcal{E}^{LJ-system} = \mathcal{E}^{LJ}(x) + c\mathcal{E}^{osc}(x)$. The harmonic potential is given by:

$$\mathcal{E}^{osc}(x) = \frac{1}{2} \sum_i \|x_i - x_{COM}\|^2 \quad (278)$$

where x_{COM} refers to the center of mass of the system. We set $r_m = 1, \tau = 1, \epsilon = 2.0$ 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 $t_{start} = 0.99$ 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 Figs. A1 and A2, 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.

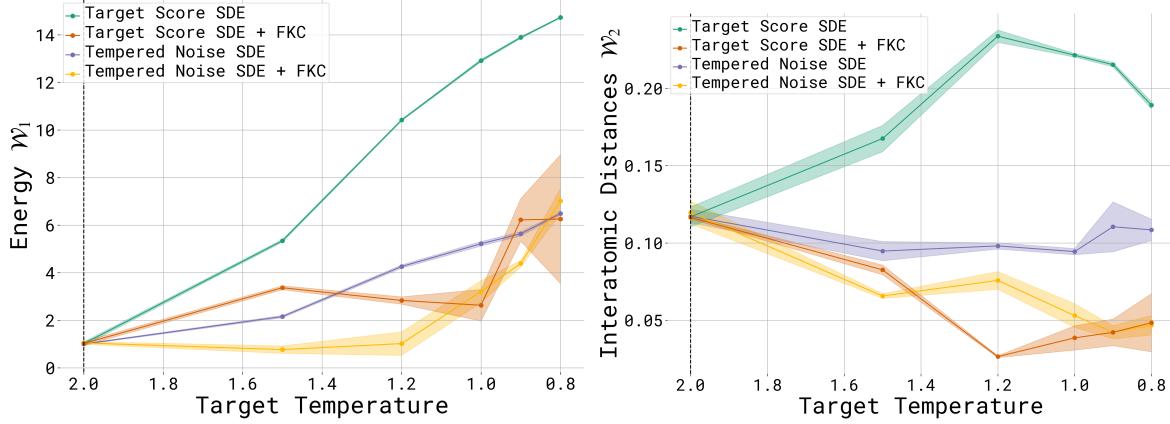


Figure A3. **Left:** 1-Wasserstein between energy distributions and **Right:** 2-Wasserstein between distributions of interatomic distances of MCMC samples from the annealed distribution and generated samples.

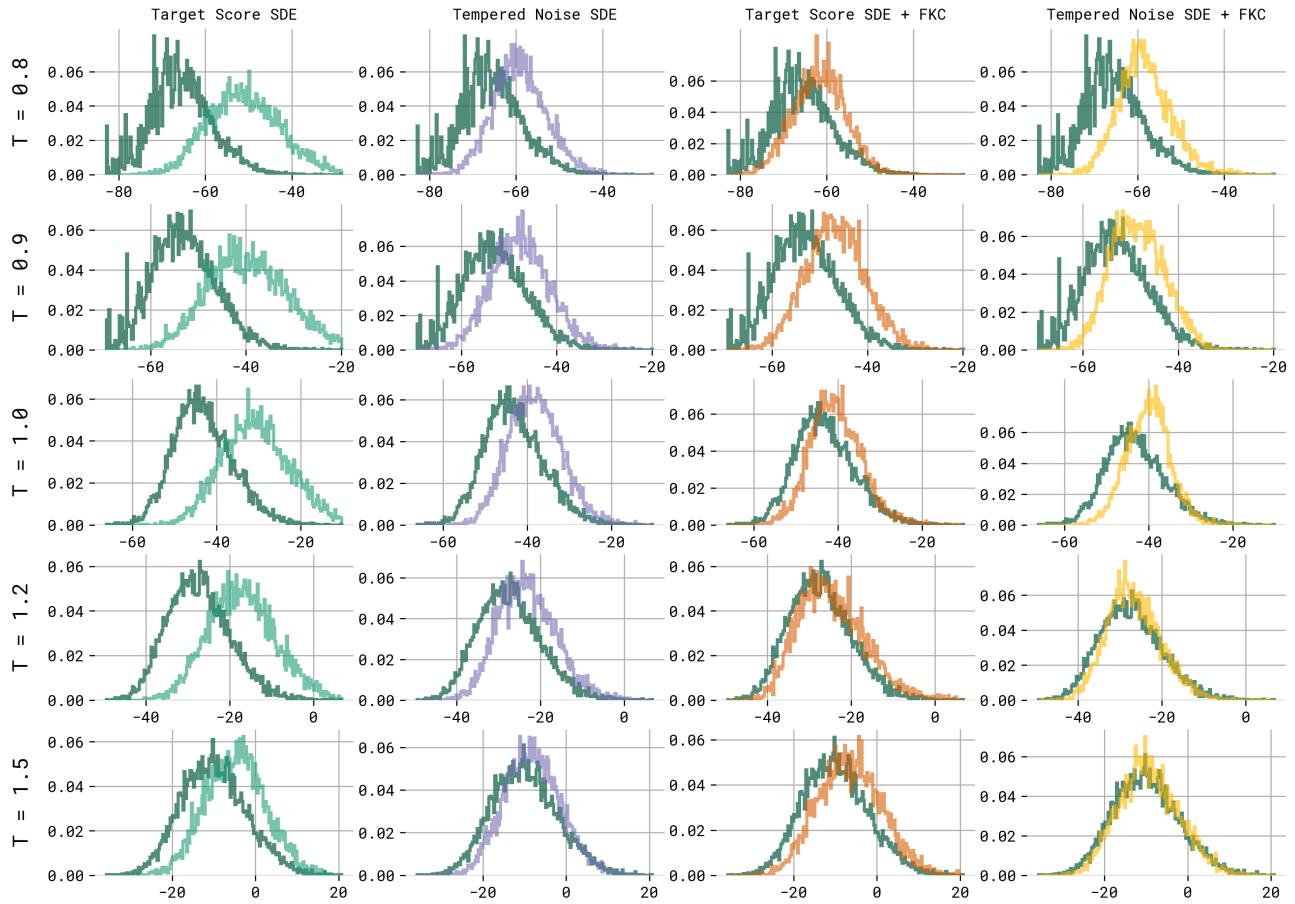


Figure A4. Energy distributions of samples generated with temperature annealing compared to the MCMC samples (in dark green), at different target temperatures. The starting temperature is $T_L = 2.0$.

Table A3. Comparison of EDM2+FKC (✓) with EDM2+CFG (✗) for image generation using EDM2. With sweep over batchsize K . For all metrics, we report CLIP and ImageReward (IR) scores averaged over 10,000 images.

FKC	γ	BS	N	CLIP (\uparrow)	IR (\uparrow)
✓	40	1	32	28.75	-0.24
✓	40	2	32	29.03	-0.07
✓	40	4	32	29.11	0.02
✓	40	8	32	29.14	0.05
✓	40	16	32	29.04	0.05
✓	40	32	32	29.00	0.04
✓	40	64	32	28.95	0.01

Sampling Reference distributions To generate reference distributions from the Lennard-Jones-13 potential we use Pyro (Bingham et al., 2019) and a No-U-Turn sampler (Hoffman & Gelman, 2014) with default arguments. We use 20k warmup steps and collect 20k samples from the 10 chains for each temperature.

Additional results In Table A2 we can see additional results extending Table 3 for intermediate temperatures. Here we can see that generally the same patterns hold with one exception—DEM on a target temperature of 1.0 is better than FKC with $\beta = 2.0$ on Energy- \mathcal{W}_1 and Energy- \mathcal{W}_2 metrics. This means that DEM has on this temperature has better local fidelity but slightly worse global fidelity. This is quite interesting as we know DEM was originally developed and therefore tuned with a temperature of 1.0 on this dataset. Our hypothesis is that some hyperparameters are specifically tailored to this setting. It is quite interesting then that FKC can still perform better than DEM on global metrics for this temperature.

In Fig. A3 we can see the Energy- \mathcal{W}_1 and Interatomic Distance \mathcal{W}_2 metrics plotted against the target temperature using a model trained at temperature 2.0. Here we see that FKC performs well across all temperatures for our global metric of interatomic distances and across energy \mathcal{W}_1 distances, although at low target temperatures the Energy- \mathcal{W}_1 metric gets worse for all methods. We note that this is after excluding roughly 2-3% of samples with unacceptably bad energy from the Target Score SDE and Tempered Noise SDE. Therefore even though the lines are close here, we still prefer the FKC samplers.

F.4. EDM2 image generation

Additional experimental details For all experiments with EDM2 we use the default classifier free guidance weight $\beta = 1.4$. For the γ ablation we use batch size 32 for FKC and for the steps experiment we use batch size 8 for FKC.

Additional Results In Table A3 we experiment with FKC batch size. Theoretically larger batch sizes should be better as it corresponds to larger sets for the importance sampling step. We see that FKC gets higher scores with larger batch sizes, even with batch size 2 with performance plateauing after batch size ≈ 8 .

F.5. Latent vs. ambient space image models

We apply the FKC method for generating images with Stable Diffusion-XL (SDXL), a latent diffusion model. We show performance of SDXL+FKC on the GenEval benchmark in Table A4, but we do not observe any significant increase from SDXL with vanilla CFG. While there are examples where FKC improves the semantic accuracy of SD-XL generations, (see Fig. A5), the gain is not consistent when evaluated on 1000 prompts. There are multiple potential reasons for this behaviour, e.g. sampling from the geometric average is not better than CFG for these metrics, or the dimensionality of the latent space is too high, resulting in weights with too much variance. Investigating the effectiveness of these methods with latent diffusion models could be an interesting area of future study; in the present study, we find FKC to be consistently helpful when applied to models in the ambient space.

F.6. Multi-Target Structure-Based Drug Design

Additional experimental details for main experiments Following Zhou et al. (2024), we align the target pockets in 3D space and generate sample coordinates for each pocket using an SE(3)-equivariant graph neural network over 1000 integration steps. We then use our Product of Experts (PoE) scheme (Prop. 3.3) to guide ligand generation.

We note that the PoE weight computation in Eq. (27) necessitates equal sample dimensionality, otherwise resampling would be skewed to favor samples of higher dimensions. This requires the molecules within a batch to have the same number of atoms. This motivates our use of fixed size molecule bins for generation.

Table A4. Image generation using SDXL with classifier-free guidance (CFG) on the GenEval benchmark (Ghosh et al., 2023). For all metrics mean values are reported. * indicates values directly taken from leaderboard.

Model	β	Overall	Single object	Two object	Counting	Colors	Position	Color attribution
GenEval original 553 prompts								
CLIP retrieval*	—	0.35	0.89	0.22	0.37	0.62	0.03	0.00
SD-1.5*	—	0.43	0.97	0.38	0.35	0.76	0.04	0.06
SDXL*	—	0.55	0.98	0.74	0.39	0.85	0.15	0.23
SDXL (our run)	7.5	0.57	0.99	0.80	0.46	0.86	0.11	0.22
SDXL+FKC	5.5	0.58	0.99	0.77	0.49	0.87	0.10	0.22
SDXL+FKC	7.5	0.57	0.99	0.78	0.46	0.83	0.13	0.23
GenEval 1000 prompts								
SDXL (our run)	7.5	0.58	0.99	0.79	0.45	0.88	0.11	0.21
SDXL+FKC	5.5	0.57	0.99	0.79	0.42	0.86	0.13	0.21
SDXL+FKC	7.5	0.57	0.99	0.80	0.45	0.83	0.13	0.22



Figure A5. Examples where SDXL+FKC (bottom) outperforms EDM2+CFG (top).

Our baseline is the target score SDE with $\beta = 0.5$, which is equivalent to DualDiff from Zhou et al. (2024) and also corresponds to an averaging of scores (Liu et al., 2022). We also generate molecules conditioned on a single protein pocket using TargetDiff from Guan et al. (2023), but dock the molecules to both targets in a protein pair to understand the need for conditioning on two pockets simultaneously.

SDE Component Analysis In Table A5, we show the performance of varying the following SDE settings for dual-target drug design: SDE type, β , and the presence/absence of FKC resampling. Here, we report metrics for generating molecules on a single protein pocket pair (UniProt IDs P23786/P05023) as the validation set. We generate molecules a batch 32 molecules for 5 different molecule lengths, which were sampled from the original training distribution (Guan et al., 2023): {15, 19, 23, 27, 35}.

We study the impact of the following changing the following settings:

Inverse temperature (β) We find that as we increase β from 0.5 to 2.0 the product of the docking scores of the protein pair increases, though the delta increase is larger at smaller β s.

FKC. Next, we try turning FKC on at a fixed β . We find that performance improves at both $\beta = 1.5$ and $\beta = 2.0$, although the improvement at $\beta = 2.0$ is larger. However, this comes at a cost of diversity and the uniqueness of molecules generated.

t_{\max} Given that resampling is helpful in terms of improving the quality of the final molecules but decreases molecular diversity, we investigate setting a t_{\max} for our best β settings, where we resample only when $\tau \leq t_{\max}$. We find that

Table A5. Performance of generated molecules with different SDE settings. We generate 32 molecules for 5 molecule sizes for one protein pair for each setting. Lower docking scores are better. Values are reported as averages over all generated molecules in each run. "Better than ref." is the percentage of ligands with better docking scores than known reference molecules for *both* targets (the mean docking score for the reference molecules is -8.255 ± 1.849). We also report the diversity, validity & uniqueness, and quality, which refers to the percentage of molecules that are valid, unique, have QED ≥ 0.6 and SA ≤ 4.0 (Lee et al., 2025b). Bolded values are the best metrics within each set of midlines. For $\beta = 1$, target score and tempering noise match (Prop. 3.3).

β	FKC	t_{\max}	SDE Type	$(P_1 * P_2)(\uparrow)$	$\max(P_1, P_2)(\downarrow)$	$P_1(\downarrow)$	$P_2(\downarrow)$	Better than ref. (\uparrow)	Div. (\uparrow)	Val. & Uniq. (\uparrow)	Qual. (\uparrow)
0.5	✗	—	Target Score	67.657 \pm 11.985	-7.667 \pm 0.687	-8.377 \pm 0.661	-7.986 \pm 0.948	0.251 \pm 0.199	0.886\pm0.006	0.969 \pm 0.062	0.244 \pm 0.161
1.0	✗	—	Target Score	73.360 \pm 14.423	-7.929 \pm 0.763	-8.843 \pm 0.899	-8.174 \pm 0.989	0.378 \pm 0.311	0.884 \pm 0.008	0.962 \pm 0.023	0.231 \pm 0.170
1.5	✗	—	Target Score	75.213 \pm 15.779	-8.085 \pm 0.856	-8.980 \pm 0.935	-8.258 \pm 1.024	0.402\pm0.339	0.880 \pm 0.012	0.988 \pm 0.015	0.250 \pm 0.159
2.0	✗	—	Target Score	75.551\pm16.345	-8.089\pm0.899	-8.966\pm0.884	-8.309\pm1.112	0.391 \pm 0.331	0.881 \pm 0.011	0.994\pm0.012	0.288\pm0.179
1.5	✗	—	Target Score	75.213 \pm 15.779	-8.085\pm0.856	-8.980\pm0.935	-8.258 \pm 1.024	0.402 \pm 0.339	0.880\pm0.012	0.988\pm0.015	0.250\pm0.159
1.5	✓	—	Target Score	75.798\pm32.984	-7.438 \pm 2.507	-8.582 \pm 3.200	-8.829\pm1.193	0.446\pm0.454	0.651 \pm 0.102	0.475 \pm 0.169	0.100 \pm 0.157
2.0	✗	—	Target Score	75.551 \pm 16.345	-8.089 \pm 0.899	-8.966 \pm 0.884	-8.309 \pm 1.112	0.391 \pm 0.331	0.881\pm0.011	0.994\pm0.012	0.288\pm0.179
2.0	✓	—	Target Score	91.845\pm28.421	-8.977\pm1.433	-9.984\pm1.533	-8.978\pm1.434	0.671\pm0.419	0.617 \pm 0.049	0.475 \pm 0.132	0.044 \pm 0.073
1.5	✓	0.4	Target Score	74.558 \pm 14.361	-7.961 \pm 0.785	-8.883 \pm 0.885	-8.322 \pm 1.083	0.372 \pm 0.328	0.883\pm0.021	0.981 \pm 0.038	0.262 \pm 0.222
1.5	✓	0.5	Target Score	80.421 \pm 16.567	-8.365 \pm 0.905	-9.314 \pm 0.882	-8.539 \pm 1.077	0.494 \pm 0.378	0.867 \pm 0.014	0.994\pm0.012	0.288\pm0.233
1.5	✓	0.6	Target Score	83.405\pm19.024	-8.530\pm1.070	-9.516\pm1.083	-8.646 \pm 1.098	0.485 \pm 0.441	0.820 \pm 0.024	0.925 \pm 0.078	0.244 \pm 0.196
1.5	✓	0.7	Target Score	83.100 \pm 19.354	-8.434 \pm 0.912	-9.420 \pm 1.229	-8.723 \pm 1.227	0.503\pm0.441	0.799 \pm 0.030	0.888 \pm 0.094	0.162 \pm 0.205
1.5	✓	1.0	Target Score	75.798 \pm 32.984	-7.438 \pm 2.507	-8.582 \pm 3.200	-8.829\pm1.193	0.446 \pm 0.454	0.651 \pm 0.102	0.475 \pm 0.169	0.100 \pm 0.157
2.0	✓	0.4	Target Score	79.734 \pm 17.631	-8.331 \pm 0.981	-9.283 \pm 0.968	-8.467 \pm 1.133	0.498 \pm 0.388	0.876\pm0.012	0.988\pm0.015	0.306\pm0.234
2.0	✓	0.5	Target Score	84.949 \pm 19.056	-8.569 \pm 0.978	-9.529 \pm 1.018	-8.796 \pm 1.220	0.514 \pm 0.423	0.851 \pm 0.025	0.938 \pm 0.059	0.244 \pm 0.160
2.0	✓	0.6	Target Score	87.983 \pm 22.856	-8.790 \pm 1.231	-9.619 \pm 1.101	-8.988\pm1.442	0.569 \pm 0.468	0.818 \pm 0.011	0.888 \pm 0.073	0.212 \pm 0.223
2.0	✓	0.7	Target Score	85.168 \pm 21.258	-8.574 \pm 1.097	-9.514 \pm 1.028	-8.827 \pm 1.452	0.593 \pm 0.484	0.786 \pm 0.034	0.806 \pm 0.096	0.231 \pm 0.224
2.0	✓	1.0	Target Score	91.845\pm28.421	-8.977\pm1.433	-9.984\pm1.533	-8.978 \pm 1.434	0.671\pm0.419	0.617 \pm 0.049	0.475 \pm 0.132	0.044 \pm 0.073
0.5	✗	—	Target Score	67.657 \pm 11.985	-7.667 \pm 0.687	-8.377 \pm 0.661	-7.986 \pm 0.948	0.251 \pm 0.199	0.886 \pm 0.006	0.969\pm0.062	0.244 \pm 0.161
0.5	✗	—	Tempered Noise	71.606\pm15.139	-7.838\pm0.872	-8.727\pm0.878	-8.085\pm1.088	0.362\pm0.274	0.887\pm0.007	0.944 \pm 0.098	0.300\pm0.212
0.5	✓	0.6	Target Score	77.100 \pm 16.533	-8.243 \pm 0.949	-9.112 \pm 0.898	-8.337 \pm 1.045	0.417 \pm 0.337	0.877 \pm 0.008	0.975\pm0.023	0.212 \pm 0.155
0.5	✓	0.6	Tempered Noise	78.501\pm15.388	-8.323\pm0.919	-9.127\pm0.770	-8.496\pm1.100	0.496\pm0.308	0.879\pm0.016	0.931 \pm 0.064	0.250\pm0.163
2.0	✗	—	Target Score	75.551 \pm 16.345	-8.089\pm0.899	-8.966 \pm 0.884	-8.309 \pm 1.112	0.391 \pm 0.331	0.881\pm0.011	0.994\pm0.012	0.288\pm0.179
2.0	✗	—	Tempered Noise	75.868\pm16.154	-8.045 \pm 0.909	-8.977\pm0.978	-8.352\pm1.095	0.460\pm0.390	0.874 \pm 0.008	0.994\pm0.012	0.262 \pm 0.186
2.0	✓	0.6	Target Score	87.983\pm22.856	-8.790\pm1.231	-9.619\pm1.101	-8.988\pm1.442	0.569\pm0.468	0.818\pm0.011	0.888\pm0.073	0.212 \pm 0.223
2.0	✓	0.6	Tempered Noise	79.696 \pm 18.087	-8.229 \pm 0.997	-9.104 \pm 0.921	-8.681 \pm 1.481	0.464 \pm 0.429	0.796 \pm 0.038	0.838 \pm 0.109	0.331\pm0.274
2.0	✓	0.7	Target Score	85.168\pm21.258	-8.574\pm1.097	-9.514 \pm 1.028	-8.827\pm1.452	0.593\pm0.484	0.786 \pm 0.034	0.806 \pm 0.096	0.231 \pm 0.224
2.0	✓	0.7	Tempered Noise	84.969 \pm 18.906	-8.531 \pm 0.980	-9.809\pm1.170	-8.545 \pm 0.991	0.589 \pm 0.427	0.796\pm0.035	0.850 \pm 0.121	0.262\pm0.154

setting t_{\max} to a value in $[0.5, 0.7]$ generates molecules that are higher in quality compared to always resampling or no resampling for $\beta = 1.5$. For $\beta = 2.0$, the performance slightly decreases, but the diversity and uniqueness of the molecules is much higher at the end. Setting t_{\max} to 0.6 gives a good tradeoff in terms of generating molecules that perform well vs. maintaining diversity, and so we proceed with $\beta = 2.0$ and $t_{\max} = 0.6$ for the final experiments.

SDE Type Finally, we try using different types of SDEs. We find that at lower β , the Tempered Noise SDE performs better with and without FKC. At higher β , however, using the Tempered Noise SDE does not significantly change performance or decreases performance. Thus, for the main experiments, we proceed with the Target Score SDE.

Visualizing docked molecules In Fig. A6, we visualize the molecules with the highest docking scores to the protein pair GRM5/RRM1 (UniProt IDs P41594 and P23921, respectively) at each molecule size.

Correlation between FKC weights and target metric In Fig. A7, we plot the Spearman rank correlation of FKC weights and docking scores of the final generated molecules for two protein pair examples to understand whether sampling according to these weights will select for better molecules. We observe that for a protein pair where molecules improve with FKC resampling, there tends to be a positive correlation between the FKC weight and docking score (avg. $\rho = 0.275$ for an example protein pair), while this trend is less clear for a case where FKC resampling does not improve the molecule (avg. $\rho = 0.087$ for an example protein pair).

F.7. Molecule SMILES generation using latent diffusion models

We also investigate generating molecular SMILES strings conditioned on functional properties, which describe the desired function that the molecule should have. Molecules often need to possess multiple properties (e.g. bind to protein X and be non-toxic) (Wang et al., 2025). By controlling for these properties during the molecular generation process (as opposed to post-hoc filtering), we aim to increase the probability of discovering molecules that exhibit all desired characteristics, thereby improving the efficiency of hit identification.



Figure A6. Molecules generated from our method (target score SDE with $\beta = 2.0$ and FKC resampling) and baselines in the binding pock-
ets of two proteins: GRM5 (UniProt ID P41594) and RRM1 (UniProt ID P23921) for all 5 molecule sizes considered ($\{15, 19, 23, 27, 35\}$ atoms). Docking scores for each molecule and target are above each image; lower docking scores are better. The QED of the molecule is
above each model name. The binding pocket is shaded in light green.

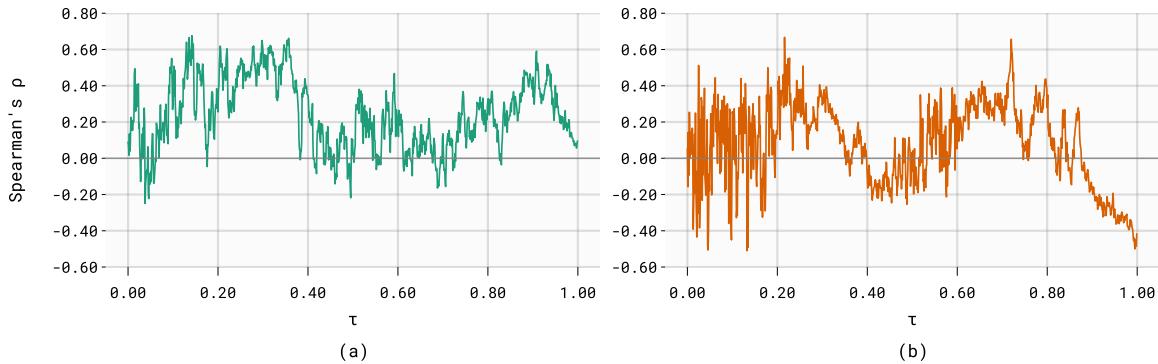


Figure A7. Spearman rank correlation coefficient (ρ) of FKC weights computed at each timestep τ and the docking score of the final molecule for (a) a protein pair where the final generated molecule improves with FKC and (b) a failure case where the molecule has a worse docking score than the baseline. Note that during inference, we only resample for $\tau \in [0.0, 0.6]$.

Table A6. Multi-property molecule generation results (PoE). For a set of two target properties (P_1 and 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 $P_1 * P_2$ 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.

P_1 / P_2	SDE Type	β	FKC	P_1 top-10 (\uparrow)	P_2 top-10 (\uparrow)	(P_1, P_2) top-1 (\uparrow)	Div. (\uparrow)	Val. & Uniq. (\uparrow)	Qual. (\uparrow)
JNK3	Target Score	0.5	✗	0.212 ± 0.016	0.356 ± 0.046	$(0.500, 0.580)$	0.910 ± 0.000	0.713 ± 0.027	0.127 ± 0.015
	Tempered Noise	1.5	✗	0.341 ± 0.039	0.468 ± 0.041	$(0.590, 0.560)$	0.881 ± 0.002	0.813 ± 0.025	0.352 ± 0.012
GSK3 β	Target Score	0.5	✓	0.342 ± 0.012	0.502 ± 0.034	$(0.500, 0.720)$	0.882 ± 0.002	0.832 ± 0.021	0.360 ± 0.021
	Tempered Noise	1.5	✗	0.342 ± 0.012	0.502 ± 0.034	$(0.500, 0.720)$	0.882 ± 0.002	0.832 ± 0.021	0.360 ± 0.021
JNK3	Target Score	0.5	✗	0.090 ± 0.018	0.434 ± 0.065	$(0.150, 0.472)$	0.915 ± 0.001	0.671 ± 0.022	0.228 ± 0.011
	Tempered Noise	1.5	✗	0.132 ± 0.032	0.550 ± 0.036	$(0.280, 0.469)$	0.884 ± 0.001	0.650 ± 0.021	0.258 ± 0.020
DRD2	Target Score	0.5	✓	0.141 ± 0.020	0.617 ± 0.040	$(0.360, 0.655)$	0.884 ± 0.005	0.661 ± 0.018	0.252 ± 0.014
	Tempered Noise	1.5	✗	0.141 ± 0.020	0.617 ± 0.040	$(0.360, 0.655)$	0.884 ± 0.005	0.661 ± 0.018	0.252 ± 0.014
GSK3 β	Target Score	0.5	✗	0.146 ± 0.034	0.528 ± 0.077	$(0.051, 0.908)$	0.914 ± 0.001	0.709 ± 0.021	0.203 ± 0.015
	Tempered Noise	1.5	✗	0.228 ± 0.016	0.649 ± 0.084	$(0.550, 0.655)$	0.884 ± 0.002	0.774 ± 0.015	0.303 ± 0.012
DRD2	Target Score	0.5	✓	0.266 ± 0.061	0.638 ± 0.036	$(0.520, 0.796)$	0.885 ± 0.002	0.774 ± 0.017	0.307 ± 0.012
	Tempered Noise	1.5	✓	0.266 ± 0.061	0.638 ± 0.036	$(0.520, 0.796)$	0.885 ± 0.002	0.774 ± 0.017	0.307 ± 0.012

Model 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.

Experiment setup: TDC oracles We consider three proteins oracles from Therapeutic Data Commons (TDC) (Huang et al., 2021): JNK3, GSK3 β , DRD2, which predicts whether or not a molecule binds to a protein. Note that while this task is similar in nature to the objective of SBDD, we are generating molecules conditioned on a functional text description instead of a 3D protein pocket. However, we could also consider other functional property descriptions, such as molecular solubility, toxicity, etc.

Prompts To generate molecules that inhibit a specific protein, we prompt the model with "This molecule inhibits {protein_name}", following Wang et al. (2025).

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 and synthetic accessibility (SA) ≤ 4.0 . This metric was taken from Lee et al. (2025b).

Results: TDC oracles We aim to generate molecules that satisfy the function of binding each protein when taking all combinations of the protein pairs. In Table A6, we show the best performance for each set of molecules and in Table A8 we ablate different SDE components. 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. We also note that PoE+FKC tends to generate more molecules that are unique, valid and are higher drug-like quality, although their diversity decreases slightly, which is a common tradeoff. In practice, we find that the FKC weights with the latent diffusion model have a large variance during molecule generation. This is problematic, as a large number of

Table A7. Docking scores of generated molecules to P_1 =ATP1A1 and P_2 =CPT2. We used the Tempered Noise SDE with $\beta = 1.5$ and generated 32 molecules.

FKC	(P_1, P_2) top-10 (\downarrow)	(P_1, P_2) top-1 (\downarrow)	Div. (\uparrow)
✗	$-6.65 \pm 1.05, -7.36 \pm 0.854$	$(-8.87, -8.13)$	0.921
✓	$(-7.49 \pm 0.71, -8.31 \pm 0.94)$	$(-8.41, -9.73)$	0.895

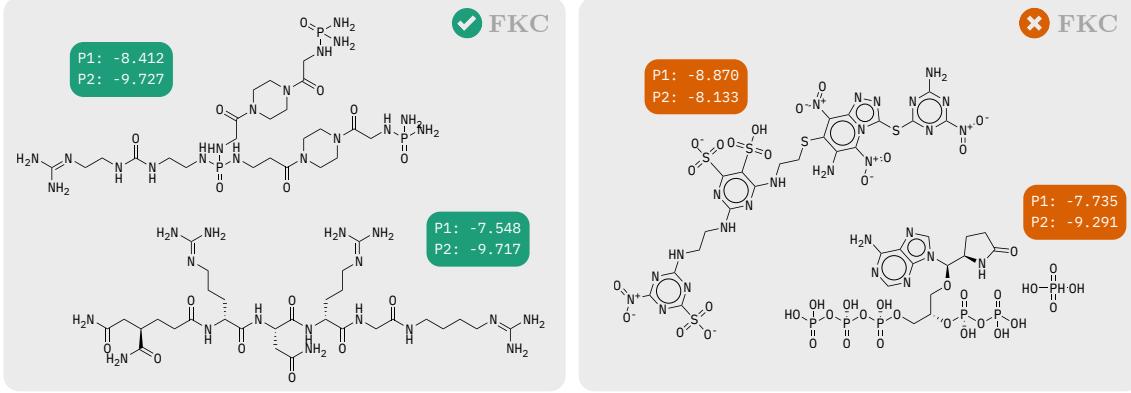


Figure A8. Molecules with best docking scores for binding to ATP1A1 (P_1) and CPT2 (P_2) from PoE with FKC (left) and without (right).

samples are thrown away. Furthermore, we noted that the score was not always well-conditioned. To ameliorate this, for all experiments using LDMol, 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).

Experiment setup: protein docking Finally, we consider a more challenging setting of protein-ligand docking, where we generate molecules using LDMol based on text-based prompts of binding to the proteins ATP1A1 (UniProt ID P05023) and CPT2 (UniProt ID P23786), and then evaluate them using docking. The protein pockets were obtained from Zhou et al. (2024) and the final generated molecules were docked using AutoDock Vina (Eberhardt et al., 2021).

Results: protein docking Table A7 shows the docking scores of molecules, and we find that incorporating FKC generates molecules with better scores. While ligands are typically generated using SBDD, we find it interesting that text-prompt generation is able to produce molecules that have reasonably good docking scores; known binders to ATP1A1 and CPT2 have docking scores of -8.168 and -9.174, respectively (Zhou et al., 2024). We visualize the top molecules in Fig. A8.

Table A8. Multi-property molecule generation results. For a set of two target properties (P_1 and P_2), we take the set of the top-10 best performing molecules as the molecules with the highest $P_1^*P_2$ 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).

P_1 P_2	SDE Type	β	FKC	P_1 top-10 (\uparrow)	P_2 top-10 (\uparrow)	(P_1, P_2) top-1 (\uparrow)	Div. (\uparrow)	Val. & Uniq. (\uparrow)	Qual. (\uparrow)
JNK3	Target Score	0.5	✗	0.212 ± 0.016	0.356 ± 0.046	(0.500, 0.580)	0.910 ± 0.000	0.713 ± 0.027	0.127 ± 0.015
	Tempered Noise	0.5	✗	0.225 ± 0.028	0.385 ± 0.042	(0.440, 0.690)	0.909 ± 0.001	0.723 ± 0.016	0.134 ± 0.006
	—	1.0	✗	0.289 ± 0.022	0.429 ± 0.018	(0.470, 0.580)	0.898 ± 0.002	0.811 ± 0.008	0.205 ± 0.011
	—	—	✓	0.342 ± 0.029	0.442 ± 0.051	(0.600, 0.650)	0.897 ± 0.002	0.804 ± 0.015	0.205 ± 0.015
GSK3 β	Target Score	—	✗	0.336 ± 0.031	0.484 ± 0.052	(0.480, 0.780)	0.886 ± 0.003	0.816 ± 0.013	0.336 ± 0.022
	Target Score	1.5	✓	0.351 ± 0.0340	0.447 ± 0.026	(0.590, 0.780)	0.886 ± 0.003	0.823 ± 0.024	0.356 ± 0.037
	Tempered Noise	—	✗	0.341 ± 0.039	0.468 ± 0.041	(0.590, 0.560)	0.881 ± 0.002	0.813 ± 0.025	0.352 ± 0.012
	Tempered Noise	—	✓	0.342 ± 0.012	0.502 ± 0.034	(0.500, 0.720)	0.882 ± 0.002	0.832 ± 0.021	0.360 ± 0.021
JNK3	Target Score	0.5	✗	0.090 ± 0.018	0.434 ± 0.065	(0.150, 0.472)	0.915 ± 0.001	0.671 ± 0.022	0.228 ± 0.011
	Tempered Score	0.5	✗	0.066 ± 0.015	0.571 ± 0.187	(0.110, 0.943)	0.914 ± 0.002	0.678 ± 0.0187	0.236 ± 0.020
	—	1.0	✗	0.087 ± 0.028	0.624 ± 0.094	(0.100, 0.978)	0.903 ± 0.001	0.675 ± 0.022	0.241 ± 0.010
	—	—	✓	0.094 ± 0.024	0.635 ± 0.067	(0.413, 0.550)	0.899 ± 0.002	0.686 ± 0.025	0.263 ± 0.023
DRD2	Target Score	—	✗	0.136 ± 0.046	0.582 ± 0.067	(0.490, 0.640)	0.886 ± 0.003	0.639 ± 0.019	0.241 ± 0.017
	Target Score	1.5	✓	0.102 ± 0.031	0.620 ± 0.148	(0.320, 0.541)	0.885 ± 0.006	0.659 ± 0.022	0.274 ± 0.028
	Tempered Noise	—	✗	0.132 ± 0.032	0.550 ± 0.036	(0.280, 0.469)	0.884 ± 0.001	0.650 ± 0.021	0.258 ± 0.020
	Tempered Noise	—	✓	0.141 ± 0.020	0.617 ± 0.040	(0.360, 0.655)	0.884 ± 0.005	0.661 ± 0.018	0.252 ± 0.014
GSK3 β	Target Score	0.5	✗	0.146 ± 0.034	0.528 ± 0.077	(0.051, 0.908)	0.914 ± 0.001	0.709 ± 0.021	0.203 ± 0.015
	Tempered Score	0.5	✗	0.162 ± 0.025	0.543 ± 0.063	(0.430, 0.965)	0.914 ± 0.001	0.697 ± 0.013	0.198 ± 0.017
	—	1.0	✗	0.202 ± 0.023	0.620 ± 0.057	(0.660, 0.726)	0.908 ± 0.002	0.773 ± 0.021	0.238 ± 0.021
	—	—	✓	0.190 ± 0.022	0.666 ± 0.093	(0.240, 0.986)	0.907 ± 0.002	0.784 ± 0.010	0.254 ± 0.019
DRD2	Target Score	—	✗	0.240 ± 0.030	0.636 ± 0.066	(0.350, 0.804)	0.894 ± 0.002	0.759 ± 0.015	0.290 ± 0.016
	Target Score	1.5	✓	0.222 ± 0.036	0.584 ± 0.068	(0.630, 0.580)	0.891 ± 0.003	0.740 ± 0.027	0.283 ± 0.020
	Tempered Score	—	✗	0.228 ± 0.016	0.649 ± 0.084	(0.550, 0.655)	0.884 ± 0.002	0.774 ± 0.015	0.303 ± 0.012
	Tempered Score	—	✓	0.266 ± 0.061	0.638 ± 0.036	(0.520, 0.796)	0.885 ± 0.002	0.774 ± 0.017	0.307 ± 0.012