Composition and Control with Distilled Energy Diffusion Models and Sequential Monte Carlo

James Thornton
Apple
Louis Béthune
Apple
Ruixiang Zhang
Apple
Arwen Bradley
Apple
Preetum Nakkiran
Apple
Shuangfei Zhai
Apple

Abstract

Diffusion models may be formulated as a time-indexed sequence of energy-based models, where the score corresponds to the negative gradient of an energy function. As opposed to learning the score directly, an energy parameterization is attractive as the energy itself can be used to control generation via Monte Carlo samplers. Architectural constraints and training instability in energy parameterized models have so far yielded inferior performance compared to directly approximating the score or denoiser. We address these deficiencies by introducing a novel training regime for the energy function through distillation of pre-trained diffusion models, resembling a Helmholtz decomposition of the score vector field. We further showcase the synergies between energy and score by casting the diffusion sampling procedure as a Feynman Kac model where sampling is controlled using potentials from the learnt energy functions. The Feynman Kac model formalism enables composition and low temperature sampling through sequential Monte Carlo.

1 Introduction

Diffusion models (Sohl-Dickstein et al., 2015; Song and Ermon, 2020; Song et al., 2021) have come to dominate the generative modelling landscape, exhibiting state of the art performance (Dhariwal and Nichol, 2021) across domains and modalities (De Bortoli et al., 2022), including self-supervised learning (Chen et al., 2024), natural science applications (Arts et al., 2023) and for optimal transport (Bortoli et al., 2021).

Despite the success of diffusion models, there are still a number of challenges. Firstly, diffusion models are known to have slow and expensive training (Jeha et al., 2024; Zhang et al., 2023). Secondly, effective conditioning of diffusion models remains an open challenge (Wu et al., 2024; Zhao et al., 2024b). Re-using, finetuning and adapting pretrained models has become an active area of research; both to overcome lengthy pretraining and to introduce additional conditioning not considered during training (Ye et al., 2024; Du et al., 2021b). In addition, many heuristic guidance weighting methods and prompt engineering techniques have been proposed to control generation. Such approaches are poorly understood (Bradley and Nakkiran, 2024), often require ad-hoc weighting and trial-and-error sampling to reach desired samples.

A diffusion model may be viewed as a sequence of time-indexed energy-based models, where the gradient of the energy is learnt rather than the energy itself (Song and Kingma, 2021; Salimans and Ho, 2021). It was shown in Du et al. (2023) how the energy interpretation of diffusion models may be used to better condition and compose pretrained diffusion models in order to generate novel distributions, such as composed distributions, rather than the default reverse process. Du et al. (2023) achieves this through Markov Chain Monte Carlo (MCMC) and annealed Langevin dynamics. Whilst a promising direction, energy-parameterised diffusions are inherently cumbersome to train and annealed MCMC sampling requires an excessive number of network evaluations.

Energy-parameterized diffusion models require computing multiple gradients through the energy function during training; first with respect to (w.r.t) the input state to recover a score, then w.r.t parameters for training. This exacerbates the already lengthy training entailed by denoising score-matching.

Contributions The purpose of this work is twofold. First to address training instability of energy-parameterized diffusion models, and secondly to introduce a new class of diffusion model samplers using the energy function for controllable generation within a Feynman Kac - Sequential Monte Carlo framework.

Proceedings of the 28[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

We summarize our key contributions as follows:

- We introduce a novel training procedure and parameterization to efficiently distill pretrained diffusion models into energy based models, whilst avoiding the high-variance loss of denoising score-matching. Our method can be interpreted as a *conservative projection* of a pretrained score.
- We showcase the merits of our approach in terms of generative performance, consistently achieving superior FID to prior energy-parameterised models for the datasets considered.
- We describe a general framework of how diffusion models may be used as the underlying Markov process of a Feynman Kac model (FKM); we detail how prior SMC based diffusions may be viewed as particular cases.
- Finally, we demonstrate how the energy function may be used to construct modality agnostic potentials within FKMs. This enables temperature controlled sampling, as well as composition of diffusion models via SMC.

2 Background

2.1 Diffusion Models

Consider data distribution p_{data} on support $\mathcal{X}$, and let the stochastic process $(\mathbf{X}_t)_{t=0}^T$ be given by the following dynamics; known as the forward process:

$$\mathrm{d}\mathbf{X}_t = f(t)\mathbf{X}_t\mathrm{d}t + g(t)\mathrm{d}\mathbf{B}_t, \quad \mathbf{X}_0 \sim p_0 := p_{\text{data}}, \quad (1)$$

with Brownian motion, $(\mathbf{B}_t)_{t\in[0,T]}$, drift $f : \mathbb{R} \to \mathbb{R}$ and scale $g : \mathbb{R} \to \mathbb{R}$ applied coordinate wise and let p_t denotes the marginal density of $\mathbf{X}_t$.

Diffusion models (Song et al., 2020, 2021; Ho et al., 2020) generate new samples by simulating from the stochastic process (2) with density denoted $q^\lambda_{0:T}$, initialization $\tilde{\mathbf{X}}_0 \sim p_T$ and $\tilde{\mathbf{B}}_t$ denotes another Brownian motion, independent to the forward process:

$$\mathrm{d}\tilde{\mathbf{X}}_t = \left[-f(t)\mathbf{X}_t + g^2(t)\tfrac{1+\lambda^2}{2}\nabla \log p_{T-t}(\tilde{\mathbf{X}}_t)\right]\mathrm{d}t + \lambda g(t)\mathrm{d}\tilde{\mathbf{B}}_t. \quad (2)$$

Parameter $\lambda > 0$ controls the degree of stochasticity within $q^\lambda_{0:T}$ (Zhang and Chen, 2021, 2022). In particular, for $\lambda = 1$, $p_{0:T} = q^1_{0:T}$, hence $q^1_{0:T}$ corresponds to the time-reversal of (1) (Haussmann and Pardoux, 1986; Anderson, 1965). Setting $\lambda = 0$, results in the probability flow ODE (Song et al., 2021). The marginal distributions of (2) match those of (1), i.e. $q^\lambda_t = p_t$, for all t and all $\lambda \geq 0$, hence (2) generates the data distribution $q^\lambda_0 = p_0 = p_{\text{data}}$.

Training. The score term, $\nabla \log p_t(\mathbf{X}_t)$, is generally intractable but can be expressed as the solution to a regression problem on the conditional score, then approximated by training a parameterized function s^*_θ. This is known as denoising score-matching (DSM) (Song and Ermon, 2019; Vincent, 2011):

$$s^*_\theta = \arg\min_{s_\theta} \mathbb{E}_{p_{0,t}}[\|s_\theta(\mathbf{X}_t, t) - \nabla_{x_t} \log p_{t|0}(\mathbf{X}_t|\mathbf{X}_0)\|^2].$$

Given the drift function in (1) is typically chosen to be linear in state $\mathbf{X}_t$ and applied coordinate-wise, then the forward process may be sampled in closed form using $\mathbf{X}_t|x_0 = \alpha_t x_0 + \sigma_t\epsilon$, $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$, for some time-indexed coefficients $\alpha_t, \sigma_t \in \mathbb{R}^+$, (Särkkä and Solin, 2019; Song et al., 2021). The conditional score $\nabla_{x_t} \log p_{t|0}(\mathbf{X}_t|\mathbf{X}_0)$ is therefore tractable. Alternatively, based on Tweedie's formula (Efron, 2011; Robbins, 1956): $\nabla \log p_t(x_t) = \sigma_t^{-2}(\alpha_t \mathbb{E}_{X_0|x_t}[\mathbf{X}_0|x_t] - x_t)$; one may approximate the expected denoiser $\mathbb{E}_{\mathbf{X}_0|x_t}[\mathbf{X}_0|x_t]$ via regression as in (3):

$$D^*_\theta = \arg\min_{D_\theta} \mathbb{E}_{p_{0,t}}[\|D_\theta(\mathbf{X}_t, t) - \mathbf{X}_0\|^2]. \quad (3)$$

Conditional Generation. Conditional generation is typically achieved via a conditional score, which can either be trained by DSM or decomposed into a unconditional and guidance term. The unconditional score can be pre-trained via DSM and the guidance term $\nabla \log p(y \mid x_t)$, which can be approximated in many ways such as via some classifier (Dhariwal and Nichol, 2021); classifier on a denoised state (Chung et al., 2023) or simply trained via denoising (Ho and Salimans, 2022; Denker et al., 2024):

$$\nabla \log p_t(x_t \mid y) = \nabla \log p_t(x_t) + \omega \nabla \log p_t(y \mid x_t). \quad (4)$$

Here $\omega \geq 1$ heuristically adjusts guidance strength, similar to temperature controlled sampling. Classifier-free guidance is the most commonly used training approach, estimating the guidance term by $\nabla \log p(y \mid x_t) = \nabla \log p(x_t \mid y) - \nabla \log p_t(x_t)$, where each individual term is trained via conditional DSM.

2.2 Energy Based Models

Energy-based models (EBMs) (LeCun et al., 2006) approximate density $p_{\text{data}} \approx p_\theta := Z^{-1}e^{-E_\theta}$ using θ parameterized potential $E_\theta(x)$, for normalising constant Z. The seminal work of Teh et al. (2003), later improved by Du et al. (2021b), introduced contrastive training of EBM by taking gradient steps on $-\mathbb{E}_{x\sim p_0}[E_\theta(x)] + \mathbb{E}_{x\sim p_\theta}[E_\theta(x)]$. Sampling p_θ typically entails expensive MCMC methods however.

Diffusion models as a sequence of energy based models. Given the idealised score, $\nabla \log p_t$, is a gradient one could avoid MCMC and use denoising score matching to learn a sequence of energy functions $(E_\theta(\cdot, t))_t$ such that $-\nabla_{x_t} E_\theta(x_t, t) \approx \nabla \log p_t(x_t)$, i.e. $s_\theta(x_t, t) := -\nabla_{x_t} E_\theta(x_t, t)$, henceforth referred to as

an energy-parameterisation. This is in contrast to the usual diffuson model parameterisation where score or denoiser is approximated directly with a neural network $s_\theta(x_t, t) \approx \nabla \log p_t(x_t)$. Energy parameterized diffusion models were first shown to be possible for image datasets in Salimans and Ho (2021) by careful choice of architecture, yet thus far remains noticeably inferior to unconstrained diffusion models.

2.3 Sequential Monte Carlo

Before delving into our method, we briefly recap Sequential Monte Carlo (SMC) (Doucet, 2001; Chopin and Papaspiliopoulos, 2020) for later use in Section 4. SMC entails propagating K particles initially sampled from some distribution M_0 through a sequence of proposal, importance weighting, and resampling steps. The resampling steps are crucial to ensuring computation is focused on promising particles, and to avoiding weight degeneracy. A simplified algorithm is presented in Algorithm 1, any resampling approaches could be used, in practice we use adaptive resampling (Del Moral et al., 2011) with the systematic resampler (Chopin and Papaspiliopoulos, 2020, Chapter 4).

SMC enables an approximate change of measure through the Feynman Kac model (FKM) framework (Chopin and Papaspiliopoulos, 2020; Del Moral, 2004). A FKM consists of an initial distribution M_0; some time indexed Markov transition kernels $(M_t)_t$, which we can sample from; and non-negative potential functions $(G_t)_t$, $G_0 : \mathcal{X} \to \mathbb{R}^+$, $G_t : \mathcal{X}^2 \to \mathbb{R}^+$.

$$M(\mathrm{d}x_{0:T}) = M_0(\mathrm{d}x_0) \prod_{t=1}^{T} M_t(\mathrm{d}x_t | x_{t-1}) \tag{5}$$

$$Q(\mathrm{d}x_{0:T}) \propto G_0(x_0) \prod_{t=1}^{T} G_t(x_t, x_{t-1}) M(\mathrm{d}x_{0:T}) \tag{6}$$

The use of potentials permits a change of measure from the proposal from Markov process (5) to the FKM distribution (6), where (6) can be approximately simulated with SMC or particle filtering as in Algorithm 1.

Algorithm 1 Generative SMC

Sample $\mathbf{X}_0^k \overset{\text{i.i.d.}}{\sim} M_0$ for $k \in [K]$
Weight $\omega_1^k = G_0(\mathbf{X}_0^k)$ for $k \in [K]$
for $t = 1, ..., T$ **do**
  Normalize weights $w_{t-1}^k \propto \omega_{t-1}^k$, $\sum_{k=1}^K w_{t-1}^k = 1$
  Resample $\tilde{\mathbf{X}}_{t-1}^k \sim \sum_{k=1}^K w_{t-1}^k \delta_{\mathbf{X}_{t-1}^k}$ for $k \in [K]$
  Proposal $\mathbf{X}_t^k \sim M(\cdot | \tilde{\mathbf{X}}_{t-1}^k)$ for $k \in [K]$
  Weight $\omega_t^k = G(\mathbf{X}_t^k, \tilde{\mathbf{X}}_{t-1}^k)$
end for
Return: samples $(\mathbf{X}_T^k)_k$

3 Distilled Energy Diffusion Models

3.1 Training Instability with Energy Parameterised Diffusion Models

Denoising score-matching (Vincent, 2011; Song and Ermon, 2019) is a promising simulation-free alternative to contrastive learning for training energy based models (Salimans and Ho, 2021; Song and Kingma, 2021). We call DSM with an energy-parameterised score E-DSM. Although E-DSM never quite learns exactly the energy at time $t = 0$; it can approximate the energy arbitrarily close and been used successfully in generative modelling (Du et al., 2023; Salimans and Ho, 2021) and sampling (Phillips et al., 2024).

Unfortunately, E-DSM suffers from training instability, as shown in Figure 2. We attribute this instability due two effects: firstly that DSM has a high variance loss (Jeha et al., 2024), and secondly network architecture limitations. Unlike in regular DSM or contrastive training, E-DSM requires taking gradients of E_θ with respect to both the state and parameters i.e. $\nabla_\theta \nabla_x E_\theta(x, t)$ during training. The effective network in E-DSM, $-\nabla_x E_\theta(x, t)$, may be viewed informally as the composition of $E_\theta : \mathbb{R}^d \to \mathbb{R}$ and operation $\nabla_x : \mathbb{R} \to \mathbb{R}^d$. The second operation, ∇_x lacks any normalisation or residual connections common in modern unconstrained neural networks typically used for diffusion models. Such stability measures have been shown to be crucial (Karras et al., 2024b, 2022) to ensuring stable training and generative performance.

We tackle both weaknesses by first providing a more stable loss, and secondly with careful parameterisation and initialization of the energy-diffusion network.

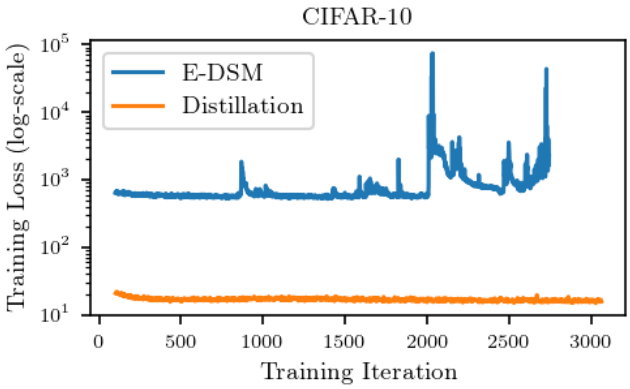


Figure 2: E-DSM loss (blue): $\mathbb{E}\|D_\theta(\mathbf{X}_t, t) - \mathbf{X}_0\|$ without gradient clipping vs distillation loss (orange): $\mathbb{E}\|D_\theta(\mathbf{X}_t, t) - D_\phi^{teach}(\mathbf{X}_t, t)\|$ during training of a diffusion model for CIFAR10. Initial 100 iterations cut.

3.2 Distillation Loss

We first introduce the conservative projection loss as:

$$\arg\min_{\theta} \mathbb{E}_{p_t}[\|\nabla u_\theta(\mathbf{X}_t, t) - v(\mathbf{X}_t, t)\|^2], \qquad (7)$$

where u_θ is some flexibly parameterised function, such as a neural network. A conservative vector field is the gradient of some potential. The loss (7) aims to learn a potential u_θ and hence conservative vector field ∇u_θ which is closest in squared Euclidean distance to v, not necessarily conservative.

Consider ODEs generated by v and the minimizer of (7), u_{θ^*}:

$$\mathrm{d}\mathbf{X}_t = v(\mathbf{X}_t, t)\mathrm{d}t \qquad \mathrm{d}\mathbf{X}_t = u_{\theta^*}(\mathbf{X}_t, t)\mathrm{d}t. \qquad (8)$$

Directly using (Liu, 2022, Theorem 5.2), if v is locally bounded, and (8)(left) has a unique solution generating density p_t, then the marginals of ODEs in (8) coincide. Minimising (7) may therefore be considered a type of Helmholtz decomposition of v, where the "rotation-only" component of v is removed, discussed further in Appendix A.

Note: a more general class of Bregman Helmoholtz losses have been considered in the seminal work of Liu (2022) in the context of rectified flows.

Distilling a Score into an Energy. In the context of training energy based models, we consider $u_\theta = -E_\theta$. Ideally we would use $v = \nabla \log p_t$, in such a case (7) would simply be (non-denoising) score-matching (Hyvärinen, 2005). We do not have access to $\nabla \log p_t$ so instead use a pre-trained score-function s_ϕ^{teach}, i.e. $v(\mathbf{X}_t, t) = s_\phi^{teach}$ as proxy:

$$\arg\min_{\theta} \mathbb{E}_{p_{0,t}}[\|\nabla E_\theta(\mathbf{X}_t, t) + s_\phi^{teach}(\mathbf{X}_t, t)\|^2]. \qquad (9)$$

By the arguments above, we do not require s_ϕ^{teach} be conservative, as the minimizer of (9) will generate the same distribution as s_ϕ^{teach} via the probability flow ODE and hence generate a distribution close to the data distribution, if s_ϕ^{teach} is well trained.

Expressed as Denoising. The loss (9) may equivalently be written as a denoising loss with target $D_\phi^{teach}(x_t, t) \approx \mathbf{E}[\mathbf{X}_0|x_t]$, as in (10) where again by Tweedie's formula, one may write $s_\phi^{teach}(xt, t)$ in terms of denoiser $D_\phi^{teach}(x_t, t) = \alpha_t^{-1}[x_t + \sigma_t^2 s_\phi^{teach}(x_t, t)]$.

$$\arg\min_{D_\theta} \mathbb{E}_{p_{0,t}}[\|D_\theta(\mathbf{X}_t, t) - D_\phi^{teach}(\mathbf{X}_t, t)\|^2]. \qquad (10)$$

The corresponding distilled denoiser is related to the energy through $D_\theta(x_t, t) = \alpha_t^{-1}[x_t\sigma_t^2 - \nabla E_\theta(x_t, t)]$.

Losses (10) and (3) differ only in the regression target: $D_\phi^{teach}(\mathbf{X}, t) \approx \mathbf{E}[\mathbf{X}_0|x_t]$ vs $\mathbf{X}_0$ respectively. Given $\mathbf{E}[\mathbf{X}_0|x_t]$ is the minimizer of (3) and at large time t, $\mathbf{E}[\mathbf{X}_0|x_t]$ is quite different to $\mathbf{X}_0$, and hence intuitively targeting $D_\phi^{teach}(\mathbf{X}, t) \approx \mathbf{E}[\mathbf{X}_0|x_t]$ directly results in more stable training than (3).

Motivated by the desire to reduce training instability, we have framed this distillation loss in terms of learning energy-parameterised diffusion models; however, the same loss could also be applied to train unconstrained diffusion models through distillation.

Scores are approximately conservative. Similar to Lai et al. (2023), we observe that well-trained score networks are approximately conservative, except at close to $t = 0$, where the score exhibits a Lipschitz singularity (Yang et al., 2023), see Figure 4. Here *conservativity* of a score network may be quantified by the asymmetry of its Jacobian (see Appendix A).

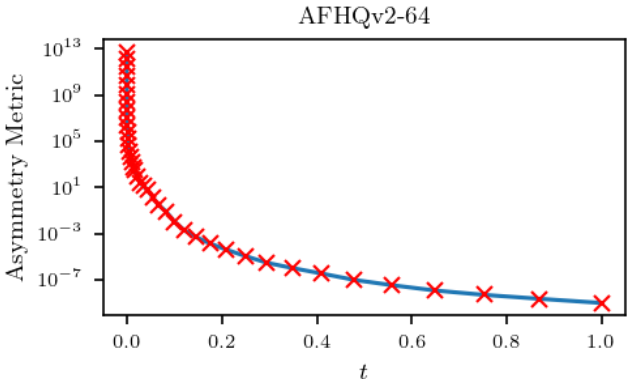


Figure 4: Asymmetry metric in log-scale, $\|\mathbf{J} - \mathbf{J}^T\|^2$ where Jacobian $\mathbf{J} = \mathbf{D}_x s_\theta(x_t, t)$ of score network s_θ trained via DSM on AFHQv2-64, $t \in [0, 1]$.

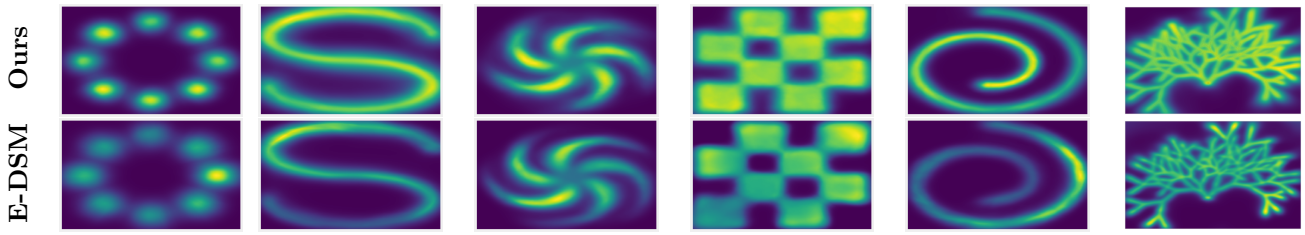


Figure 1: Density plot of $p_E \propto \exp -E_\theta(x_t, t)$, where E_θ is trained via distillation of pre-trained diffusion networks (Ours) (top) vs via energy parameterized denoising score matching (E-DSM).

3.3 Parameterization and Initialization

Pre-conditioning. To further reduce training instability of energy-parameterised diffusion models; we use the preconditioning ($c_{\text{s}}, c_{out}, c_{\text{in}}, c_{\text{t}}$) of Karras et al. (2022), parameterizing the denoiser, D_θ as:

$$D_\theta(x_t, t) = c_{\text{s}}(t)x_t + c_{out}(t)\nabla_x F_\theta(c_{\text{in}}(t)x_t, c_{\text{t}}(t)).$$

We replace the unconstrained network in Karras et al. (2022) with the gradient of network F_θ. By Tweedie (Efron, 2011), we compute the energy via:

$$E_\theta(x_t, t) = \frac{1-\alpha_t c_{\text{s}}(t)}{2\sigma_t^2}\|x_t\|^2 - \frac{\alpha_t c_{out}(t)}{c_{\text{in}}(t)\sigma_t^2} F_\theta(c_{\text{in}}(t)x_t, c_{\text{t}}(t)).$$

Network parameterization. We parameterize the network F_θ such that its gradient takes a similar network structure to score/ denoising networks. Consider a network architecture, h_θ, known to work well for DSM, such as a U-Net for image-based diffusion models. The following parameterization forces $\nabla_x F_\theta$ to resemble $h_\theta(x_t, t)$ with the addition of a residual term, where $\mathbf{D}_x h_\theta$ denotes Jacobian of h_θ:

$$F_\theta(x_t, t) = h_\theta(x_t, t) \cdot x_t \tag{11}$$

$$\nabla_x F_\theta(x_t, t) = x_t \cdot \mathbf{D}_x h_\theta(x_t, t) + h_\theta(x_t, t) \tag{12}$$

This structure is important for initialization.

Network initialization. Motivated by recent successes in initializing models with pre-trained diffusions (Lee et al., 2024; Kim et al., 2024) we do the same for our energy-parameterized network. In particular, we set h_θ to be the same architecture as the teacher network D_ϕ^{teach} and initialize with teacher parameters $\theta \leftarrow \phi$. This simple technique is effective due to choice in parameterization (11), and results in drastically faster convergence (see Appendix E) and better generative performance overall.

4 Composition and Control

4.1 Feynman Kac Diffusion Models

As discussed in Section 2, the Feynman Kac model (FKM) formalism (Chopin and Papaspiliopoulos, 2020; Del Moral, 2004) is a simple yet principled framework to perform an approximate change of measure from a given Markov process according to a user-provided potentials. Given diffusion models are expensive to train, it is desirable to use pretrained models.

Denote the discretization of the generative diffusion process (2) as $q^\lambda(x_{t_{0:N}})$, and the network approximation as $q_\theta^\lambda(x_{t_{0:N}})$ in (13) for any ODE/ SDE solver on discretisation $T = t_0 \geq t_1 \geq \ldots \geq t_N = 0$. We choose the underlying Markov measure for a FKM to be q_θ^λ or similarly, some conditioned process $q_\theta^\lambda(\cdot|y)$ for conditioning signal y e.g. labels. Explicit forms of $q_\theta^\lambda(x_{t_{i+1}}|x_{t_i})$ are given in Appendix C.

$$q_\theta^\lambda(x_{t_{0:N}}) = q_{t_0}^\lambda(x_{t_0}) \prod_{i=0}^{N} q_\theta^\lambda(x_{t_{i+1}}|x_{t_i}) \tag{13}$$

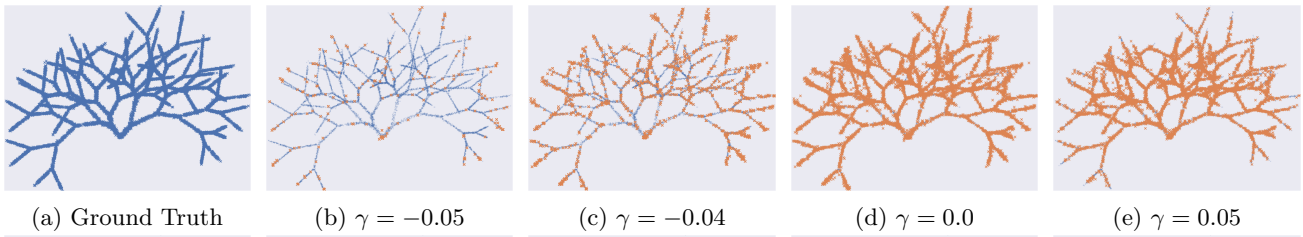
The intermediate FKM distributions by running Algorithm 1 with potentials $(G_i)_i$ and Markov process (13) for $n \leq N$ are then given by:

$$Q(x_{t_{0:n}}) \propto q_\theta^\lambda(x_{t_{0:n}}) G_0(x_{t_0}) \prod_{i=1}^{n} G_i(x_{t_i}, x_{t_{i-1}}) \tag{14}$$

4.2 Temperature-Controlled Generation

Let $\gamma_t \in \mathbb{R}$ be some time-indexed inverse temperature parameter. Given access to density p_{t_i}, one could set $G_i(x_{t_i}, x_{t_{i-1}}) = p(x_{t_i})^{\gamma_{t_i}}$. Rearranging (13) gives:

$$Q(x_{t_{0:N}}) \propto \prod_{i=0}^{N} p(x_{t_i})^{\gamma_{t_i}} q^\lambda(x_{t_{0:N}}).$$



(a) Ground Truth (b) $\gamma = -0.05$ (c) $\gamma = -0.04$ (d) $\gamma = 0.0$ (e) $\gamma = 0.05$

(f) $\gamma = 0.1$ (g) $\gamma = 0.2$ (h) $\gamma = 0.5$ (i) $\gamma = 1.0$ (j) $\gamma = 5.0$

Figure 3: **SMC Sampling of Feynman Kac Diffusion Models** for $G_i(x_{t_i}, x_{t-1}) = \exp\{-\gamma_{t_i} E_\theta(x_{t_i}\} \approx p(x_{t_i})^\gamma_{t_i}$. The fractal distribution, inspired by Karras et al. (2024a), is obtained by fitting Gaussian mixtures to each branch and appending recursively. Ground truth samples shown in (faded) blue and generated samples shown in orange.

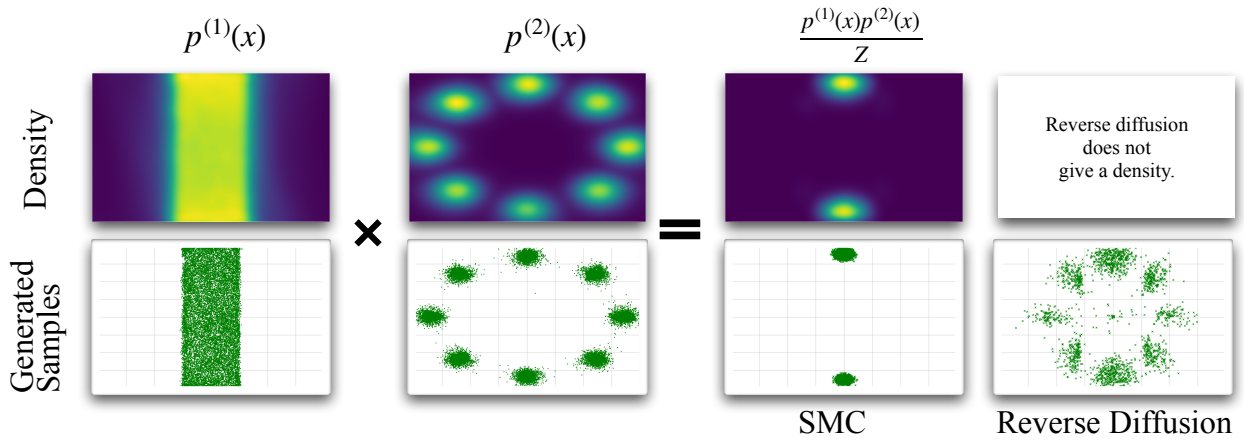


Figure 5: Simple 2D Composition failure from Du et al. (2023). Top row: Learnt densities, $e^{-E_\theta^{(i)}}$ for each $p_t^{(i)}$ and $e^{-E_\theta^{(1)}-E_\theta^{(2)}}$. Bottom row: generated samples per $p_t^{(i)}$, as well as the SMC generation using (16) and reverse diffusion of summed scores, $q_{\theta,\lambda}^{(1)+(2)}(x_{t_{0:N}})$.

Recall from Section 2, regardless of choice of λ, the marginals of the backward and forward process match, $q^\lambda(x_t) = p(x_t)$ for any t, hence:

$$Q(x_{t_{0:N}}) \propto p(x_{t_N})^{1+\gamma} q^\lambda(x_{t_{0:N-1}}|x_{t_N}) \prod_{i=0}^{N-1} p(x_{t_i})^\gamma$$

Figure 3 illustrates how setting $\gamma \geq 0$ results a more concentrated distribution, $p(x_{t_N})^{1+\gamma}$. Similarly, $\gamma < 0$ results lower density regions being sampled with greater probability. This is biased toward rare events.

In practice, we substitute $G_i(x_{t_i}, x_{t-1}) \leftarrow e^{-\gamma_{t_i} E_\theta(x_{t_i})}$ as an approximation, proportional to $p(x_{t_i})_{t_i}^\gamma$.

It is common to proxy low temp. generation with high guidance weights, as detailed in Section 2. Our SMC approach provides an alternative which may be used with unconditional, conditional models. An example demonstrating low-temperature generation is shown in Section 5.4 for conditional image generation.

4.3 Compositional Generation

EBMs enable composition of pretrained models as logical operators can be expressed as functions of the score and energy (Du et al., 2020a; Liu et al., 2022), see Appendix D. We focus here on the AND operation.

Unlike for EBMs the summed scores from diffusion models at $t > 0$ do not always match the score for the composed distribution (Du et al., 2023). Consider time indexed densities $p_t^{(1)}$ and $p_t^{(2)}$, and denote the composed score:

$$s_t^{(1)+(2)}(x_t, t) = \nabla \log p_t^{(1)}(x_t) + \nabla \log p_t^{(2)}(x_t) \quad (15)$$

Let $q_\lambda^{(1)+(2)}(x_{t_{0:N}})$ denote the process (2) with composed score (15), and consider FKM given by (16).

$$M_t(x_{t_{i+1}}|x_{t_i}) = q_{t_0}^\lambda(x_{t_0}) \prod_{i=0}^{N} q_\lambda^{(1)+(2)}(x_{t_{i+1}}|x_{t_i}) \quad (16)$$
$$M_t(x_{t_{i+1}}|x_{t_i}) G_i(x_{t_{i+1}}, x_{t_i}) = p_t^{(1)}(x_{t_{i+1}}) p_t^{(2)}(x_{t_{i+1}}).$$

In practice we use the approximation $q_{\theta,\lambda}^{(1)+(2)}(x_{t_{0:N}})$ defined by summing the trained score functions for $p_t^{(1)}$ and $p_t^{(2)}$, and similarly approximate, up to a scalar, each $p_t^{(i)}$ with $e^{-E_\theta^{(i)}(\cdot,t)}$. Expanding the FKM intermediate distributions for this choice of G in (14) gives a sequence of product densities coinciding with the target density we wish to generate. A simple example of this is illustrated in Figure 5, note that the density is given by the energy function, and hence reverse diffusion does not provide a density directly.

4.4 Bounded Generation

We consider how constraints (Lou and Ermon, 2023; Fishman et al., 2024, 2023) and dynamic thresholding (Saharia et al., 2022) can be imposed via FKM potentials. Fishman et al. (2024) adjusts sampling by rejecting transitions if proposals x_{t_i} fall outside a specified region, B. This resembles a FKM with potential $G_i(x_{t_i}, x_{t_{i-1}}) = \mathbb{I}_B(x_{t_i})$. Similarly, dynamic thresholding entails clipping the denoiser to be within a unit-cube at generation time. This also resembles a FKM setting $G_i(x_{t_i}, x_{t_{i-1}}) = \mathbb{I}_{[-1+\delta,1-\delta]^d}(D_\theta(x_t, t))$, $\delta > 0$.

The above choices of regions are simple, but quite crude. One could instead use the energy as a softer alternative, for example by choosing $G_i(x_{t_i}, x_{t_{i-1}}) = \exp\{-\gamma_t E_\theta(x_t, t)\}$ or $G_i(x_{t_i}, x_{t_{i-1}}) = \exp\{-\gamma_t E_\theta(D_\theta(x_t, t), \epsilon)\}$, $\epsilon \approx 0$. See generated examples in Appendix B.

5 Experiments

Full experimental details including network architectures and training recipes are provided in Appendix E.

5.1 2D Experiments

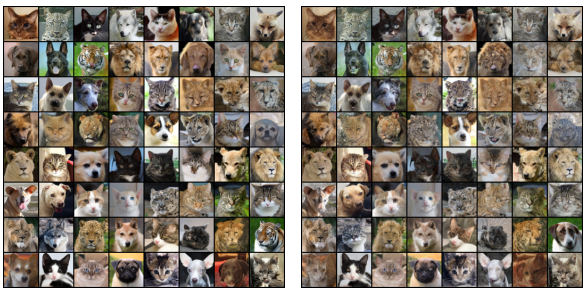
E-DSM vs Distillation. Figure 1 provides qualitative comparison of our distillation approach compared to E-DSM, We used a feedforward network with sine nonlinearity. There is a general uneven density exhibited within E-DSM, not present in our approach.

Temperature Controlled Generation. Figure 3 illustrates the effects of temperature on the fractal dataset. We are able to generate either the high density regions in low temperature regime, and to target low density regions (at the boundary of the support) with higher temperatures.

Compositional Generation. Figure 5 demonstrates the failure of reverse diffusion in composition. Our SMC based composition approach however correctly recovers the joint distribution $p_1(x)p_2(x)$. A similar result has been observed in Du et al. (2023), where they correct with MCMC rather than SMC.

5.2 Generative Performance of Diffusion-Energy Models

We first verify our improvements for training energy-parameterized diffusion models in terms of generative performance, measured by Frechet Inception Distance (FID) (Heusel et al., 2017) on standard EBM datasets: CIFAR-10 (Krizhevsky et al., 2009) and CelebA-64 (Liu et al., 2015) for both unconditional generation in Table 1 and conditional generation in Table 2. In the interest of transparency, we display the Number of Function Evaluations (NFE) involved in the generative process. We further showcase our method with AFHQv2-64 (Choi et al., 2020), FFHQ-64 (Karras et al., 2019) in Table 3 and visually in Figure 6. We compare to recent energy-parameterized approaches (Gao et al., 2020; Zhu et al., 2023; Hill et al., 2023; Schröder et al., 2024), E-DSM, and popular methods such as diffusion (Karras et al., 2022) and flow-matching (Lipman et al., 2022; Liu, 2022). Note: here we do not use SMC but standard generation with the gradient of the learnt energy.



(a) Distilled, Ours (b) E-DSM

Figure 6: Samples from AFHQv2-64 using energy parameterized diffusion models trained with our distillation method vs denoising score-matching (E-DSM)

Our method exhibits significantly better performance than other baselines using an energy-parameterization, particularly for CIFAR10 and AFHQv2. Although we achieve only modest performance improvement verses E-DSM on face-datasets, we note that the E-DSM results took significantly longer to train and required careful selection of learning rate and gradient clipping for stability, see Appendix E.

Table 1: Unconditional performance by NFE and FID for CIFAR-10 and CelebA. * denotes our result.EDM was used for the teacher score.

Method	CIFAR-10		CelebA-64	
	NFE	FID	NFE	FID
Diffusion / Flow				
EDM* Karras et al. (2022)	35	2.21	79	1.89
FM* Liu (2022) Lipman et al. (2022)	100	2.96	100	3.05
Energy				
EDLEBM Schröder et al. (2024)	500	73.58	500	36.73
CDLEBM Pang et al. (2020)	500	70.15	500	37.87
DRL Gao et al. (2020)	180	9.58	180	5.98
HDEBM Hill et al. (2023)	75	8.06	115	4.13
E-DSM*	35	6.17	79	2.87
CDRL Zhu et al. (2023)	90	4.31	−−	−−
Ours*	35	3.01	79	2.60

Table 2: Conditional performance by NFE and FID for CIFAR-10 and CelebA.* denotes our result. EDM was used for the teacher score.

Method	CIFAR-10		CelebA-64	
	NFE	FID	NFE	FID
Diffusion / Flow				
EDM* Karras et al. (2022)	35	1.90	79	1.88
FM* Liu (2022) Lipman et al. (2022)	100	2.69	100	2.95
Energy				
E-DSM*	35	4.49	79	2.03
Ours*	35	2.71	79	1.95

Table 3: Unconditional performance by NFE and FID for AFHQv2 and FFHQ 64. * denotes our result. EDM was used for the teacher score.

Method	AFHQ-64		FFHQ-64	
	NFE	FID	NFE	FID
Diffusion / Flow				
EDM* Karras et al. (2022)	79	2.35	79	2.61
FM* Liu (2022) Lipman et al. (2022)	100	2.73	100	3.30
Energy				
E-DSM*	79	4.57	79	2.71
Ours*	79	3.88	79	2.64

5.3 Composition of Image Models

As detailed in Appendix E, we train separate energy-parameterized diffusion models for subsets of the CelebA dataset with attributes *male* and *glasses*, $p_t^{(\text{male})}$ and $p_t^{(\text{glasses})}$. We compose pretrained models via SMC detailed in Section 4.3, using potential $G_i = \left(\frac{p_{t_i}^{(\text{male})} p_{t_i}^{(\text{glasses})}}{M_{t_i}}\right)^{\gamma_{t_i}}$ with schedule $(\gamma_t)_t$ to control diversity. Figure 7 shows a uniform subsample

Figure 7: Composition: Male AND Glasses.

from generated batch of size 64, qualitatively showing our method works for image datasets. Repetition indicates resampling can reduce diversity however.

5.4 Low Temperature Sampling

We train a conditional energy-diffusion model on CelebA, generate 128 samples for multiple conditions using a base sampler and via low temperature (temp.) SMC with inverse temp. parameter $\gamma_t = 0.1$, then assess images-condition adherence using a CLIP score, detailed in Appendix E. Table 4 shows the superior performance of low temp. SMC sampling for condition adherence.

6 Discussion

6.1 Related Prior Work

Training Energy Based Models. A number of recent works aim to improve EBM training. Zhu et al. (2023) uses a diffusion model to reduce the number of Langevin steps within the recovery likelihood approach of Gao et al. (2020). Schröder et al. (2024) eliminates the need for MCMC and ∇_x-computation of the energy during training by using a contrastive loss with forward noising process instead of MCMC, coined Energy Divergence (ED). ED is a promising alternative to E-DSM and has connections to score-matching, but, as shown in Table 1, it is not yet competitive, and suffers from a bias by choice of noisy energy function.

Composition with MCMC. Similar to our work, Du et al. (2023) also uses energy-parameterized diffusion models but performs controlled generation with MCMC rather than SMC. SMC is known to suffer weight degeneracy high dimension, resulting in lack of diversity across particles, MCMC does not suffer from this, though requires additional non-parallel steps which is time consuming. The approaches are however complementary, and indeed one may perform MCMC after resampling steps to promote diversity.

Sequential Monte Carlo in Diffusion Models. Many recent works use SMC within diffusion models for conditional generation, we detail the FKM formulations of these works in in Appendix B.

Wu et al. (2024) uses twisted SMC with a classifier-guided proposal (Dhariwal and Nichol, 2021) and potentials approximated with diffusion posterior sampling (Chung et al., 2023), which has been detailed as a FKM by concurrent work (Zhao et al., 2024b). Cardoso et al. (2024) and Dou and Song (2024) tackle linear inverse problems where potentials have a closed form using Gaussian conjugacy. Li et al. (2024) perform SMC for both discrete and continuous diffusion models whereby potentials consist of a reward function applied to $\mathbb{E}[\mathbf{X}_0|x_t]$.

Liu et al. (2024) corrects conditional generations using an adversarially trained density ratio potential, and scales this to text-to-image models.

Table 4: Measuring condition adherence with CLIP.

Condition	Low Temp.	Base
(1): Man with black hair	0.75	0.71
(2): Blonde woman, lipstick	0.63	0.53
(3): Smiling old man	0.95	0.85
(4): Young woman, no hair	0.97	0.91
(5): Man with make-up	0.40	0.35

SMC for LLMs. SMC is not only popular within diffusion models, but has been successful within large language models (LLMs) (Lew et al., 2023; Zhao et al., 2024a). Lew et al. (2023) uses a FKM formulation with indicator based potential functions similar to as detailed in Section 4.4, and Zhao et al. (2024a) discuss using SMC for text using potentials from reward functions or learning such potentials via contrastive twist learning, similar to contrastive learning for EBMs.

6.2 Concurrent work

Since submission/ acceptance of our work [1], there have been a number of relevant concurrent works.

FKM Interpretation. Singhal et al. (2025) similarly to Zhao et al. (2024b) and this work, detail sampling diffusion models in terms of KFM. Singhal et al. (2025) follow Li et al. (2024) in using reward functions based potentials but focus on text-to-image reward, and explore further heuristics such as or combining rewards via sum or max; and sampling $\mathbf{X}_0|x_t$ via nested diffusion (Elata et al., 2024) as input to their reward rather than using $\mathbb{E}[\mathbf{X}_0|x_t]$ as done in Li et al. (2024).

SMC for discrete diffusion. Lee et al. (2025) use SMC for low temperature sampling for discrete diffusion models. Xu et al. (2024) use a pretrained autoregressive likelihood model applied to samples $\mathbf{X}_0|x_t$ for a potential within discrete diffusion sampling.

Composition. Skreta et al. (2024) construct a cheap density estimator by simulating from an SDE, which can be computed at sampling time if using reverse diffusion solver, though it is not clear if this can be used in conjunction with resampling and Langevin corrector schemes. Skreta et al. (2024) then use this estimator to perform composition-type sampling, however their logical AND appears to differ from other more commonly used logical AND operations, in that it targets samples with equal probability between classes rather than generating both classes, e.g. "a CAT and a DOG" results in a cat/dog hybrid optical illusion rather than a separate cat and separate dog in one image.

Bradley et al. (2025) explore composition more formally, establishing types of composition and cases where summing scores is sufficient without need for SMC correction as performed in this work or with MCMC correction from (Du et al., 2023).

6.3 Limitations

Multiple-networks. Our distillation loss requires access to pretrained models. Exploring multi-headed networks for joint training of both score and energy were may be an interesting direction to pursue, this would avoid the need for pretrained networks and reduce NFE at sampling time.

Diversity. Resampling may result in a loss of diversity for poorly constructed potentials and proposals. Investigating approximate resampling techniques which preserve diversity such as in (Corenflos et al., 2021; Ma et al., 2020; Zhu et al., 2020) may be practical mitigation strategy.

Mixing of Scores. Score-matching and hence our distillation loss may suffer practical issues for supports with isolated components, resulting in learning the incorrect mixing proportions (Wenliang and Kanagawa, 2020) - known as the blindness of score-matching. Whilst this may not pose an issue in our setting for $t > 0$ due to Gaussian noise connecting the support, there may be issues in using energy functions trained with score matching very close to $t = 0$.

6.4 Future Considerations

Scale and modalities. Whilst we have successfully demonstrated our methods on medium size image datasets, we are yet to verify the performance on other modalities or larger datasets. A first step would be to apply this to latent space (Vahdat et al., 2021; Rombach et al., 2022) for higher resolution images. Similarly, the energy function is modality agnostic and could be applied to other fields such as molecular dynamics, (Arts et al., 2023).

Other application of the energy function. There are a plethora of applications requiring the energy worth exploring, for example the NEGATION or UNION operations also require access to time-indexed densities (Du et al., 2020b; Koulischer et al., 2024). Similar to classical EBMs, our learnt energy functions may also be used for unsupervised learning (Du et al., 2021a) and reasoning tasks (Du et al., 2022).

The benefit of conservative scores. Although conservative score approximations are not strictly necessary for generative modeling (Horvat and Pfister, 2024), our method does enable one to learn SOTA performant diffusion models with strictly conservative scores. Conservative scores have been remarked as crucial in molecular dynamics (Arts et al., 2023); as well as provide attractive theoretical properties (Daras et al., 2024) in terms of generalization.

Given the pursuit optimal transport (OT) has attracted a lot of attention (Bortoli et al., 2021; Thornton et al., 2022; Liu et al., 2023; Shi et al., 2023), it is worth remarking that strictly conservative drifts are required to recover OT with ReFlow (Liu, 2022).

[1] Submission October 2024

Acknowledgements

We thank Rob Brekelmans for feedback on an early draft of this paper and Kevin Li for references (Horvat and Pfister, 2024; Wenliang and Kanagawa, 2020) and fruitful discussion on the advantages/ disadvantages of conservative scores.

Bibliography

D. G. Anderson. Iterative procedures for nonlinear integral equations. *Journal of the ACM (JACM)*, 12(4):547–560, 1965.

M. Arts, V. Garcia Satorras, C.-W. Huang, D. Zugner, M. Federici, C. Clementi, F. Noé, R. Pinsler, and R. van den Berg. Two for one: Diffusion models and force fields for coarse-grained molecular dynamics. *Journal of Chemical Theory and Computation*, 19 (18):6151–6159, 2023.

V. D. Bortoli, J. Thornton, J. Heng, and A. Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 2021.

J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/jax-ml/jax`.

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

A. Bradley, P. Nakkiran, D. Berthelot, J. Thornton, and J. M. Susskind. Mechanisms of projective composition of diffusion models. *arXiv preprint arXiv:2502.04549*, 2025.

G. Cardoso, Y. J. el idrissi, S. L. Corff, and E. Moulines. Monte carlo guided denoising diffusion models for bayesian linear inverse problems. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=nHESwXvxWK`.

X. Chen, Z. Liu, S. Xie, and K. He. Deconstructing denoising diffusion models for self-supervised learning. *arXiv preprint arXiv:2401.14404*, 2024.

Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020.

N. Chopin and O. Papaspiliopoulos. *An Introduction to Sequential Monte Carlo*. Springer, 2020.

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

A. Corenflos, J. Thornton, G. Deligiannidis, and A. Doucet. Differentiable particle filtering via entropy-regularized optimal transport. In *International Conference on Machine Learning*, pages 2100–2111. PMLR, 2021.

G. Daras, Y. Dagan, A. Dimakis, and C. Daskalakis. Consistent diffusion models: Mitigating sampling drift by learning to be consistent. *Advances in Neural Information Processing Systems*, 36, 2024.

V. De Bortoli, E. Mathieu, M. Hutchinson, J. Thornton, Y. W. Teh, and A. Doucet. Riemannian score-based generative modeling. *Advances in Neural Information Processing Systems*, 2022.

P. Del Moral. *Feynman-kac formulae*. Springer, 2004.

P. Del Moral, A. Doucet, and A. Jasra. On adaptive resampling procedures for sequential monte carlo methods. hal-inria rr-6700-2008. *Bernoulli*, pages 2496–2534, 2011.

A. Denker, F. Vargas, S. Padhy, K. Didi, S. Mathis, V. Dutordoir, R. Barbano, E. Mathieu, U. J. Komorowska, and P. Lio. Deft: Efficient fine-tuning of conditional diffusion models by learning the generalised h-transform. *arXiv preprint arXiv:2406.01781*, 2024.

P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

Z. Dou and Y. Song. Diffusion posterior sampling for linear inverse problem solving: A filtering perspective. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=tplXNcHZs1`.

A. Doucet. Sequential monte carlo methods in practice, 2001.

Y. Du, S. Li, and I. Mordatch. Compositional visual generation with energy based models. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6637–6647. Curran Associates, Inc., 2020a.

Y. Du, S. Li, and I. Mordatch. Compositional visual generation with energy based models. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6637–6647. Curran Associates, Inc., 2020b. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/49856ed476ad01fcff881d57e161d73f-Paper.pdf`.

Y. Du, S. Li, Y. Sharma, J. Tenenbaum, and I. Mordatch. Unsupervised learning of compositional energy concepts. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 15608–15620. Curran Associates, Inc., 2021a. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/838aac83e00e8c5ca0f839c96d6cb3be-Paper.pdf.

Y. Du, S. Li, J. Tenenbaum, and I. Mordatch. Improved contrastive divergence training of energy-based models. In *International Conference on Machine Learning*, pages 2837–2848. PMLR, 2021b.

Y. Du, S. Li, J. Tenenbaum, and I. Mordatch. Learning iterative reasoning through energy minimization. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5570–5582. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/du22d.html.

Y. Du, C. Durkan, R. Strudel, J. B. Tenenbaum, S. Dieleman, R. Fergus, J. Sohl-Dickstein, A. Doucet, and W. S. Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. In *International Conference on Machine Learning*, pages 8489–8510. PMLR, 2023.

B. Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106 (496):1602–1614, 2011. doi: 10.1198/jasa.2011.tm11181. URL https://doi.org/10.1198/jasa.2011.tm11181. PMID: 22505788.

N. Elata, B. Kawar, T. Michaeli, and M. Elad. Nested diffusion processes for anytime image generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5018–5027, 2024.

N. Fishman, L. Klarner, V. De Bortoli, E. Mathieu, and M. J. Hutchinson. Diffusion models for constrained domains. *Transactions on Machine Learning Research*, 2023.

N. Fishman, L. Klarner, E. Mathieu, M. Hutchinson, and V. De Bortoli. Metropolis sampling for constrained diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

R. Gao, Y. Song, B. Poole, Y. N. Wu, and D. P. Kingma. Learning energy-based models by diffusion recovery likelihood. *arXiv preprint arXiv:2012.08125*, 2020.

U. G. Haussmann and E. Pardoux. Time reversal of diffusions. *The Annals of Probability*, 14(4):1188–1205, 1986.

M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

M. Hill, Z. Xuan, Y. Han, and G.-J. Qi. Tackling unconditional generation for highly multimodal distributions with hat diffusion EBM, 2023. URL https://openreview.net/forum?id=4hiJ3KPDYe.

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

J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models, 2020.

C. Horvat and J.-P. Pfister. On gauge freedom, conservativity and intrinsic dimensionality estimation in diffusion models. *arXiv preprint arXiv:2402.03845*, 2024.

A. Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005. URL http://jmlr.org/papers/v6/hyvarinen05a.html.

P. Jeha, W. Grathwohl, M. R. Andersen, C. H. Ek, and J. Frellsen. Variance reduction of diffusion model's gradients with taylor approximation-based control variate. *arXiv preprint arXiv:2408.12270*, 2024.

T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

T. Karras, M. Aittala, T. Aila, and S. Laine. Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364*, 2022.

T. Karras, M. Aittala, T. Kynkäänniemi, J. Lehtinen, T. Aila, and S. Laine. Guiding a diffusion model with a bad version of itself. *arXiv preprint arXiv:2406.02507*, 2024a.

T. Karras, M. Aittala, J. Lehtinen, J. Hellsten, T. Aila, and S. Laine. Analyzing and improving the training dynamics of diffusion models. In *Computer Vision and Pattern Recognition (CVPR)*, 2024b.

B. Kim, Y.-G. Hsieh, M. Klein, M. Cuturi, J. C. Ye, B. Kawar, and J. Thornton. Simple reflow: Improved techniques for fast flow models. *arXiv preprint arXiv:2410.07815*, 2024.

F. Koulischer, J. Deleu, G. Raya, T. Demeester, and L. Ambrogioni. Dynamic negative guidance of diffusion models. *arXiv preprint arXiv:2410.14398*, 2024.

A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

C.-H. Lai, Y. Takida, N. Murata, T. Uesaka, Y. Mitsufuji, and S. Ermon. Fp-diffusion: Improving score-based diffusion models by enforcing the underlying score fokker-planck equation. In *International Conference on Machine Learning*, pages 18365–18398. PMLR, 2023.

Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, F. Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

C. K. Lee, P. Jeha, J. Frellsen, P. Lio, M. S. Albergo, and F. Vargas. Debiasing guidance for discrete diffusion with sequential monte carlo. *arXiv preprint arXiv:2502.06079*, 2025.

S. Lee, Z. Lin, and G. Fanti. Improving the training of rectified flows. *arXiv preprint arXiv:2405.20320*, 2024.

A. K. Lew, T. Zhi-Xuan, G. Grand, and V. K. Mansinghka. Sequential monte carlo steering of large language models using probabilistic programs. *arXiv preprint arXiv:2306.03081*, 2023.

X. Li, Y. Zhao, C. Wang, G. Scalia, G. Eraslan, S. Nair, T. Biancalani, S. Ji, A. Regev, S. Levine, et al. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. *arXiv preprint arXiv:2408.08252*, 2024.

Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

G.-H. Liu, A. Vahdat, D.-A. Huang, E. A. Theodorou, W. Nie, and A. Anandkumar. I2sb: Image-to-image schrodinger bridge. *arXiv preprint arXiv:2302.05872*, 2023.

N. Liu, S. Li, Y. Du, A. Torralba, and J. B. Tenenbaum. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, pages 423–439. Springer, 2022.

Q. Liu. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.14577*, 2022.

Y. Liu, Y. Zhang, T. Jaakkola, and S. Chang. Correcting diffusion generation through resampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8713–8723, 2024.

Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

A. Lou and S. Ermon. Reflected diffusion models. In *International Conference on Machine Learning*, pages 22675–22701. PMLR, 2023.

X. Ma, P. Karkus, D. Hsu, and W. S. Lee. Particle filter recurrent neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5101–5108, 2020.

B. Pang, T. Han, E. Nijkamp, S.-C. Zhu, and Y. N. Wu. Learning latent space energy-based prior model. *Advances in Neural Information Processing Systems*, 33:21994–22008, 2020.

A. Phillips, H.-D. Dau, M. J. Hutchinson, V. De Bortoli, G. Deligiannidis, and A. Doucet. Particle denoising diffusion sampler. *arXiv preprint arXiv:2402.06320*, 2024.

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

H. E. Robbins. An empirical bayes approach to statistics. 1956. URL `https://api.semanticscholar.org/CorpusID:26161481`.

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

C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

T. Salimans and J. Ho. Should EBMs model the energy or the score? In *Energy Based Models Workshop - ICLR 2021*, 2021. URL `https://openreview.net/forum?id=9AS-TF2jRNb`.

S. Särkkä and A. Solin. *Applied stochastic differential equations*, volume 10. Cambridge University Press, 2019.

T. Schröder, Z. Ou, J. Lim, Y. Li, S. Vollmer, and A. Duncan. Energy discrepancies: a score-independent loss for energy-based models. *Advances in Neural Information Processing Systems*, 36, 2024.

M. Seitzer. pytorch-fid: FID Score for PyTorch. `https://github.com/mseitzer/pytorch-fid`, August 2020. Version 0.3.0.

Y. Shi, V. De Bortoli, A. Campbell, and A. Doucet. Diffusion schr\" odinger bridge matching. *arXiv preprint arXiv:2303.16852*, 2023.

R. Singhal, Z. Horvitz, R. Teehan, M. Ren, Z. Yu, K. McKeown, and R. Ranganath. A general frame-

work for inference-time scaling and steering of diffusion models. *arXiv preprint arXiv:2501.06848*, 2025.

M. Skreta, L. Atanackovic, A. J. Bose, A. Tong, and K. Neklyudov. The superposition of diffusion models using the itô density estimator. *arXiv e-prints*, pages arXiv–2412, 2024.

J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015.

J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution, 2020.

Y. Song and D. P. Kingma. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

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

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

Y. W. Teh, M. Welling, S. Osindero, and G. E. Hinton. Energy-based models for sparse overcomplete representations. *Journal of Machine Learning Research*, 4(Dec):1235–1260, 2003.

J. Thornton, M. Hutchinson, E. Mathieu, V. De Bortoli, Y. W. Teh, and A. Doucet. Riemannian diffusion schr\"odinger bridge. *Continuous Time Methods for Machine Learning, International Conference of Machine Learning,*, 2022.

A. Vahdat, K. Kreis, and J. Kautz. Score-based generative modeling in latent space. *arXiv preprint arXiv:2106.05931*, 2021.

P. Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7): 1661–1674, 2011.

L. K. Wenliang and H. Kanagawa. Blindness of score-based methods to isolated components and mixing proportions. *arXiv preprint arXiv:2008.10087*, 2020.

L. Wu, B. Trippe, C. Naesseth, D. Blei, and J. P. Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

M. Xu, T. Geffner, K. Kreis, W. Nie, Y. Xu, J. Leskovec, S. Ermon, and A. Vahdat. Energy-based diffusion language models for text generation. *arXiv preprint arXiv:2410.21357*, 2024.

Z. Yang, R. Feng, H. Zhang, Y. Shen, K. Zhu, L. Huang, Y. Zhang, Y. Liu, D. Zhao, J. Zhou, et al. Lipschitz singularities in diffusion models. In *The Twelfth International Conference on Learning Representations*, 2023.

H. Ye, H. Lin, J. Han, M. Xu, S. Liu, Y. Liang, J. Ma, J. Zou, and S. Ermon. Tfg: Unified training-free guidance for diffusion models. *arXiv preprint arXiv:2409.15761*, 2024.

H. Zhang, Y. Lu, I. Alkhouri, S. Ravishankar, D. Song, and Q. Qu. Improving efficiency of diffusion models via multi-stage framework and tailored multi-decoder architectures. *arXiv preprint arXiv:2312.09181*, 2023.

Q. Zhang and Y. Chen. Diffusion normalizing flow. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 16280–16291. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/876f1f9954de0aa402d91bb988d12cd4-Paper.pdf`.

Q. Zhang and Y. Chen. Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902*, 2022.

S. Zhao, R. Brekelmans, A. Makhzani, and R. Grosse. Probabilistic inference in language models via twisted sequential monte carlo. *arXiv preprint arXiv:2404.17546*, 2024a.

Z. Zhao, Z. Luo, J. Sjölund, and T. B. Schön. Conditional sampling within generative diffusion models. *arXiv preprint arXiv:2409.09650*, 2024b.

M. Zhu, K. Murphy, and R. Jonschkowski. Towards differentiable resampling. *arXiv preprint arXiv:2004.11938*, 2020.

Y. Zhu, J. Xie, Y. Wu, and R. Gao. Learning energy-based models by cooperative diffusion recovery likelihood. *arXiv preprint arXiv:2309.05153*, 2023.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. Yes.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. Yes.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. Not Applicable.
2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. Yes.
   (b) Complete proofs of all theoretical results. Yes.
   (c) Clear explanations of any assumptions. Yes.
3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). Not Applicable.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). Yes.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). Yes.
   (d) A description of the computing infrastructure used (e.g., type of GPUs, internal cluster, or cloud provider). Yes.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator if your work uses existing assets. Yes.
   (b) The license information of the assets, if applicable. Yes.
   (c) New assets either in the supplemental material or as a URL, if applicable. Yes.
   (d) Information about consent from data providers/curators. Not Applicable.
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. Not Applicable.
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. Not Applicable.
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. Not Applicable.
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. Not Applicable.

A On the conservativity of score networks

A.1 Conservative Projection

The conservative projection detailed in Section 3 results in the Helmholtz decomposition of the pre-trained score vector field. Generally, consider a vector field $v_t : \mathcal{X} \to \mathcal{X}$, providing certain conditions hold (Liu, 2022), then one may decompose $v_t(x_t) = \nabla_x f_t(x_t) + r_t(x_t)$ where $r_t : \mathcal{X} \to \mathcal{X}$, $f_t(x_t) : \mathcal{X} \to \mathbb{R}$ and $\nabla \cdot r_t = 0$.

Liu (2022, Theorem 5.2) proves a more general case coined the Bregman Helmholtz decomposition for convex $c : \mathcal{X} \to \mathbb{R}$ and conjugate $c^*(x) := \sup_y\{x \cdot y - c(y)\}$, that the optimal f^* of (17) for vector field v_t:

$$\inf_f \int c(v_t(X_t)) - v_t(X_t) \cdot g_t(X_t) + c^*(g_t(X_t))\mathrm{d}t \qquad g_t = \nabla c^* \odot \nabla f_t \tag{17}$$

yields an orthogonal decomposition: $v_t = \nabla c^* \odot \nabla f_t^* + r_t$ where r_t is measure preserving. This implies that given the same initialization stochastic processes defined by vector fields v_t and $\nabla c^* \odot \nabla f_t^*$ have the same marginals.

Our particular case (7) is a specific case of minimizing (17) for squared Euclidean cost $c(x) = c^*(x) = \frac{1}{2}\|x\|^2$; $\nabla c^*(x) = x$. Ideally, if it were available we would use $v_t(x_t) = \nabla \log p_t(x_t)$; but in practice we use $v_t(x_t) = s_\theta(x_t, t)$, for some pre-trained score function s_θ. This specific case of flow matching on a score vector field is also remarked in (Liu, 2022, Sec 5), but in a different context.

A.2 Measuring conservativity

By Poincare's star shaped lemma, if a function $g : \mathbb{R}^d \to \mathbb{R}^d$ on star-shaped support has a symmetric Jacobian, then there exists function $F : \mathbb{R}^d \to \mathbb{R}$ such that $g = \nabla F$. Let $\mathbf{J}$ denote some Jacobian matrix. The Jacobian is typically too large and expensive to compute in full, instead the asymmetry of the Jacobian may be approximated efficiently using Hutchinson's trace estimator: $\|\mathbf{J} - \mathbf{J}^T\|^2 = \mathbb{E}_{\nu\sim\mathcal{N}(\mathbf{0},\mathbb{I})}\|\nu^T\mathbf{J} - \mathbf{J}\nu\|^2$,

$$\begin{aligned}
\text{trace}\left((\mathbf{J} - \mathbf{J}^T)^T(\mathbf{J} - \mathbf{J}^T)\right) &= \mathbb{E}_{\nu\sim\mathcal{N}(\mathbf{0},\mathbb{I})}\nu^T\left((\mathbf{J} - \mathbf{J}^T)^T(\mathbf{J} - \mathbf{J}^T)\right)\nu \\
&= \mathbb{E}_{\nu\sim\mathcal{N}(\mathbf{0},\mathbb{I})}\|\nu^T(\mathbf{J} - \mathbf{J}^T)\|^2 \\
&= \mathbb{E}_{\nu\sim\mathcal{N}(\mathbf{0},\mathbb{I})}\|\nu^T\mathbf{J} - \nu^T\mathbf{J}^T\|^2 \\
&= \mathbb{E}_{\nu\sim\mathcal{N}(\mathbf{0},\mathbb{I})}\|\nu^T\mathbf{J} - \mathbf{J}\nu\|^2
\end{aligned}$$

where $\nu^T\mathbf{J}$ and $\mathbf{J}\nu$ may be computed efficiently with vector-Jacobian and Jacobian-vector products. A Monte Carlo approximation is used in Figure 4 $\frac{1}{n}\sum_{i=1}^n \|\nu_i^T\mathbf{J} - \mathbf{J}\nu_i\|^2$, and Figure 8 is normalized.

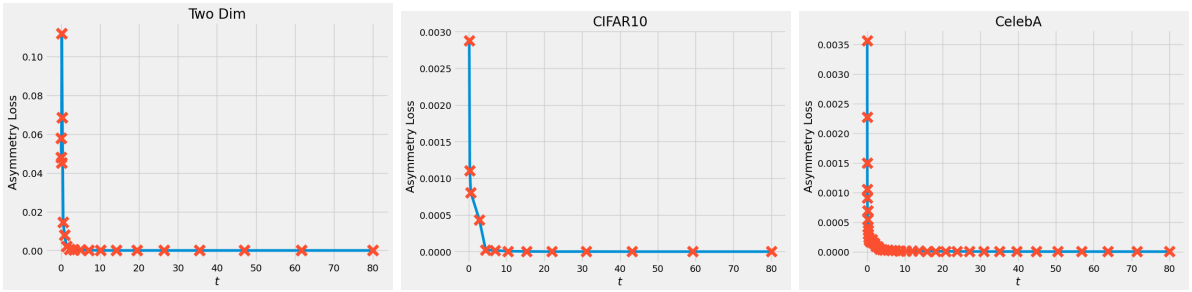


Figure 8: Asymmetry metric approximation $\frac{\|\mathbf{J}-\mathbf{J}^T\|^2}{\|\mathbf{J}\|^2} \approx \frac{\sum_{i=1}^n \|\nu_i^T\mathbf{J}-\mathbf{J}\nu_i\|^2}{\sum_{i=1}^n \|\nu_i^T\mathbf{J}\|^2}$ for $\nu_i \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$ where $\mathbf{J} = \mathbf{D}_x s_\theta(x_t, t)$ of score network s_θ trained via DSM on 2D spiral (left) CIFAR10 (middle) and CelebA (right).

B Feynman Kac Model Potentials

We detail here a few other Feynman Kac Model (FKM) potentials, alluded to in Section 3.

B.1 Bounded Generation

By setting $G_i(x, y) = \mathbb{I}_B(x)$, one forces the generative trajectory to be within region $B \subset \mathcal{X}$.

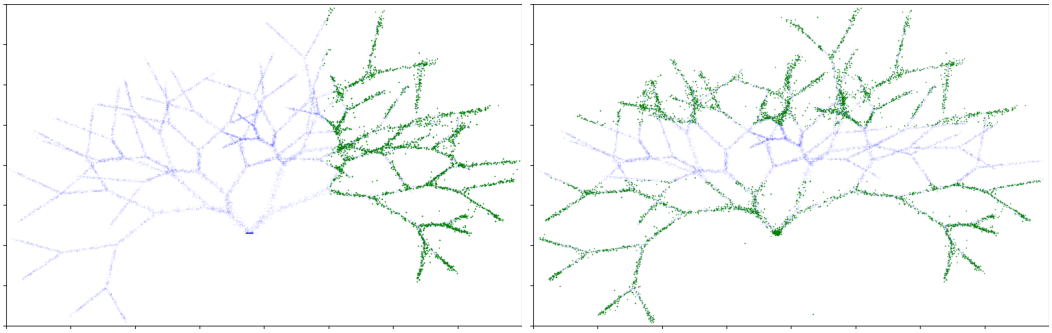


Figure 9: SMC sampling of 2D fractal dataset with FKM potential $G_i(x, y) = \mathbb{I}_B(x)$. Left: $B = [0.25, 1] \times \mathbb{R}$. Right: $B = \mathbb{R} \times [0.25, 1] \cup \times[-1., -0.1]$. Here blue faded dots are the ground truth data and green points are generated by SMC for the FKM.

B.2 Twisted SMC for Conditional Generation

Wu et al. (2024) consider the setting of a pretrained unconditional diffusion model and with to sample from a conditional model. They first use classifier guidance whereby the guidance term is approximated using DPS (Chung et al., 2023), then use SMC to correct guided diffusion models. In the ideal setting the corresponding FKM may be expressed as:

$$M(x_{t_{i+1}}|x_{t_i}) = q^\lambda(x_{t_{i+1}}|x_{t_i}) \tag{18}$$

$$G_0(x_{t_0}) = p(y|x_{t_0}) \tag{19}$$

$$G_t(x_{t_{i+1}}, x_{t_i}) = \frac{p(y|x_{t_{i+1}})q^\lambda(x_{t_{i+1}}|x_{t_i})}{p(y|x_{t_i})q^\lambda(x_{t_{i+1}}|x_{t_i}, y)} \tag{20}$$

where $p(y|x_t)$ denotes a classifier on noisy data for label y.

Wu et al. (2024) assumes the setting where one does not have access to $p(y|x_t)$ or $q^\lambda(x_{t_{i+1}}|x_{t_i}, y)$ but has access to a trained classifier $p_\theta(y|x_0)$ on data without noise; a rained denoiser D_θ, found from the score model, and trained unconditional score model s_θ to approximate $q^\lambda(x_{t_{i+1}}|x_{t_i})$ as follows:

$$p_\theta^{DPS}(y|x_{t_i}) = p_\theta(y|D_\theta(x_{t_i}, t_i)) \tag{21}$$

$$s_\theta^{DPS}(x_{t_i}, t_i, y) = s_\theta(x_{t_i}, t_i) + \nabla_{x_t} \log p_\theta^{DPS}(y|x_{t_i}) \tag{22}$$

$$q_\theta^{DPS,\lambda}(x_{t_{i+1}}|x_{t_i}, y) = \mathcal{N}(x_{t_i} + \Delta_i \frac{\lambda^2 + 1}{2}[-f(t_i)x_{t_i} + g^2(t_i)s_\theta^{DPS}(x_{t_i}, t_i, y)], \Delta_i \lambda^2 g(t_i)^2 \mathbb{I}) \tag{23}$$

This yields the guided FKM model:

$$M(x_{t_{i+1}}|x_{t_i}) = q_\theta^{DPS,\lambda}(x_{t_{i+1}}|x_{t_i}, y) \tag{24}$$

$$G_0(x_{t_0}) = p_\theta^{DPS}(y|x_{t_0}) \tag{25}$$

$$G_t(x_{t_{i+1}}, x_{t_i}) = \frac{p_\theta^{DPS}(y|x_{t_{i+1}})q_\theta^\lambda(x_{t_{i+1}}|x_{t_i})}{p_\theta^{DPS}(y|x_{t_i})q_\theta^{DPS,\lambda}(x_{t_{i+1}}|x_{t_i}, y)}. \tag{26}$$

C Sampling Diffusion Models

Let the step size be denoted $\Delta_i = (t_{i+1} - t_i)$. The simplest implementation of the reverse process (2) is:

$$q_\theta^\lambda(x_{t_{i+1}}|x_{t_i}) = \mathcal{N}(x_{t_i} + \Delta_i[-f(t_i)x_{t_i} + g^2(t_i)\frac{\lambda^2 + 1}{2}s_\theta(x_{t_i}, t_i)], \Delta_i \lambda^2 g(t_i)^2 \mathbb{I}) \tag{27}$$

There exist other more advanced solvers including DDIM (stochastic) (Song et al., 2020) and a plethora of ODE solvers and higher order SDE and ODE solvers such as the Heun solver used in Karras et al. (2022).

D Composition

Logical compositional operations - *AND*, *OR*, *NOT* - of EBMs may be implemented by transforming the density for EBMs as follows (Du et al., 2023, 2020b; Liu et al., 2022):

AND: $p^{\text{AND}} \propto \prod_i p^{(i)}$. **OR**: $p^{\text{OR}} \propto \sum_i p^{(i)}$. **NOT**: $p^{\text{NOT}} \propto \frac{p^{(1)}}{(p^{(2)})^{\alpha}}$, for some weight $\alpha > 0$, depending on density.

Hence one may approximately sample the densities corresponding to each operation via Langevin dynamics. The score is sufficient for **AND** operations, but the density itself is needed for **OR** and **NOT** operations. However, due to score matching learning unnormalised densities, summing such densities for **OR** composition may be theoretically problematic. In practice this can be resolved with heuristic weighting.

As noted by Du et al. (2023), arithmetic operations of scores at $t > 0$ do not in general recover the noisy score corresponding to the composed scores at $t = 0$; i.e. $p_t^{\text{AND}} = \prod_i p_t^{(i)} \neq \int (\prod_i p^{(i)}) \mathrm{d}p_{t|0}$.

This can lead to a failure in reverse diffusion for compositional generation. One may instead use annealed Langevin dynamics to target p_t^{AND} and then gradually anneal $t \to 0$ to get approximate samples of p_0^{AND} (Du et al., 2023).

In this work we propose an alternative but complementary approach to annealed Langevin dynamics, in using SMC to target a sequence of distributions which also gradually converge to the desired composition. We specifically target the *AND* operation, but other operations can be targeted similarly. Here the density is required for all logical operations.

Consider schedule $(\gamma_{t_i})_{i=1}^N$:, where $\gamma_{t_N} = 1$ and FKM model with potentials:

$$G_i = \frac{(p_{t_i}^{(1)} p_{t_i}^{(2)})^{\gamma_{t_i}} M_{t_i}^{1-\gamma_{t_i}}}{M_{t_i}}. \tag{28}$$

The resulting FKM distributions are

$$Q(x_{t_{0:N}}) \propto M_{t_{0:N}}(x_{t_{0:N}}) G_0(x_{t_0}) \prod_{i=1}^N G_i(x_{t_i}|x_{t_{i-1}}) = \prod_{i=1}^N (p_{t_i}^{(1)} p_{t_i}^{(2)})^{\gamma_{t_i}} M_{t_i}^{1-\gamma_{t_i}} = p_{t_N}^{(1)} p_{t_N}^{(2)} \left[\prod_{i=1}^{N-1} (p_{t_i}^{(1)} p_{t_i}^{(2)})^{\gamma_{t_i}} M_{t_i}^{1-\gamma_{t_i}} \right], \tag{29}$$

hence recovers the desired marginal $p_{t_N}^{(1)} p_{t_N}^{(2)}$ at time t_N.

E Experimental Details

E.1 Training Details

Compute. All experiments were carried out using $A100$ 40GB GPUs on single nodes of up to 8 GPUs.

Training Parameters For pre-training via DSM, the learning rates 0.0002 was used for AFHQv2, CelebA and FFHQ, learning rate 0.001 is used for CIFAR10. The same learning rates were used for distilling the denoisers into energy-parameterised models.

For E-DSM, 0.0002 learning rate was used for CIFAR10, with gradient clipping gradients above norm of 10, other E-DSM clipping was at norm of 1.

All experiment used a linear warm-up learning rate schedule for 10_000 steps.

CIFAR10 experiments used batch size 512, all other image experiments used batch size 256.

EMA was held constant at a rate of 0.9992 for all experiments.

Network Parameters: We used the *SongNet* NCSNPP network as per (Karras et al., 2022; Song et al., 2021) with the following hyper parameters:

Table 5: Network Parameters.

Parameter	**CIFAR10-32**	**AFHQV2-64**	**FFHQ-64**	**CelebA-64**
normalization	"GroupNorm"	"GroupNorm"	"GroupNorm"	"GroupNorm"
nonlinearity	"swish"	"swish"	"swish"	"swish"
nf	128	128	128	128
ch_mult	[2, 2, 2]	[1, 2, 2, 2]	[1, 2, 2, 2]	[1, 2, 2, 2]
num_res_blocks	4	4	4	4
attn_resolutions)	(16,)	(16,)	(16,)	(16,)
resamp_with_conv	True	True	True	True
fir	True	True	True	True
fir_kernel	[1, 3, 3, 1]	[1, 3, 3, 1]	[1, 3, 3, 1]	[1, 3, 3, 1]
skip_rescale	True	True	True	True
resblock_type	"biggan"	"biggan"	"biggan"	"biggan"
progressive	"none"	"none"	"none"	"none"
progressive_input	"residual"	"residual"	"residual"	"residual"
progressive_combine	"sum"	"sum"	"sum"	"sum"
attention_type	"ddpm"	"ddpm"	"ddpm"	"ddpm"
embedding_type	"fourier"	"fourier"	"fourier"	"fourier"
init_scale	0.0	0.0	0.0	0.0
fourier_scale	16	16	16	16
conv_size	3	3	3	3
num_scales	18	18	18	18
dropout	0.13	0.25	0.05	0.1

Sampling. Generative modelling results using energy parameterized diffusions table 2, table 1, table 3 follow (Karras et al., 2022) using the Heun solver on the ODE where $\lambda = 0$.

For low temperature sampling we can also set $\lambda = 0$ given the marginals for all λ coincide, we do not require to divide by the transition density in the FKM potential. For compositional sampling where we require transition density, we use $\lambda = 1$.

Evaluation. Frechet Inception distance (Heusel et al., 2017) was used as the evaluation metric based on Inceptionv3 features (Szegedy et al., 2016) using a re-implementation based on (Seitzer, 2020).

E.2 Composition

We train two separate models on subsets of the data, which may overlap for labels "Eyeglasses=1" and "Male=1". We then run SMC using the weighted compositional FKM detailed in Appendix D, with $\gamma_t = 0.01$; to avoid collapsing to a single sample due to weight degeneracy at time $t = 0$; we do not resample for $t < 0.1$. It has been observed (Karras et al., 2024a) that conditioning primarily occurs in the middle of the diffusion trajectory and towards $t = 0$ all visual features are present but the image is simply sharpened.

The full batch of 64 images, which were then sub-sampled to show in the main document in Figure 7 are given below:

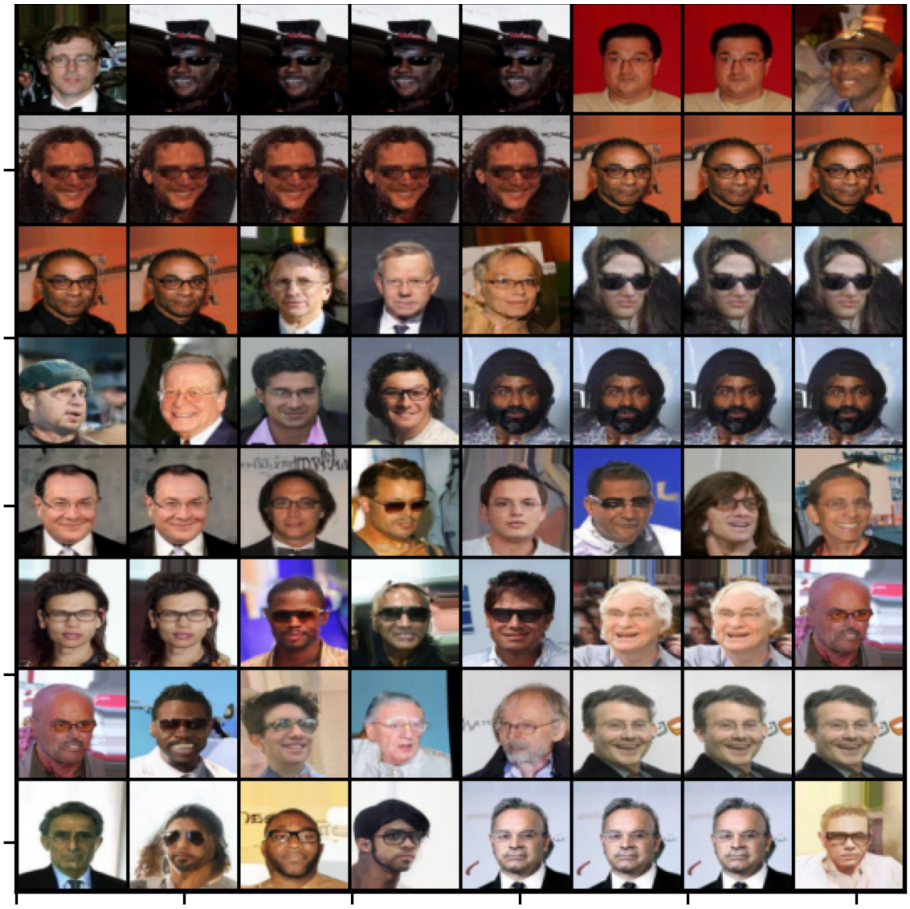


Figure 10: Composition: Male AND Glasses.

E.3 Low Temperature Generation

In order to test condition adherence we perform regular sampling of (2) with $\lambda = 0$ and then using SMC with low temperature weighting setting $\gamma_t = 0.1$. We consider the following 5 cases in the Table 6.

Table 6: CelebA Attribute Conditions.

Attribute	**1**	**2**	**3**	**4**	**5**
5_o_Clock_Shadow	-1	-1	-1	-1	-1
Arched_Eyebrows	-1	-1	-1	-1	-1
Attractive	1	1	-1	1	1
Bags_Under_Eyes	-1	-1	-1	-1	-1
Bald	1	-1	-1	-1	-1
Bangs	-1	-1	-1	-1	-1
Big_Lips	-1	-1	-1	1	-1
Big_Nose	-1	-1	-1	-1	-1
Black_Hair	-1	1	-1	-1	1
Blond_Hair	-1	-1	-1	-1	-1
Blurry	-1	-1	-1	-1	-1
Brown_Hair	-1	-1	-1	-1	-1
Bushy_Eyebrows	-1	-1	-1	-1	-1
Chubby	-1	-1	-1	-1	-1
Double_Chin	-1	-1	-1	-1	-1
Eyeglasses	-1	-1	-1	-1	-1
Goatee	-1	-1	-1	-1	-1
Gray_Hair	-1	-1	1	-1	-1
Heavy_Makeup	-1	1	-1	1	-1
High_Cheekbones	1	1	-1	1	-1
Male	-1	1	1	-1	1
Mouth_Slightly_Open	-1	-1	-1	-1	-1
Mustache	-1	-1	-1	-1	-1
Narrow_Eyes	-1	-1	-1	-1	-1
No_Beard	-1	-1	-1	-1	-1
Oval_Face	-1	-1	-1	-1	-1
Pale_Skin	-1	-1	-1	-1	-1
Pointy_Nose	-1	-1	-1	-1	-1
Receding_Hairline	-1	-1	-1	-1	-1
Rosy_Cheeks	-1	-1	-1	-1	-1
Sideburns	-1	-1	-1	-1	-1
Smiling	1	1	1	1	1
Straight_Hair	-1	-1	-1	-1	-1
Wavy_Hair	-1	-1	-1	-1	-1
Wearing_Earrings	-1	-1	-1	-1	-1
Wearing_Hat	-1	-1	-1	-1	-1
Wearing_Lipstick	-1	-1	-1	1	-1
Wearing_Necklace	-1	-1	-1	-1	-1
Wearing_Necktie	-1	-1	-1	-1	-1
Young	1	1	-1	1	1

We then compute the average CLIP score (Radford et al., 2021) for each of a batch of 128, where the image is up-sampled to 224, and the following text descriptions are used:

- "a photo of a young woman with no hair"
- "a photo of a man wearing make-up"
- "a photo of an older man who is smiling"
- "a photo of a blonde woman with lipstick"
- "a photo of a man with black hair"

E.4 Faster Convergence with Distillation

Human face datasets are somewhat less diverse than AFHQv2 or CIFAR10, so we notice E-DSM does not struggle as much. Although E-DSM and our distilled approaches yield similar final FID scores for CelebA and FFHQ datasets, the time to train is a significant factor motivating our method.

Given the careful initialization and pretrained model we notice that our distilled training yields good performance with relatively few training steps.

With FFHQ-64, after $30,000$ training iterations, E-DSM from scratch yields FID scores of 7.69, DSM yields FID of 6.18 and our distilled model yields scores of 3.36. It takes approx. $200,000$ iterations for E-DSM to reach FID score below 3.3.

At $30,000$ iterations of training on the unconditional CelebA dataset; E-DSM achieves FID of 5.88, DSM of 5.18 and our distilled approach of 2.92. It takes a further approx. $100,000$ iterations for E-DSM and DSM to reach FID below 3.

F Licences

- JAX Apache-2.0 license Bradbury et al. (2018)
- CelebA: non-commercial research purposes (Liu et al., 2015)
- CIFAR-10: MIT license (Krizhevsky et al., 2009)
- FFHQ: Creative Commons BY-NC-SA 4.0 license (Karras et al., 2019)
- AFHQv2: Creative Commons BY-NC 4.0 license (Choi et al., 2020)
- Inception-v3 model: Apache V2.0 license (Szegedy et al., 2016)
- CLIP: MIT license (Radford et al., 2021)