MCMC-Correction of Score-Based Diffusion Models for Model Composition

Anonymous authors
Paper under double-blind review

Abstract

Diffusion models can be parameterized in terms of either a score or an energy function. The energy parameterization is attractive as it enables sampling procedures such as Markov Chain Monte Carlo (MCMC) that incorporates a Metropolis–Hastings (MH) correction step based on energy differences between proposed samples. Such corrections can significantly improve sampling quality, particularly in the context of model composition, where pre-trained models are combined to generate samples from novel distributions. Score-based diffusion models, on the other hand, are more widely adopted and come with a rich ecosystem of pre-trained models. However, they do not, in general, define an underlying energy function, making MH-based sampling inapplicable. In this work, we address this limitation by retaining the score parameterization and introducing a novel MH-like acceptance rule based on line integration of the score function. This allows the reuse of existing diffusion models while still combining the reverse process with various MCMC techniques, viewed as an instance of annealed MCMC. Through experiments on synthetic and real-world data, we show that our MH-like samplers offer comparable improvements to those obtained with energy-based models, without requiring explicit energy parameterization.

1 Introduction

Significant advancements have recently been achieved in generative modelling across various domains (Brock et al., 2019; Brown et al., 2020; Ho et al., 2020). These models have become potent priors for a wide range of applications, including code generation Li et al. (2022), text-to-image generation Saharia et al. (2022), question-answering Brown et al. (2020), and many others Güngör et al. (2023); Wynn & Turmukhambetov (2023). Among the generative models, diffusion models (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Ho et al., 2020) have arguably emerged as the most powerful class. Diffusion models learn to denoise corrupted inputs in small, gradual steps and are capable of generating samples from complex distributions. They have been successful in many domains, such as generating highly realistic images (Dhariwal & Nichol, 2021), modeling temporal point processes (Lüdke et al., 2023) and even generating neural network parameters (Wang et al., 2024).

Diffusion models also offer the capability of composed sampling, which combines pre-trained models to generate samples from a new distribution. This approach, known as model composition, has a rich history (Jacobs et al., 1991; Hinton, 2002; Mayraz & Hinton, 2000; Liu et al., 2022). For diffusion models, the most common form of composition is classifier-guided sampling, where the reverse process is augmented by a separate classifier model (Sohl-Dickstein et al., 2015; Dhariwal & Nichol, 2021; Ho & Salimans, 2021), but other compositions have also been explored (Du et al., 2023). The ability to compose new models without having to re-learn the individual components is especially appealing for diffusion models since their ever-increasing size and data hunger make them exceedingly costly to train (Aghajanyan et al., 2023). Therefore, developing sampling methods that work for pre-trained diffusion models is valuable.

The foundation of composed sampling for diffusion models is score-based, where we interpret diffusion models as predictors of the score function for the marginal distribution at each diffusion step (Song et al., 2021). From this perspective, MCMC methods, such as the Langevin algorithm (LA) (Roberts & Stramer, 2002) or

Hamiltonian Monte Carlo (HMC) sampling (Duane et al., 1987), emerge as viable options to incorporate. Augmenting the standard reverse process with additional MCMC sampling has been shown to improve composed sampling for diffusion models (Du et al., 2023; Song et al., 2021). However, we are restricted to unadjusted variants of these samplers, namely Unadjusted LA (U-LA) and Unadjusted HMC (U-HMC), which only require utilization of the score. This limitation means we cannot incorporate a Metropolis-Hastings (MH) correction step (Metropolis et al., 1953; Hastings, 1970), which requires evaluating the unnormalized density.

An intriguing alternative to directly modeling the score function is to model the marginal distribution with an energy function, from which the score can be obtained through explicit differentiation (Salimans & Ho, 2021; Song & Ermon, 2019). This parameterization connects diffusion models and energy-based models (EBMs) (LeCun et al., 2006) and offers several desirable properties. With an energy parameterization, we can evaluate the unnormalized density and guarantee a proper score function. This, in turn, enables an MH correction step when employing an MCMC-method, where the MH acceptance probability is computed from the energy function. Adding such a correction step has been shown to improve sampling performance in composed models (Du et al., 2023). Nevertheless, the score parameterization remains far more popular, as it avoids the direct computation of the gradient of the log density.

In this study, we build on the work in (Du et al., 2023) and introduce a novel approach to obtain an MH-like correction step directly from pre-trained diffusion models without relying on an energy-based parameterization. Specifically, we use a connection between the score and the energy to estimate the MH acceptance probability by approximating a line integral along the vector field generated by the score. This enables an improved sampling procedure for various pre-trained score-parameterized diffusion models. We find that our approximate method quantitatively results in improvements comparable to the energy parameterization without having to estimate the energy directly.

In summary, our main contributions are:

- We show that MH-like correction sampling can be directly applied to score-based models without requiring additional training.
- We introduce two efficient algorithms to approximate the energy difference used in MH and demonstrate that our pseudo-energy difference more accurately represents analytical energy differences than an explicitly trained energy model in a toy example while performing on par with the energy model on MNIST.
- We establish that the sampling accuracy improvements achieved with MCMC for energy-based models can also be attained for score-based models while offering superior runtime performance.

2 Background

2.1 Diffusion Models

We consider Gaussian diffusion models initially proposed by Sohl-Dickstein et al. (2015) and further improved by Song & Ermon (2019); Ho et al. (2020). Starting with a sample from the data distribution $x_0 \sim q(\cdot)$, we construct a Markov chain of latent variables $x_1, \ldots, x_T$ by iteratively introducing Gaussian noise to the sample $q(x_t|x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I\right)$, where $\beta_t \in [0, 1)$, $\forall t = 1, \ldots, T$ are known. For large enough T we have $q(x_T) \approx \mathcal{N}\left(x_T; 0, I\right)$.

A diffusion model learns to gradually denoise samples by modeling the distribution of the previous sample in the chain $p_\theta(x_{t-1} \mid x_t), t = 1, \ldots, T$. Approximate samples from the data distribution $q(x_0)$ are obtained by starting from $x_T \sim \mathcal{N}\left(0, I\right)$ and sequentially sampling less noisy versions of the sample until the noise is removed. This is called the *reverse process*.

The reverse distribution is typically modeled as $p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$, since the posterior $q(x_{t-1}|x_t)$ can be well-approximated by a Gaussian distribution when the noise magnitude β_t is sufficiently small. The mean is parameterized as $\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sigma_t}\epsilon_\theta(x_t, t)\right)$, where α_t and σ_t are positive and

defined by $\{\beta_t\}_{t=1}^T$ Ho et al. (2020). The noise prediction model $\epsilon_\theta(x_t, t)$, typically a neural network, is learned from data. We assume $\Sigma_\theta(x_t, t) = \beta_t I$ throughout unless otherwise stated.

2.2 Energy-based Models

Energy based-models (EBM) represent probability distributions with a scalar, non-negative energy function E_θ, by assigning low energy to regions of the input space where the probability is high and high energy to regions where the distribution has little or no support:

$$\begin{aligned} p_\theta(x_t, t) &= \frac{1}{Z_\theta(t)} \exp\left(-\frac{1}{\sigma_t} E_\theta(x_t, t)\right), \\ Z_\theta(t) &= \int \exp\left(-\frac{1}{\sigma_t} E_\theta(x_t, t)\right) \mathrm{d}x_t. \end{aligned} \tag{1}$$

Here, we define E_θ as a time-dependent function and deliberately choose not to absorb σ_t (introduced in the previous section) into E_θ, to maintain a more explicit connection to diffusion models, as clarified in the next section. This time dependency can be seen as a sequence of energy functions, one for each diffusion step t. The normalization constant Z_θ is typically intractable, prohibiting computing a normalized density. However, Z_θ does not depend on the input x_t, making the so-called *score function* easy to compute:

$$\nabla_x \log p_\theta(x_t, t) = -\frac{1}{\sigma_t} \nabla_x E_\theta(x_t, t), \tag{2}$$

even though the gradient of the energy function can be costly to compute in practice.

2.3 Energy and Score Parameterized Diffusion Models

A popular method for training EBMs is denoising score matching (DSM). In DSM, assuming the data is perturbed by Gaussian noise, the loss function becomes identical to the one used for diffusion models (up to a factor of σ_t^2) (Song et al., 2021). This is achieved by identifying the noise prediction model, $\epsilon_\theta(x_t, t)$, as an EBM:

$$\epsilon_\theta(x_t, t) = \nabla_x E_\theta(x_t, t), \tag{3}$$

i.e., under the additional constraint that $\epsilon_\theta(x_t, t)$ defines a proper score. Thus, an EBM and a plain diffusion model only differ in their parameterization. We refer to the first as using an *energy parameterization* via E_θ, while the second, since ϵ_θ describes a pseudo-score, is referred to as using a *score parameterization*.

Both parameterizations have their advantages and disadvantages. The energy parameterization can evaluate the density $p_\theta(x_t, t)$ up to a normalization $Z_\theta(t)$, which enables various MCMC methods. Furthermore, by making the score equal to the gradient of an actual scalar function, we ensure a proper score. On the other hand, to evaluate the score function, E_θ must be explicitly differentiated, which can be costly.

The score parameterization is more flexible as it predicts an arbitrary vector field. While there is some empirical evidence that this improves sampling performance in diffusion processes (Du et al., 2023), this difference may primarily stem from model architecture (Salimans & Ho, 2021). Nevertheless, the score parameterization's direct estimation of the score function makes it more efficient for reverse process sampling and remains the more widely adopted approach. In the next section, we describe how these parameterizations affect the design of MCMC samplers for diffusion models.

3 MCMC Sampling For Diffusion Models

MCMC sampling is a promising strategy for improving diffusion model sampling since it can be combined with the reverse process. Just like the reverse process, there are MCMC methods which base their kernels on the score function, such as the Unadjusted Langevin Algorithm (U-LA) and the Unadjusted Hamiltonian

Monte Carlo (U-HMC) (Roberts & Stramer, 2002; Duane et al., 1987; Neal et al., 1996). For example, with U-LA we use the kernel

$$k_t\left(x^{\tau+1}|x^{\tau}\right) = \mathcal{N}\left(x^{\tau+1}; x^{\tau} + \delta_t \nabla_x \log p_\theta(x^{\tau}, t), 2\delta_t I\right),$$

at diffusion step t, where $x^0 = x_t$, δ_t is the step size, and the chain is iterated for L_t steps.

These methods are called unadjusted since as L_t grows, these samplers will converge to the target distribution, but only for infinitesimal step sizes δ_t. By adding, for instance, a Metropolis–Hastings (MH) correction step, we can sample with larger step sizes and still converge to the target distribution (Metropolis et al., 1953; Hastings, 1970). With the correction, we sample a candidate $\hat{x} \sim k_t\left(\cdot \mid x^{\tau}\right)$ and accept it as the new iterate with probability

$$\alpha = \min\left(1, \frac{p_\theta(\hat{x}, t)}{p_\theta(x^{\tau}, t)} \frac{k_t\left(x^{\tau} \mid \hat{x}\right)}{k_t\left(\hat{x} \mid x^{\tau}\right)}\right). \tag{4}$$

That is, we set the new iterate $x^{\tau+1} = \hat{x}$ with probability α, otherwise $x^{\tau+1} = x^{\tau}$.

The model p_θ appears in the acceptance probability as a ratio, which means that a normalised density is not required to compute α, since the normalisation constant cancels out. When p_θ is parameterized as an EBM (see (1)), the probability ratio simplifies to

$$\frac{p_\theta(\hat{x}, t)}{p_\theta(x^{\tau}, t)} = \exp\left(\frac{1}{\sigma_t}\Big(E_\theta(x^{\tau}, t) - E_\theta(\hat{x}, t)\Big)\right). \tag{5}$$

This allows us to directly evaluate the MH acceptance probability, making it straightforward to construct an adjusted MCMC sampler. This offers a key advantage over the score parameterization, where only an approximation of the score is accessible which cannot directly be used to compute the probability ratio needed in MH.

3.1 Sampling from Composed Models

Composed sampling is a powerful feature of diffusion models that enables sampling from new target distributions by combining multiple pre-trained models. Rather than retraining a model for every new task or data combination, one can reuse existing components. This flexibility is especially appealing in large-scale settings, where retraining is often prohibitively expensive.

The most common form of composition is *guidance* (Dhariwal & Nichol, 2021), where the goal is to sample from a distribution conditioned on a class label y,

$$q(x_0 \mid y) \propto q(x_0) q(y \mid x_0). \tag{6}$$

This is implemented by modifying the score function at each diffusion step as

$$\nabla_x \log p_\theta(x_t, t) + \lambda \nabla_x \log p_\varphi(y \mid x_t, t), \tag{7}$$

where p_θ is an unconditional diffusion model and p_φ is a classifier predicting class y. A hyperparameter λ controls the strength of the conditioning. We refer to this approach as *classifier-full guidance*. Other variants include reconstruction guidance (Chung et al., 2023; Ho et al., 2022) and classifier-free guidance (Ho & Salimans, 2021).

More generally, Du et al. (2023) explore a range of composition types beyond guidance, including *products*, *negations*, and *mixtures*. A product distribution—of which guidance can be seen as a special case—is defined as

$$q^{\Pi}(x_0) \propto \prod_i q^i(x_0), \tag{8}$$

and leads to the composed model at diffusion step t,

$$p^{\Pi}_\theta(x_t, t) \propto \prod_i p^i_{\theta_i}(x_t, t) = \exp\left(-\frac{1}{\sigma_t} \sum_i E^i_{\theta_i}(x_t, t)\right). \tag{9}$$

This distribution is then used as the target in MCMC sampling, resulting in improved sampling performance.

Importantly, the factorization in (8) only strictly holds at $t = 0$; at intermediate diffusion steps, the composed model $p_\theta^\Pi(x_t, t)$ does not generally correspond to the true marginal of any product data distribution Du et al. (2023). This becomes problematic when relying solely on the reverse process, which assumes access to a valid score function for the true intermediate marginals. However, this construction remains valid and effective from the perspective of *annealed MCMC* Neal (2001), where the overall sampling procedure is interpreted as a chain targeting a sequence of gradually evolving distributions. From this viewpoint, the intermediate distributions $p_\theta^\Pi(x_t, t)$ are treated as design choices that guide the chain toward the final target $q^\Pi(x_0)$, and asymptotic correctness is still preserved. In practice, since diffusion models are trained using denoising score matching, the sampling process converges to a denoised version of $q^\Pi(x_0)$, which can be made arbitrarily close to the true distribution by construction.

Note for models using a score-based parameterization, a pseudo-score for this type of composition is equal to $-\frac{1}{\sigma_t}\sum_i \epsilon_{\theta_i}^i(x_t, t)$.

4 MCMC Correction Step For Score Parameterization

We propose combining the energy parameterization properties with the performance and practical accessibility of the score parameterization. Instead of using an energy parameterization and computing the score by differentiation, we take the complementary approach: using a score parameterization and computing the change in (pseudo-)energy by integrating the score.

4.1 Pseudo-energy Difference and MH-like Correction

This section describes how MCMC acceptance probabilities can be approximated given a score function. The MH acceptance probability in (4) is based on the relative probability of the new candidate $\hat{x}$ and the current sample x^τ. The transition probabilities given by the kernel $k_t(\cdot \mid \cdot)$ are assumed to be simple to compute, and we focus on the quotient $p_\theta(\hat{x}, t)/p_\theta(x^\tau, t)$. To compute the MH acceptance probability α, we only need to evaluate the unnormalized target distribution. For an EBM, this can be expressed in terms of the difference in energy at $\hat{x}$ and x^τ, see (5). That is, we do not need to compute the absolute value of the energy, only the difference.

To express the acceptance probability in terms of the score function of an EBM, we write the difference in energy as a line integral over a curve $\mathcal{C}$

$$E_\theta(x^\tau, t) - E_\theta(\hat{x}, t) = -\int_{\mathcal{C}} \nabla_r E_\theta(r, t) \cdot \mathrm{d}r = -\int_0^1 \nabla_r E_\theta(r(s), t) \cdot r'(s)\,\mathrm{d}s, \tag{10}$$

where $r(s)$ is a parameterization of $\mathcal{C}$ such that $r(0) = x^\tau$ and $r(1) = \hat{x}$. The choice of curve is arbitrary (under mild conditions), since E_θ is a scalar field.

For a score-parametrized diffusion model, we propose using a similar approach and calculating an MH-like ratio as follows:

$$\alpha = \min\left(1, \exp\left[\frac{1}{\sigma_t} f(\hat{x}, x^\tau, t)\right] \frac{k_t\left(x^\tau \mid \hat{x}\right)}{k_t\left(\hat{x} \mid x^\tau\right)}\right), \tag{11}$$

where

$$f(\hat{x}, x^\tau, t) = -\int_0^1 \epsilon_\theta(r(s), t) \cdot r'(s)\,\mathrm{d}s, \tag{12}$$

representing our constructed *pseudo-energy difference*. This expression can be seen as integrating the vector field ϵ_θ along a path from x^τ to $\hat{x}$, thereby approximating the change in a scalar potential—if such a potential existed. Note that if $\epsilon_\theta(x, t) = \nabla_x F(x, t)$ for some function F, (12) can be interpreted as recovering an (unknown) energy function, and in this case (11) agrees with (4). In general, however, no such function F

exists, and the expression (12) depends on the path r that is integrated over. Nevertheless, we propose using (11) to directly model an MH-like acceptance probability to be used in an MCMC sampling scheme.

Since (12) in general depends on the path r between x^τ and $\hat{x}$, we propose two variants for the curve $\mathcal{C}$. The first is a straight line connecting the two points. The second is a curve that passes through intermediate points where the score function $\epsilon_\theta(x, t)$ is already evaluated as part of the MCMC proposal step—for instance, the leapfrog trajectory in HMC. The motivation behind the second option is computational: since methods like HMC already require score evaluations at multiple points to propose x^τ, we can reuse these same evaluations to compute the pseudo-energy difference. By aligning the integration path with the proposal trajectory, we achieve higher numerical accuracy without incurring additional model evaluations.

Specifically, we approximate the line integral with the trapezoidal rule, where the number of line segments used to approximate the curve $\mathcal{C}$ is treated as a hyperparameter. Note that we have to evaluate ϵ_θ at some internal points on $\mathcal{C}$, incurring an additional computational burden (except for those we can re-use in the HMC case), but we avoid differentiating the model by estimating the score function directly, using ϵ_θ. Conversely, the energy parameterization only evaluates the energy at x^τ and $\hat{x}$, but has to differentiate E_θ to obtain the score.

An overview of the full sampling procedure is provided in Algorithm 1. At each diffusion step, an optional reverse update is followed by an MCMC refinement targeting the intermediate distribution. This formulation aligns with the annealed MCMC framework, where both the reverse step and the MCMC kernel act as design choices guiding the chain toward the final distribution. Including the reverse step typically improves sample quality (Du et al., 2023).

Algorithm 1 Annealed MCMC with MH-like correction for score-based diffusion models

Require: Score function $\epsilon_\theta(\cdot, t)$, schedule parameters β_t, α_t, σ_t, total steps T, MCMC steps L_t, kernel step size δ_t, integration segments n

$x_T \sim \mathcal{N}(0, I)$ ▷ Initialize from prior
for $t = T$ **to** 1 **do**
  $\epsilon \sim \mathcal{N}(0, I)$
  $x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sigma_t} \epsilon_\theta(x_t, t) \right) + \sqrt{\beta_t}\epsilon$ ▷ (Optional) reverse step
  if $t > 1$ **then**
    $x^0 = x_{t-1}$ ▷ Initialize MCMC chain
    for $\tau = 1$ **to** L_t **do**
      Propose candidate $x^\tau \sim k_{t-1}(\cdot \mid x^{\tau-1}; \epsilon_\theta, \delta_{t-1}, \sigma_{t-1})$ ▷ k_{t-1}: LA or HMC kernel
      Compute $f(x^\tau, x^{\tau-1}, t-1)$ via n-segment line integral
      Compute MH-like acceptance probability α using Eq. (11)
      Accept/reject: $x^\tau \leftarrow x^\tau$ with prob. α, else $x^\tau \leftarrow x^{\tau-1}$
    end for
    $x_{t-1} = x^L$ ▷ Use final sample from MCMC
  end if
end for
return x_0 ▷ Final denoised sample

4.2 MH-correction for Composition Models

The pseudo-energy difference for compositions can be derived based on their specific definitions. Our proposed method applies directly to product compositions. We calculate a pseudo-energy difference, corresponding to $E^\Pi_\theta(x^\tau, t) - E^\Pi_\theta(\hat{x}, t)$ for an EBM (defined in (9)), as

$$-\int_0^1 \sum_i \epsilon^i_{\theta_i}(r(s), t) \cdot r'(s), \mathrm{d}s. \tag{13}$$

Guidance is a specific case of product composition, where the pseudo-score is composed of two terms according to (7): the unconditional diffusion model $\epsilon_\theta(x_t, t)$ and the score of a classifier $p_\varphi(y \mid x_t, t)$. Since $p_\varphi(y \mid x_t, t)$

can be evaluated directly, only the pseudo-energy difference for $\epsilon_\theta(x_t, t)$ requires computation using the line integral in (13).

The pseudo-energy difference for a negation composition (as defined in (Du et al., 2023)) can be computed analogously to products, as negations follow a similar additive structure in their pseudo-scores.

Mixture compositions (as defined in (Du et al., 2023)), on the other hand, cannot be expressed as a pseudo-energy difference, since mixtures do not naturally conform to an additive structure analogous to products or negations. However, mixtures can be addressed by first sampling a component distribution according to the mixture definition and then generating a sample from that distribution. The MH-correction can subsequently be applied to this sampled distribution, providing a seamless way to handle mixture compositions within our framework.

This generalization allows our method to support advanced use cases such as classifier guidance, multi-modal fusion, and spatially structured prompts, without requiring retraining or access to energy-based models.

5 Results

In this section, we present an empirical evaluation of our MH-like correction method, examining both the accuracy of the pseudo-energy differences and the quality of the generated samples. The experiments are designed to span a spectrum of difficulty: from controlled, low-dimensional setups where models can be trained from scratch and analytical solutions are available, to more realistic high-dimensional scenarios involving pre-trained models. Our two primary objectives are (1) to compare our proposed approach against a true energy parameterization when available, and (2) to assess the sampling improvements achieved over the standard reverse process when augmented with MCMC steps.

The experiments in Sections 5.1, 5.2, and the first part of 5.3 involve training diffusion models using both energy and score parameterizations. The score parameterization follows a noise prediction model, $\epsilon_\theta(x_t, t)$, while the energy parameterization defines an energy function as $E_\theta(x_t, t) = \|x_t - s_\theta(x_t, t)\|_2^2$, as in (Du et al., 2023). We use identical network architectures for ϵ_θ and s_θ. Both models are trained with the standard diffusion loss (Ho et al., 2020), with the energy model's score function obtained through explicit differentiation.

The later experiments utilize only pre-trained score-based diffusion models, as pre-trained energy-based models are unavailable for direct comparison. We evaluate both unadjusted and MH-corrected versions of Langevin and Hamiltonian Monte Carlo, comparing them against the standard reverse process, which serves as the baseline.

For the MH-like correction, we examine two types of integration paths: line and curve. The line follows a direct path between x^τ and $\hat{x}$, while the curve integrates along the trajectory formed by HMC leapfrog steps. The number of points for the trapezoidal rule's mesh is treated as a hyperparameter. Since points like x^τ and $\hat{x}$ are already included, the hyperparameter refers to the additional points, which are evenly distributed along the curve.

Complete training details, hyperparameter settings, and implementation specifics are deferred to Appendix A.

5.1 Evaluating Pseudo-Energy Differences

To evaluate the accuracy of pseudo-energy differences, we conducted experiments on a synthetic 2D dataset, generated from a bivariate Gaussian distribution to allow access to analytical solutions, and a higher-dimensional dataset, MNIST (Deng, 2012). For each experiment, we trained 10 score and energy models independently from scratch. For evaluation, we sampled 2k pairs of points (x_t^1, x_t^2) via the forward process at various diffusion steps t, and these pairs were used to compute the (pseudo-)energy difference ΔE for both the score and energy models (and analytically when available). The pseudo-energy difference was computed along a linear curve connecting the two points, using five points for numerical integration.

2D Gaussian: For the 2D Gaussian dataset, the relative error metric is defined as $|\Delta E_{\text{pred}} - \Delta E_{\text{true}}|/|\Delta E_{\text{true}}|$, where ΔE_{pred} is the predicted energy difference and ΔE_{true} is the analytical energy difference. The median relative error was calculated across all sampled pairs for each trained model, and the mean and standard

deviation of this metric were computed across the 10 models. Interestingly, the score model achieved a lower relative error 0.071 ± 0.005 compared to the energy model 0.084 ± 0.004, demonstrating better alignment with the true energy differences.

MNIST: For the MNIST dataset, where analytical energy differences are unavailable, we used a symmetric relative error metric defined as $2|\Delta E_{\text{score}} - \Delta E_{\text{energy}}|/(|\Delta E_{\text{score}}| + |\Delta E_{\text{energy}}|)$. The median relative error was calculated across all sampled pairs for each trained model, and the mean and standard deviation were computed across the 10 models. This yielded a mean relative error of 0.030 ± 0.002 indicating that the energy differences predicted by the score and energy models align closely, even in this higher-dimensional setting.

5.2 2D Composition

To investigate the effectiveness of our MH-like correction in a controlled yet expressive setting, we replicate the 2D composition experiment introduced by Du et al. (2023), using their publicly available codebase[1] as a foundation. Our experimental setup mirrors theirs unless otherwise specified.

A 2D density pair is composed via multiplication into a complex distribution, as in (9): a Gaussian mixture with 8 modes in a circle and a uniform distribution covering two of the modes. For a visual representation of the two individual distributions and their resulting product distribution together with samples from the reverse diffusion and HMC corrected samples, see Figure 1. The baseline reverse diffusion process uses $T = 100$ steps. In the MCMC variants, following Du et al. (2023), we omit the optional reverse step for a fair comparison. MCMC sampling runs for $L_t = 10$ at each t, with (U-)HMC using 3 leapfrog steps per MCMC step. We evaluate performance using three metrics. The first is negative log-likelihood (NLL), which

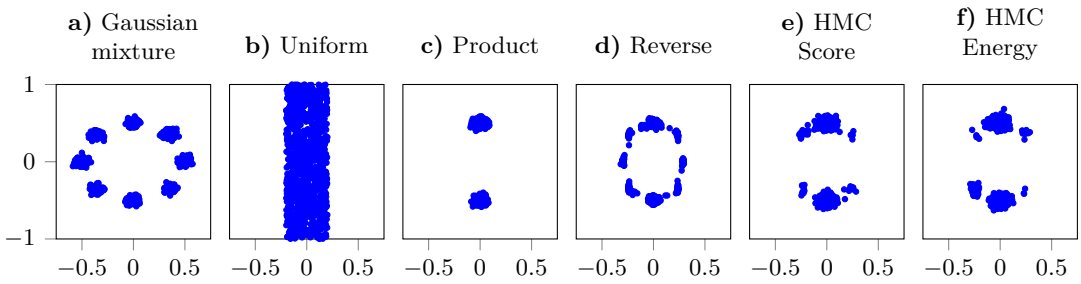


Figure 1: Samples from: **a-b):** the component distributions: a Gaussian mixture and a uniform distribution, **c):** the true product distribution, **d):** a standard score parameterized reverse process, **e-f):** HMC sampling using score and energy parameterization respectively.

assesses the likelihood of generated samples under the true data distribution. To address potential samples outside the true distribution's support, we extend it by adding a small uniform probability. The second metric is a Gaussian mixture model (GMM), where we fit bi-modal GMMs to samples from both the true and model distributions and compute the Frobenius norm of the variance mean difference. Finally, we use the Wasserstein-2 distance (W_2) to measure the discrepancy between the data and model distributions by computing the optimal assignment between sampled sets (Villani, 2009).

We present quantitative results for the 2D composition in Table 1(a), averaged over 10 independent trials. In each trial, we train the diffusion models from scratch and sample 2000 points using different MCMC methods. The results show that the corrected sampling methods outperform the unadjusted ones. HMC variants yield better results than Langevin, while the reverse process performs worse. Score and energy parameterizations exhibit similar NLL and GMM performance within their respective methods. However, with HMC, the score parameterization significantly outperforms the energy parameterization in W_2. Performance also saturates with as few as three points in the trapezoidal rule.

Additionally, we measured memory usage and runtime during this experiment, see Table 1(b). Score-based parameterization was more than twice as memory-efficient as energy-based parameterization and, with the

[1] `https://github.com/yilundu/reduce_reuse_recycle`

exception of LA with 8 extra trapezoidal evaluations, faster for the corresponding MCMC methods. Notably, HMC curve was nearly three times faster. While our approach requires more model evaluations, this does not necessarily make it slower or more memory-intensive than using an energy-based model. However, these results are implementation-dependent, and further investigation is needed to confirm whether these trends generalize to other setups.

Table 1: Quantitative results for different samplers in the 2D composition experiment. (a) shows performance metrics (NLL, GMM, and W_2) based on 10 independent trials, with lower values indicating better performance. (b) reports average runtime (in seconds) and peak memory consumption (in MiB). For the score parameterization, we include variants with different numbers of additional points in the trapezoidal rule (e.g., 1L, 3L, 8L) and different integration paths ("L" for a straight line and "C" for the HMC trajectory).

(a) Performance metrics

	Sampler	NLL↓	GMM↓	W_2↓
Energy	Reverse	8.22 ± 0.21	27.01 ± 1.34	5.81 ± 0.19
	U-LA	7.52 ± 0.22	14.61 ± 1.35	4.19 ± 0.45
	LA	6.50 ± 0.30	14.66 ± 1.46	4.24 ± 0.55
	U-HMC	5.72 ± 0.18	6.53 ± 0.91	4.19 ± 1.25
	HMC	$\mathbf{4.09 \pm 0.14}$	$\mathbf{3.33 \pm 0.65}$	$\mathbf{4.12 \pm 1.44}$
Score	Reverse	8.15 ± 0.24	26.88 ± 1.20	5.80 ± 0.20
	U-LA	7.57 ± 0.12	14.99 ± 0.62	4.44 ± 0.63
	LA-1L	6.45 ± 0.20	14.28 ± 1.07	4.03 ± 0.52
	LA-3L	6.61 ± 0.17	15.19 ± 0.92	4.22 ± 0.46
	LA-8L	6.53 ± 0.17	14.75 ± 0.91	4.20 ± 0.51
	U-HMC	5.77 ± 0.12	6.90 ± 0.71	3.39 ± 0.77
	HMC-1L	4.29 ± 0.13	3.72 ± 0.61	2.92 ± 1.02
	HMC-3L	$\mathbf{4.07 \pm 0.13}$	3.08 ± 0.69	$\mathbf{2.68 \pm 1.20}$
	HMC-8L	$\mathbf{4.07 \pm 0.14}$	3.17 ± 0.56	2.87 ± 0.89
	HMC-C	$\mathbf{4.07 \pm 0.12}$	$\mathbf{3.06 \pm 0.54}$	2.94 ± 0.90

(b) Runtime and memory usage

	Sampler	Time	Memory
Energy	Reverse	2.1	5252
	U-LA	2.7	5252
	LA	10.4	5252
	U-HMC	19.3	5254
	HMC	22.8	5256
Score	Reverse	1.6	2178
	U-LA	2.2	2180
	LA-1L	6.1	2180
	LA-3L	8.8	2180
	LA-8L	13.6	2180
	U-HMC	9.5	2180
	HMC-1L	8.8	2180
	HMC-3L	11.6	2180
	HMC-8L	16.4	2180
	HMC-C	7.1	2180

5.3 Guided Diffusion

We evaluate our proposed sampling methods for guided diffusion on the CIFAR-100 (Krizhevsky & Hinton, 2009) and ImageNet (Deng et al., 2009) datasets. The sampling process is based on a score function defined in (7). For both datasets, the marginal score, $\nabla_x \log q(x_t)$, is estimated using an unconditional diffusion model parameterized by a UNet architecture. For the guidance model, we use classifier-full guidance, training a time-dependent classifier to predict class labels across all diffusion steps, $p_\varphi(y \mid x_t, t)$. This classifier shares its architecture with the encoder part of the UNet used for the diffusion model and is extended with a dense output layer. The guidance scale is set to $\lambda = 20.0$ across all experiments. Sampling is based on the standard reverse process with $T = 1000$, and additional MCMC steps are incorporated to refine the generated samples.

To quantify generation quality, we use three evaluation metrics: the Fréchet Inception Distance (FID) (Heusel et al., 2017), which compares the distribution of generated and real images; classification accuracy, based on a separate pre-trained classifier applied to generated samples; and, for ImageNet, an additional top-5 accuracy metric.

CIFAR-100: For CIFAR-100, we trained the diffusion models from scratch using the same UNet architecture and training settings as in Ho et al. (2020), which were originally designed for CIFAR-10 (Krizhevsky & Hinton, 2009). The MCMC samplers add $L_t = 2$ or 6 extra MCMC steps at each diffusion step t for (U-)HMC and (U-)LA, respectively, with (U-)HMC using three leapfrog steps per MCMC step.

For this experiment, more points are needed in the trapezoidal rule's mesh than in the 2D experiment. Based on previous insights, for HMC we integrate only along the curve from the leapfrog steps, with an additional

Table 2: Accuracy and FID score for classifier-full guidance on CIFAR-100. The metrics are based on 50k generated samples for each sampling method with both energy and score models.

	Sampler	Accuracy [%]↑	FID↓
Energy	Reverse	72.6	33.4
	U-LA	**87.3**	24.6
	LA	80.0	12.7
	U-HMC	87.2	25.4
	HMC	84.9	**12.4**
Score	Reverse	74.2	31.8
	U-LA	**82.9**	25.9
	LA-8L	75.2	15.5
	U-HMC	79.0	28.6
	HMC-3C	75.8	**13.3**

midpoint evaluation, resulting in three extra model evaluations per HMC step. For LA, we use ten points along the line, resulting in eight extra evaluations per step.

Recognizing the impact of the step length on MCMC methods in general, we parameterize the step length as a function of the beta-schedule $\delta_t = a\beta_t^b$. We conducted a simple parameter search for parameters a and b, to determine a suitable step length for each MCMC variant.

The results are shown in Table 2. Average accuracy is obtained using a separate classifier trained exclusively on noise-free pairs (x_0, y), following the VGG-13-BN architecture (Simonyan & Zisserman, 2014). The table shows a general trend of improvement over the baseline reverse process when additional MCMC steps are added. In particular, the MH-corrected samplers LA and HMC show significant improvements in FID scores, which are arguably the more important metric for image generation.

Comparing the score and energy parameterizations, their performances share similar characteristics. Interestingly, the reverse process favors the score parameterization, supporting the claim that this less restricted approach better models the score function. However, the energy parameterization sees larger improvements from the added MCMC steps. This indicates, perhaps, that direct energy estimation provides a better correction step compared to our method of approximating the pseudo-energy difference from ϵ_θ. Although the energy-based method performs slightly better in this setting, our MH-corrected sampling methods achieve comparable improvements without requiring an energy model.

ImageNet: For ImageNet, training diffusion models from scratch is computationally expensive, so we rely on pre-trained models. Score-based models are publicly available through the OpenAI GitHub repository[2], as provided by Dhariwal & Nichol (2021). Unfortunately, no equivalent pre-trained energy-based models are available. Given the high computational demands of large-scale diffusion models, we focus solely on evaluating HMC and compare it to the reverse process. The HMC sampler adds $L_t = 2$ MCMC steps per diffusion step t, with each step consisting of three leapfrog steps. For the trapezoidal rule, we incorporate the points from the leapfrog steps and add two additional points between each leapfrog step. The step length parameterization and tuning follow the same procedure as in CIFAR-100.

The results can be seen in Table 3. Accuracy metrics are computed using a pre-trained RegNetX-8.0GF Radosavovic et al. (2020) classifier. The reverse process and HMC perform very similarly in average accuracy, but our method shows a slight improvement in top-5 average accuracy. HMC obtains a significantly better FID score.

5.4 Image Tapestry

As our final experiment, we conduct an image tapestry experiment, similar to Du et al. (2023) and based on their code[3]. The goal is to generate a coherent image composed of spatially localized content, each region conditioned

[2] https://github.com/openai/guided-diffusion

[3] https://github.com/yilundu/reduce_reuse_recycle

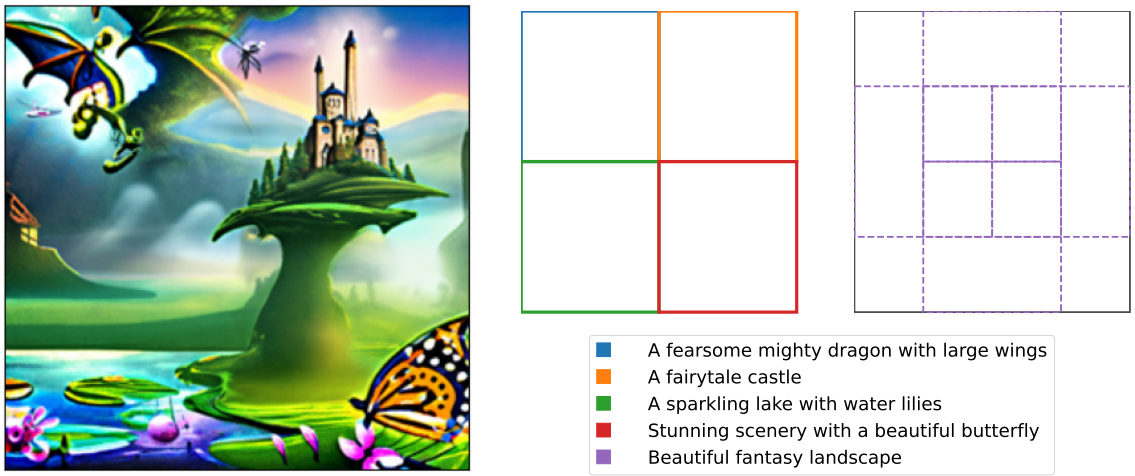
Table 3: Average accuracy, top-5 accuracy, and FID score for classifier-full guidance on ImageNet. The metrics are based on 50k generated samples for both sampling methods with score parameterizations.

	Sampler	Acc [%]↑	Acc-5 [%]↑	FID↓
Score	Reverse	**50.0**	83.9	14.5
	HMC-6C	49.9	**85.1**	**11.6**

on different prompts. This task involves both classifier-free guidance and model composition—specifically, the combination of multiple overlapping text-to-image diffusion models, each responsible for a portion of the scene.

We use a pre-trained DeepFloyd-IF model[4] as the base diffusion model. To refine the generated samples, we apply Langevin dynamics with our MH-like correction. For each diffusion step ($T = 100$), we include 15 additional Langevin steps. The pseudo-energy difference is approximated via line integration using three additional evaluation points per step. We set the classifier-free guidance scale to $\lambda = 20.0$.

The resulting image is presented in Figure 2(a), which showcases the generated tapestry with different regions displaying distinct visual content. Figure 2(b) provides a schematic overview of the used prompts and their spatial layout. In total, nine content regions are specified: four located in the corners of the image, each with unique prompts, and five overlapping in the center, all guided by the same prompt to create a unified visual theme.



(a) (b)

Figure 2: In (a), the generated tapestry image is shown with different content at various locations. In (b), the specified content and their positions are illustrated.

6 Discussion

The choice between score and energy parameterizations remains an intriguing and nuanced topic within diffusion-based generative modeling. In this work, we have provided additional empirical evidence suggesting that the score parameterization performs better in the standard reverse process.

At the same time, we have shown that performance gains often attributed to the energy parameterization can, in fact, be recovered within a score-based framework. This is achieved by approximating pseudo-energy differences using a line integral of the model's noise predictions. Notably, this allows us to incorporate MH-like

[4] `https://huggingface.co/DeepFloyd/IF-I-XL-v1.0`

correction steps into a variety of MCMC samplers—without the need to explicitly train an energy-based model—yet still attain comparable improvements in sample quality.

A particularly interesting observation is that using a curve composed only of model evaluations from the HMC sampler appears to perform on par with using a straight-line path. This suggests that the proposed correction comes at virtually no additional computational cost in this case. However, it is worth noting that in higher-dimensional settings, additional intermediate points along the integration path may be required to maintain accuracy, which could increase the computational burden. This challenge might be addressed through more efficient numerical integration techniques, or by working in a lower-dimensional latent space, as is done in latent diffusion models. One persistent drawback of the energy parameterization is that it always requires an explicit gradient computation to recover the score function.

One limitation of the score parameterization is that the learned vector field is not guaranteed to be conservative. In other words, it does not, in general, correspond to the gradient of a scalar energy function. Nonetheless, recent work by Horvat & Pfister (2024) demonstrates that a vector field does not necessarily need to be strictly conservative to generate accurate samples or estimate a density effectively. This perspective aligns with the empirical success of score-based generative models that operate without explicitly modeling an energy function. Likewise, our MH-like correction mechanism, though built on a non-conservative field, yields significant improvements when applied to the reverse process.

Still, the lack of exact conservativity may explain the slightly superior performance of the energy parameterization observed in the CIFAR-100 experiment. Developing better techniques for estimating pseudo-energy differences from score-based models—without requiring an explicitly trained energy function—thus remains a highly relevant and promising direction for future research.

7 Conclusion

We have introduced a method for extending the reverse diffusion process with MCMC sampling based on an MH-like correction step computed from the score function. This approach enables improved sampling for composed diffusion models without requiring an energy-based parameterization.

While previous work Du et al. (2023) demonstrated the benefits of MH correction under an energy parameterization, our method instead defines a pseudo-energy difference derived from the score, estimated via numerical integration. This allows us to apply MH-like corrections in the score-based setting—by far the most common in practice—and thereby make use of existing pre-trained diffusion models for composition tasks.

Our method can reuse intermediate evaluations from samplers such as HMC to compute the correction with little to no additional cost. In general, the accuracy of the MH-like correction depends on the numerical integration of the score, which may require more intermediate points as the dimensionality increases. While this can introduce some overhead, energy-based methods incur their own costs, such as differentiating the energy function. In practice, our corrected score-based samplers consistently match the performance of energy-based methods across a range of tasks, making them a practical alternative in settings where score-based models are already available.

Overall, our work extends the applicability of corrected MCMC sampling to the broad class of score-based diffusion models and opens the door to more flexible and modular composition of generative models.

References

Armen Aghajanyan, Lili Yu, Alexis Conneau, Wei-Ning Hsu, Karen Hambardzumyan, Susan Zhang, Stephen Roller, Naman Goyal, Omer Levy, and Luke Zettlemoyer. Scaling laws for generative mixed-modal language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR, 2023.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

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

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. IEEE, 2009.

Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. doi: 10.1109/MSP.2012.2211477.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Yilun Du, Conor Durkan, Robin Strudel, Joshua B. Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, Reuse, Recycle: Compositional generation with energy-based diffusion models and MCMC. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 8489–8510. PMLR, 23–29 Jul 2023.

Simon Duane, A.D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, 1987. ISSN 0370-2693.

Alper Güngör, Salman UH Dar, Şaban Öztürk, Yilmaz Korkmaz, Hasan A Bedel, Gokberk Elmas, Muzaffer Ozbey, and Tolga Çukur. Adaptive diffusion priors for accelerated mri reconstruction. *Medical Image Analysis*, pp. 102872, 2023.

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57 (1):97–109, 1970.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 8633–8646. Curran Associates, Inc., 2022.

Christian Horvat and Jean-Pascal Pfister. On gauge freedom, conservativity and intrinsic dimensionality estimation in diffusion models. *arXiv preprint arXiv:2402.03845*, 2024.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009. URL `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097, 2022.

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

David Lüdke, Marin Biloš, Oleksandr Shchur, Marten Lienen, and Stephan Günnemann. Add and thin: Diffusion for temporal point processes. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Guy Mayraz and Geoffrey E Hinton. Recognizing hand-written digits using hierarchical products of experts. *Advances in neural information processing systems*, 13, 2000.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, and Augusta H. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

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

Radford M. Neal, P. Diggle, and S. Fienberg. *Bayesian Learning for Neural Networks.* Lecture Notes in Statistics Ser.: v.118. Springer New York, 1996. ISBN 9781461207450.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10428–10436, 2020.

G. O. Roberts and O. Stramer. Langevin diffusions and Metropolis–Hastings algorithms. *Methodology & Computing in Applied Probability*, 4(4):337 – 357, 2002. ISSN 13875841.

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

Tim Salimans and Jonathan 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`.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

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

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

Cédric Villani. *The Wasserstein distances*, pp. 93–111. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-540-71050-9. URL https://doi.org/10.1007/978-3-540-71050-9_6.

Kai Wang, Zhaopan Xu, Yukun Zhou, Zelin Zang, Trevor Darrell, Zhuang Liu, and Yang You. Neural network diffusion, 2024.

Jamie Wynn and Daniyar Turmukhambetov. DiffusioNeRF: Regularizing neural radiance fields with denoising diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4180–4189, 2023.

A Experimental details

Here we provide more details about our different conducted experiments: Evaluating pseudo-energy differences, 2D composition, guided diffusion, and image tapestry.

The earlier experiments were conducted on a machine equipped with an NVIDIA GeForce RTX 3060, while the later experiments were run on a computing cluster with NVIDIA A100 Tensor Core GPUs.

A.1 Evaluating Pseudo-Energy Difference

2D Gaussian: We generated samples from a bivariate Gaussian distribution with mean $\boldsymbol{\mu} = (2, 0)^\top$ and covariance $\boldsymbol{\Sigma} = 0.1I$, where I is the identity matrix.

The diffusion models use $T = 100$ timesteps, with the noise schedule β_t following the cosine schedule proposed in Nichol & Dhariwal (2021).

We use the same neural network architectures as the base for both the score and energy models. It is a residual network consisting of a linear layer (dim $2 \to 128$) followed by four blocks, and concluding with a linear layer (dim $128 \to 2$). Within each block, the input x passes through a normalization layer, a SiLU activation, and a linear layer (dim $128 \to 256$). Subsequently, it is added with an embedded t (dim 32) that has undergone a linear layer transformation (dim $32 \to 256$). The resulting sum passes through a SiLU activation and is further processed by a linear layer (dim $256 \to 256$). After that, another SiLU activation is applied, followed by a final linear layer (dim $256 \to 128$). The output of this linear layer is then added to the original input x within the block. The embedding of t is also learnable.

MNIST: The diffusion models use $T = 1000$ timesteps, with the noise schedule β_t following the cosine schedule.

A.2 2D composition

The composed distribution is defined by a product of two components, a Gaussian mixture and a uniform distribution with non-zero values on

$$\square = \{x \in \mathbb{R}^2 : -s_i \leq x_i \leq s_i, i = 1, 2\}, \tag{14}$$

where s_1 and s_2 are equal to 0.2 and 1.0, respectively. The eight modes of the Gaussian mixture are evenly distributed on a circle with a radius of 0.5 at the angles $\frac{\pi}{4}i$ for $i = 0, \ldots, 7$, respectively. The covariance matrix at each mode is $0.03^2 \cdot I$, where I is the identity matrix.

We use the same network architecture setup for score and energy as in the 2D Gaussian case (see Section A.1).

The metric log-likelihood is ill-defined as we may generate samples where the true distribution has no support (due to the uniform distribution). We address this problem by expanding the definition set of the uniform distribution and redistributing one percent of the probability mass into this extended region. The whole set is defined as (14) except $s_1 = s_2 = 1.1$. Note that 99 percent probability mass remains inside the original definition set $\square$.

The parameter β_t follows the cosine schedule. For (U-)HMC, the damping coefficient is set to 0.5, the mass diagonal matrix has all diagonal elements equal to 1, and the stepsize for each t is 0.03. For (U-)LA, the stepsize for each t is set to 0.001.

A.3 Guided diffusion for CIFAR-100

The parameter β_t has a linear schedule as originally proposed in Ho et al. (2020). For (U)-HMC is the damping coefficient equal to 0.9 and the diagonal elements in the mass matrix are equal to β_t for each t. The values of the stepsize parameters a and b were determined through a simple parameter search for the different MCMC methods and they can be found in Table 4. This was done for both the score and energy parameterizations, where the stepsize is defined as $\delta_t = a\beta_t^b$.

Table 4: The values of the stepsize parameters a and b obtained from a random parameter search for the different MCMC methods for both score and energy parameterization in the CIFAR-100 experiment, where the stepsize is defined as $\delta_t = a\beta_t^b$.

	MCMC	Stepsize Parameters	
		a	b
Energy	U-LA	9.22	1.40
	LA	9.84	0.83
	U-HMC	0.26	1.53
	HMC	9.33	1.48
Score	U-LA	1.96	1.04
	LA	9.84	0.83
	U-HMC	0.26	1.53
	HMC	4.03	1.34

A.4 Guided diffusion for ImageNet

Again, the parameter β_t follows a linear schedule. The hyperparameters for the HMC include a damping coefficient set to 0.9, with the diagonal elements of the mass matrix being equal to β_t for each t. The stepsize parameters for HMC, obtained from a simple parameter search, are $a = 1.87$ and $b = 1.51$.

The ImageNet dataset[5] used to compute the FID score is available for free to researchers for non-commercial use.

A.5 Image tapestry

A cosine schedule is used for the parameter β_t. The stepsize parameters in this case is simply $a = 1$ and $b = 1$, i.e., $\delta_t = \beta_t$.

[5]https://image-net.org/