RETHINKING THE TRAINING OF DIFFUSION BRIDGE SAMPLERS: LOSSES AND EXPLORATION

Sebastian Sanokowski[1] [*], **Christoph Bartmann** [1], **Lukas Gruber** [1],
Sepp Hochreiter [1,2], **Sebastian Lehner** [1]
[1] ELLIS Unit Linz, LIT AI Lab, Johannes Kepler University Linz, Austria
[2] NXAI Lab & NXAI GmbH, Linz, Austria

ABSTRACT

Diffusion bridges are a promising class of deep-learning methods for sampling from unnormalized distributions. Recent works show that the Log Variance (LV) loss consistently outperforms the reverse Kullback-Leibler (rKL) loss when using the reparametrization trick to compute rKL-gradients. While the LV loss is theoretically justified for diffusion samplers with non-learnable forward processes—yielding identical gradients to the rKL loss combined with the log derivative trick—this equivalence does not hold for diffusion bridges. We point out that the LV loss does not unconditionally satisfy the data processing inequality, casting doubt on its suitability for diffusion bridges. To avoid this problem we employ the rKL loss with the log derivative trick and show that it consistently outperforms the LV loss. Furthermore, we introduce two techniques for controlling the exploration-exploitation trade-off in diffusion samplers—one based on variational annealing and the other on off-policy exploration. We validate their effectiveness on highly multimodal benchmark tasks.

1 INTRODUCTION

We consider the task of learning to generate samples $X \in \mathbb{R}^N$ from a target distribution

$$\pi_0(X, \beta) = \frac{\exp\left(-\beta\, \mathcal{E}(X)\right)}{\mathcal{Z}} \text{ where } \quad \mathcal{Z} = \int_{\mathbb{R}^{\mathbb{N}}} \exp\left(-\beta\, \mathcal{E}(X)\right) dX, \tag{1}$$

where $\mathcal{Z}$ represents the partition function, $\mathcal{E} : \mathbb{R}^N \to \mathbb{R}$ is the energy function, and β is the inverse temperature of the target distribution π_0 which is usually set to 1. In this setting, it is assumed that the energy function $\mathcal{E}$ of the target distribution can be evaluated while $\mathcal{Z}$ is unknown and computationally intractable. Sampling problems of this kind represent fundamental challenges in computational physics and chemistry and in Bayesian learning (Wu et al., 2019; Noé & Wu, 2018; Shih & Ermon, 2020). Recent approaches have focused on training generative neural networks to approximate target distributions. Early deep learning-based methods explored exact likelihood models such as normalizing flows Noé & Wu (2018) and autoregressive models Wu et al. (2019), while more recent work has turned to approximate likelihood models like diffusion samplers in continuous Zhang & Chen (2022) and discrete domains Sanokowski et al. (2024). However, diffusion samplers based on Stochastic Differential Equations (SDEs) face significant challenges in terms of practical applicability. These models require extensive tuning of hyperparameters, particularly diffusion coefficients, which become infeasible when problem scales differ across dimensions.

In this work, we make four key contributions to address these limitations of diffusion bridge-based samplers. **(i)** In addition to the usual learned drift terms of SDEs we propose learnable diffusion terms that enable dynamic adaptation of the exploration-exploitation trade-off, significantly reducing the need for manual hyperparameter tuning. Our approach yields significantly improved performance when applied to hitherto state-of-the-art samplers that build on diffusion bridges. **(ii)** We identify crucial problems in the application of the popular Log Variance (LV) (Nüsken & Richter, 2021; Richter et al., 2023) loss in these diffusion bridges. While the LV loss and the reverse

[*] Correspondance to sanokowski[at]ml.jku.at

Kullback-Leibler (rKL) are equivalent when only the reverse diffusion process is learned, we identify a critical discrepancy that arises in the application of these losses in the context of diffusion bridges and when learning SDE parameters that are shared in the forward- and reverse- diffusion process. We demonstrate that this discrepancy can drastically deteriorate model performance and violates a crucial feasibility criterion for divergences in diffusion samplers. **(iii)** By introducing a simple rKL-based loss that mitigates these limitations, we consistently outperform recent literature baselines. **(iv)** Based on this rKL-based loss we propose the usage of simple exploration methods that effectively encourage exploration in multimodal distributions and prevent mode collapse.

2 Problem Description

2.1 Diffusion Bridges

Diffusion models are generative models that are trained to map samples X_T from a simple prior distribution π_T to samples from a target distribution $X_0 \sim \pi_0$. The diffusion path that defines how samples are supposed to be transported from X_T to X_0 is determined by a so-called forward diffusion process, which is either defined by a fixed forward SDE or as in the case of diffusion bridges by a learnable forward SDE. For predefined forward stochastic differential equations (SDEs), such as variance-exploding or variance-preserving SDEs, the selection of parameters—particularly the diffusion coefficient and drift term—must be carefully calibrated to align with the target distribution. This enables the forward process to map samples to the desired prior distribution. In diffusion bridges, this problem is mitigated as the parameterized forward diffusion process can in principle learn to map to any prior distribution. Due to this flexibility diffusion bridges (De Bortoli et al., 2021; Richter et al., 2023; Vargas et al., 2024) have recently attracted increased research interest and represent the state-of-the-art in a wide range of popular sampling benchmarks. Our contributions build upon these methods and show how they can be significantly improved.

We define the forward diffusion process via the following SDE:

$$dX_t = v(X_t, t)\, dt + \sigma_t\, dW_t \quad \text{where} \quad X_0 \sim \pi_0, \tag{2}$$

where W is a Brownian motion, i.e. $dW_t = \epsilon_t \sqrt{dt}$ and $\epsilon_t \sim \mathcal{N}(\epsilon_t; 0, \mathrm{I})$. The reverse process is defined as:

$$dX_\tau = u(X_\tau, \tau)\, d\tau + \sigma_\tau\, dW_\tau \quad \text{where} \quad X_T \sim \pi_T \tag{3}$$

where time evolves in the opposite direction: $t = T - \tau$ and $u(X_\tau, \tau), X_t, dW_t$ and σ_t are each in $\mathbb{R}^N$. We follow Vargas et al. (2024) and aim to learn an optimal transport along a path π_t that is defined via an interpolation between the target and prior distributions according to $\pi_t = \pi_0^{\eta(t)}\, \pi_T^{1-\eta(t)}$ where $\eta(t) \in [0,1]$ is a monotonically increasing function with $\eta(0) = 1$ and $\eta(T) = 1$.

They propose to parameterize the forward process according to:

$$dX_t = \left(\frac{\sigma_t^2}{2} \nabla_{X_t} \log \pi_t(X_t) - u_\theta(X_t, t) \right) dt + \sigma_t\, dW_t,$$

where $X_0 \sim \pi_0$ and the reverse process according to:

$$dX_\tau = \left(\frac{\sigma_\tau^2}{2} \nabla_{X_\tau} \log \pi_\tau(X_\tau) + u_\theta(X_\tau, \tau) \right) d\tau + \sigma_\tau\, dW_\tau,$$

where $X_T \sim \pi_T$. The drift $u_\theta(X_t, t) = \sigma_t^2 s_\theta(X_t)$ where $s_\theta(X_t)$ is the control which is parameterized by a neural network. In practice the reverse process is often simulated via Euler-Maruyama integration and with $\Delta t = 1$ so that the reverse SDE generation process is given by:

$$g_\theta(X_\tau, \epsilon_\tau, \tau) = X_{\tau+1} = X_\tau + \frac{\sigma_\tau^2}{2} \nabla_{X_\tau} \log \pi_\tau(X_\tau) + u_\theta(X_\tau, \tau) + \sigma_\tau\, \epsilon_\tau.$$

2.2 Traning of Diffusion Samplers

The goal is to generate samples from the target probability distribution $\pi_0(X_0)$ by numerically integrating the SDE in Eq. 3 whose drift is a vector field $u_\theta(X_\tau, \tau)$ that is parametrized by a trainable neural network. The corresponding loss is typically based on a divergence between the underlying marginal distribution $q_\theta(X_0)$ induced by the reverse diffusion process and the target distribution $\pi_0(X_0)$. For this purpose, f-divergences are a popular choice(Csiszár, 1967):

$$D_f(P(X) \,||\, Q(X)) = \int Q(X) f\left(\frac{P(X)}{Q(X)}\right) dX,$$

where f is a convex function satisfying $f(1) = 0$ and P and Q are two probability distributions satisfying $P << Q$, i.e. P is absolutely continuous with respect to Q. The rKL, for example, corresponds to $f(t) = -\log(t)$. The choice of f significantly influences the learning dynamics in terms of mode-seeking vs. mass-covering properties. The rKL exhibits a mode-seeking behavior.

For expressive latent variable models like diffusion samplers the marginal sample probability $q_\theta(X_0)$ is typically intractable. In practice, loss functions are instead based on the joint distributions of the diffusion paths $X_{0:T}$. Diffusion paths of the forward process Eq. 2 are distributed according to $q_\theta(X_{0:T})$ and those of the reverse process Eq. 3 according to $p_\theta(X_{0:T})$.

For f-divergences the data processing inequality yields the following monotonicity relation $D_f(\pi_0(X_0) \,||\, q_\theta(X_0)) \leq D_f(\pi_0(X_{0:T}) \,||\, q_\theta(X_{0:T}))$ (Murphy, 2023). The right-hand side of this inequality is tractable for diffusion samplers and hence minimization of this variational upper bound represents a suitable learning objective. For the rKL this objective reads:

$$D_{KL}(q_\theta(X_{0:T}) \,||\, p_\theta(X_{0:T})) = \mathbb{E}_{X_{0:T} \sim q_\theta(X_{0:T})}\left[\log \frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})}\right] \tag{4}$$

In case of diffusion bridges as defined in Eq. 2 and Eq. 3 the corresponding time-discretized probability density ratios are given by:

$$\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})} = \frac{\pi_T(X_T)}{\pi_0(X_0)} \prod_{t=1}^{T} \frac{q_\theta(X_{t-1}|X_t)}{p_\theta(X_t|X_{t-1})},$$

where the conditional probability for a step in the reverse direction is given by

$$q_\theta(X_{t-1}|X_t) = \mathcal{N}(X_{t-1}; X_t + \frac{\sigma_t^2}{2} \nabla_{X_t} \log \pi_t(X_t) + u_\theta(X_t, t), \sigma_t^2)$$

and for a step in the forward direction by

$$p_\theta(X_t|X_{t-1}) = \mathcal{N}(X_t; X_{t-1} + \frac{\sigma_{t-1}^2}{2} \nabla_{X_{t-1}} \log \pi_{t-1}(X_{t-1}) - u_\theta(X_{t-1}, t-1), \sigma_{t-1}^2).$$

2.2.1 Reverse KL Divergence with reparametrization trick

The loss in Eq. 4 involves an expectation over the variational distribution which is Gaussian. This is a typical use case for the reparameterization trick (Glasserman, 2004) which is particularly popular in the context of stochastic gradient descent (Kingma & Welling, 2014; Rezende et al., 2014). This technique provides an elegant way of estimating gradients and often yields a lower variance than the log derivative trick (also known as score-function estimator or REINFORCE (Williams, 1992)). In the context of diffusion samplers, the reparameterization trick was introduced in (Zhang & Chen, 2022). In this method the trajectory $X_{0:T} \sim q_\theta(X_{0:T})$ is reparameterized as a function $f_\theta(\epsilon_{0:T}) := [g_{\theta,0}, ..., g_{\theta,T-1}]$ of independent Gaussian noise $\epsilon_t \sim \mathcal{N}(0, \mathrm{I})$ and the model parameters θ. Here, a sample at each time step is X_t is successively generated via a Euler-Maruyama update function $g_{\theta,t} := X_{t-1} = g_\theta(X_t, t, \epsilon_t)$, which is used to numerically integrate Eq. 3. Substituting this reparameterization into the rKL loss yields with slight abuse of notation:

$$D_{KL}(q_\theta(\epsilon_{0:T}) \,||\, p_\theta(\epsilon_{0:T})) = \mathbb{E}_{\epsilon_{0:T} \sim \mathcal{N}(0,I)}\left[\log \frac{q_\theta(f_\theta(\epsilon_{0:T}))}{p_\theta(f_\theta(\epsilon_{0:T}))}\right]. \tag{5}$$

This expression highlights that the gradients of the loss with respect to θ can propagate into the expectation into the deterministic function f_θ, enabling efficient gradient calculation. In diffusion samplers the frequent iterative application of $g_{\theta,t}$ is likely to contribute to the vanishing or exploding gradient problem, which might explain, why suboptimal behavior of rKL loss when minimized with usage of the reparametrization trick Zhang & Chen (2022).

2.2.2 Log Variance Loss

Richter et al. (2020) propose the LV loss as a gradient estimator for rKL-based loss functions in Bayesian variational inference and it has since then been frequently used in the context of diffusion samplers Richter et al. (2023); Vargas et al. (2024); Chen et al. (2024). In the context of GflowNets Bengio et al. (2021) this loss is also known as the Trajectory Balance (TB) loss. For diffusion bridges, i.e. when both the reverse process q_θ and the forward process p_θ involve learnable parameters θ the LV takes the following form:

$$D^{\omega}_{LV}(q_\theta(X_{0:T})\,||\,p_\theta(X_{0:T})) = \frac{1}{2}\mathbb{E}_{X_{0:T}\sim\omega}\left[\left(\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})} - b^{\omega}_{\theta}\right)^2\right], \tag{6}$$

where $b^{\omega}_{\theta} = \mathbb{E}_{X_{0:T}\sim\omega}\left[\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})}\right]$. The corresponding gradient is given by (see App. A.1):

$$\nabla_\theta D^{\omega}_{LV}(q_\theta(X_{0:T})\,||\,p_\theta(X_{0:T})) = \mathbb{E}_{X_{0:T}\sim\omega}\left[\left(\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})} - b^{\omega}_{\theta}\right)\cdot\nabla_\theta\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})}\right] \tag{7}$$

If $p_\theta(X_{0:T})$ does not contain learnable parameters this expression simplifies to:

$$\nabla_\theta D^{\omega}_{LV}(q_\theta(X_{0:T})\,||\,p(X_{0:T})) = \mathbb{E}_{X_{0:T}\sim\omega}\left[\left(\log\frac{q_\theta(X_{0:T})}{p(X_{0:T})} - b^{\omega}_{\theta}\right)\cdot\nabla_\theta\log q_\theta(X_{0:T})\right]. \tag{8}$$

When the proposal policy is chosen as $\omega = \mathrm{stop\_gradient}(q_\theta)$ the gradient of the LV loss is identical to the gradient of the rKL loss when it is trained with the log derivative trick combined with variance reduction (see Sec. 3.1). In this particular case, we will refer to this loss as the on-policy Log Variance (OP-LV) loss. For diffusion samplers, the general LV loss and the OP-LV loss were recently proposed in Richter et al. (2023) with the rationale that it represents a valid divergence. However, the LV loss is not an f-divergence and we present by a simple counter-example in App. A.5 that it can violate the data processing inequality. Consequently, we argue that its application to latent variable models is potentially problematic since it conflicts with the rationale of diffusion sampler training based on divergences of joint probabilities (Sec. 2.2). These considerations put the validity of the LV loss for diffusion bridges in question. However, we stress that the OP-LV when applied on variance preserving and variance exploding SDE-based diffusion samplers is unaffected by these arguments. This insight is relevant for instance in diffusion bridges with LV losses as in (Richter et al., 2023; Vargas et al., 2024; Chen et al., 2024).

3 Method

3.1 Reverse KL loss with log derivative trick and control variate

Several recent works demonstrate that the LV loss (Sec. 2.2.2 outperforms the rKL loss with reparametrization trick Richter et al. (2023). It is argued that this is due to the mode-seeking tendency associated with the rKL objective which results in mode collapse. While the aforementioned works applied the reparametrization trick in conjunction with the rKL objective we investigate the rKL objective with the log derivative gradient estimator. This combination remained comparably underexplored in the diffusion sampler setting so far. The gradient of this loss reads (App. A.2):

$$\begin{aligned}\nabla_\theta D_{KL}(q_\theta(X_{0:T})\,||\,p_\theta(X_{0:T})) =&\mathbb{E}_{X_{0:T}\sim q_\theta}\left[\left(\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})} - b^{q_\theta}_{\theta}\right)\nabla_\theta\log q_\theta(X_{0:T})\right]\\ &-\mathbb{E}_{X_{0:T}\sim q_\theta}\left[\nabla_\theta\log p_\theta(X_{0:T})\right],\end{aligned} \tag{9}$$

where $b^{q_\theta}_{\theta} = \mathbb{E}_{X_{0:T}\sim q_\theta}\left[\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})}\right]$ is used as a control variate. Comparing Eq. 8 with Eq. 9 shows that when $p_\theta(X_{0:T})$ does not have learnable parameters the gradients of the OP-LV loss and rKL-LD loss are identical. Conversely, when $\omega \neq \mathrm{stop\_gradient}(q_\theta)$ these two losses yield in general different gradients. Similarly, when the SDE parameters, that are shared by the forward and reverse process, are learned as described in Sec. 3.2, the gradient of the LV loss does in general not correspond to the gradient of the rKL loss. As pointed out in Sec. 3.2 this situation arises for diffusion bridges.

3.2 Learning of SDE parameters

By the chain rule for Shannon entropies Shannon (1948), we obtain the following upper bound on the entropy of the marginal distribution:

$$H(q_\theta(X_0)) \leq H(q_\theta(X_{0:T})). \tag{10}$$

Hence, to approximate a target distribution with unknown, potentially high entropy the diffusion bridge needs to be able to yield a $q_\theta(X_{0:T})$ with at least this entropy. For diffusion bridges as defined in Sec. 2.1 the entropy of the joint distribution is given by (App. A.4):

$$H(q_\theta(X_{0:T})) = H(\pi_T) + \frac{N}{2} \sum_{t=0}^{T-1} \left(1 + \log 2\pi\sigma_t^2\right), \tag{11}$$

Consequently, the upper bound for the entropy of the marginal $q_\theta(X_0)$ is determined by σ_t, the number of diffusion steps, and the entropy of the prior π_T. Practically, in our experiments we keep the diffusion coefficients constant across time, which is why we will denote it in the following as σ_{diff}. Several recent works (Blessing et al., 2024; Chen et al., 2024) learn the mean and the variance of the prior distribution π_T, which improves the exploration capabilities of the method, as this increases the entropy of the upper bound in Eq. 10. We build upon this approach by additionally learning individual entropy contributions for each dimension by learning $\sigma_{\text{diff}} \in \mathbb{R}^N$ (details in App. C.3). Our experiments in Sec. 5 show that this modification yields a significantly better performance.

3.3 Exploration Strategies

As mentioned above the tendency to result in mode-collapse is a frequent objection against rKL-based losses. For this reason, we propose the usage of exploration-based methods to alleviate the problem of mode collapse of the rKL-LD divergence.

Variational Annealing Drawing inspiration from recent advances in physics-inspired combinatorial optimization Hibat-Allah et al. (2021), we employ variational annealing to encourage exploration of the variational distribution. This approach has been shown to prevent mode collapse in discrete domain diffusion samplers for combinatorial optimization problems Sanokowski et al. (2024). The method involves initially approximating the target distribution π_0 at an inverse temperature $\beta < 1$ and then gradually increasing it to $\beta = 1$ (see Eq. 1). We adapt this technique to diffusion samplers by training the diffusion bridge along the interpolation path $\pi_t(X_t, \beta) = \pi_0(X_t, \beta)^{\eta(t)}\, \pi_T(X_t)^{1-\eta(t)}$.

Off-policy Sampling We introduce an off-policy training strategy that generates samples in regions where the reverse diffusion process has low probability densities. Inspired by MCMC literature Andrieu & Thoms (2008), we replace the Gaussian noise in the reverse diffusion process with noise from a heavy-tailed distribution. Specifically, we propose a mixture of Gaussian and Laplace distributions:

$$\widetilde{q_\theta}(X_{t-1}|X_t) = (1-\alpha)\,\mathcal{N}(X_{t-1}; X_t, \sigma_t) + \alpha\,\mathcal{L}(X_{t-1}; X_t, \gamma_t) \tag{12}$$

Here, α represents the probability of sampling from the Laplace distribution $\mathcal{L}(X_{t-1}; X_t, \gamma_t)$, and γ_t controls the Laplace distribution's variance. We set $\gamma_t = \kappa\sqrt{\frac{\pi}{2e}}\sigma_t$ with $\kappa = 1$, ensuring the Laplace distribution has the same entropy as the Gaussian distribution. The degree of exploration can be controlled by the choices of κ and α. We initialize α at a value $0 < \alpha_{\text{start}} \leq 1$ and decrease it linearly to zero, allowing the variational distribution to better adapt to the target distribution at the end of training. The off-policy distribution is incorporated into the rKL divergence using importance weights, resulting in the off-policy rKL-LD loss:

$$D_{KL}^{\widetilde{q_\theta}}(q_\theta(X_{0:T}) \,||\, p_\theta(X_{0:T})) = \mathbb{E}_{X_{0:T} \sim \widetilde{q_\theta}}\left[\frac{q_\theta(X_{0:T})}{\tilde{q}_\theta(X_{0:T})} \log \frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})}\right]$$

The gradient of this loss function is provided in App. A.3. In Sec. 5, we demonstrate that these exploration methods effectively alleviate the mode-seeking behavior of the rKL.

4 Related Work

For continuous diffusion samplers, the rKL loss is typically used with the reparametrization trick (Vargas et al., 2023; Berner et al., 2022). Zhang & Chen (2022) employ a more memory-efficient

ELBO (↑)	Seeds (26d)	Sonar (61d)	Credit (25d)	Brownian (32d)	LGCP (1600d)
CMCD-KL $\star$ (r)	$-73.51_{\pm 0.01}$	$-109.09_{\pm 0.01}$	$-507.23_{\pm 6.40}$	$0.86_{\pm 0.01}$	$478.75_{\pm 0.34}$
CMCD-LV $\star$ (r)	$-73.67_{\pm 0.01}$	$-109.50_{\pm 0.03}$	$-504.90_{\pm 0.02}$	$0.54_{\pm 0.03}$	$472.79_{\pm 0.44}$
CMCD-rKL $\star$	$-74.37_{\pm 0.01}$	$-109.69_{\pm 0.063}$	$-504.99_{\pm 0.016}$	$-0.30_{\pm 0.018}$	$\mathbf{471.91_{\pm 0.291}}$
CMCD-LV $\star$	$-74.13_{\pm 0.01}$	$-109.53_{\pm 0.062}$	$-504.91_{\pm 0.002}$	$-0.05_{\pm 0.002}$	$460.84_{\pm 0.099}$
CMCD-LV	$-73.53_{\pm 0.01}$	$-109.66_{\pm 0.015}$	$-628.39_{\pm 32.907}$ †	$-6.05_{\pm 0.028}$ †	$447.74_{\pm 0.0}$ †
CMCD-rKL-LD $\star$	$-74.10_{\pm 0.01}$	$-109.25_{\pm 0.007}$	$-504.88_{\pm 0.003}$	$0.36_{\pm 0.001}$	$466.73_{\pm 0.027}$
CMCD-rKL-LD	$\mathbf{-73.45_{\pm 0.01}}$	$\mathbf{-108.83_{\pm 0.005}}$	$\mathbf{-504.58_{\pm 0.001}}$	$\mathbf{1.06_{\pm 0.0}}$	$465.80_{\pm 0.02}$

Table 1: Results on Bayesian learning benchmarks. The ELBO (the higher the better) is reported for various methods and tasks. $\star$ denotes that σ_{diff} is not learned during training. Runs that diverge after reaching the minimum ELBO are denoted with †.

method based on stochastic adjoint sensitivity. However, this gradient estimation method was found to perform worse than reparametrization in the diffusion samplers in Berner et al. (2022). More recently, Richter et al. (2023) proposed the LV loss as an alternative and showed that it outperforms the rKL loss with the reparametrization trick. The corresponding experiments are performed with the Path Integral Sampler Zhang & Chen (2022) and the Time-Reversed Diffusion Sampler. However, they report that the application of LV to diffusion bridges results in bad performance and numerical instabilities. In (Vargas et al., 2024; Chen et al., 2024) the LV loss is employed in conjunction with diffusion bridges and both works report that it outperforms the rKL loss with the parametrization trick. Frequently, the mode-collapse tendency of rKL is given as an explanation for its inferior performance in the context of diffusion bridges (Richter et al., 2023). Successful applications of the rKL with other gradient estimators than the reparametrization trick can be found in discrete sampling problems. Examples of such problems arise in combinatorial optimization and statistical physics of spin lattices where Sanokowski et al. (2024; 2025) proposed the application of diffusion samplers based on the rKL-LD loss. Besides its mode-collapse tendency, the straightforward application of the rKL objective in diffusion samplers is criticized in Richter et al. (2023) for precluding the application of off-policy sampling methods. They already point out that it is conceivable to implement off-policy sampling strategies via an increased sampling noise in the simulation of SDEs. A learnable degree of exploration is realized in several diffusion-based methods in Blessing et al. (2024) where diffusion and friction parameters are treated as learnable parameters. However, in contrast to the present work, their implementation treats these parameters only as scalars.

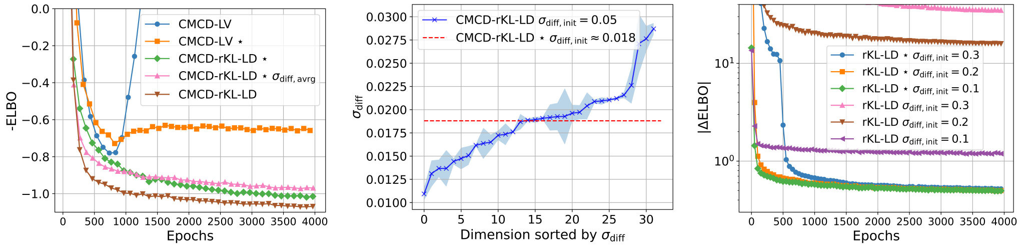

Figure 1: Models with fixed σ_{diff} are marked with $\star$. Left: Training curves on the Brownian task of CMCD trained with LV loss and rKL-LD loss. Middle: Plot of the learned σ_{diff} in ascending order at the end of training of the best run (CMCD-rKL-LD $\sigma_{\text{diff,init}} = 0.05$). On the left we show results of CMCD-rKL-LD $\star$ $\sigma_{\text{diff,init}} \approx 0.018$ where $\sigma_{\text{diff,init}}$ is initialized at the average value of the learned σ_{diff} at the end of training of CMCD-rKL-LD $\sigma_{\text{diff,init}} = 0.05$. Right: Training curves on the Seeds task, where CMCD-rKL-LD $\sigma_{\text{diff,init}}$ is compared to CMCD-rKL-LD $\star$ $\sigma_{\text{diff,init}}$ at different initializations of $\sigma_{\text{diff,init}}$.

5 Experiments

Learnable Diffusion Coefficients and Divergent Behavior of Log Variance Loss We first investigate the impact of learnable diffusion coefficients when combined with the rKL-LD loss, as shown in Fig. 1 (left and right). In Fig. 1 (left) we evaluate CMCD under several configurations on the Brownian Bayesian learning task (see App. B.1). Our baseline comparison is CMCD-LV $\star$, where the $\star$ denotes that σ_{diff} are not updated during training. This method proved highly sensitive to the

Task	Funnel (10d)		GMM-40 (50d)			MoS-10 (50d)		
Metric	Sinkhorn (↓)	ELBO (↑)	Sinkhorn (↓)	ELBO (↑)	EMC (↑)	Sinkhorn (↓)	ELBO (↑)	EMC (↑)
Ground Truth	$64.770_{\pm 64.770}$	$0.$	$875.21_{\pm 86.023}$	0	$1.$	$329.81_{\pm 42.036}$	$0.$	$1.$
CMCD-rKL ⋆	$113.38_{\pm 0.77}$	$-2.46_{\pm 0.35}$	$22451.59_{\pm 931.69}$	$-37.37_{\pm 0.10}$	$0.490_{\pm 0.201}$	$1504.11_{\pm 201.42}$	**$-19.88_{\pm 0.16}$**	$0.628_{\pm 0.048}$
CMCD-LV ⋆	**$95.90_{\pm 2.58}$**	$-0.67_{\pm 0.00}$	$2689.87_{\pm 215.51}$	$-37.37_{\pm 0.10}$	**$0.996_{\pm 0.001}$**	$1106.72_{\pm 181.88}$	$-52.52_{\pm 0.70}$	$0.971_{\pm 0.001}$
CMCD-LV	$102.94_{\pm 3.04}$ †	$-0.46_{\pm 0.01}$ †	$2756.96_{\pm 240.53}$	$-45.85_{\pm 0.17}$	**$0.996_{\pm 0.001}$**	$974.38_{\pm 118.30}$	$-43.63_{\pm 0.51}$	**$0.994_{\pm 0.001}$**
CMCD-rKL-LD ⋆	**$94.04_{\pm 2.239}$**	$-0.54_{\pm 0.01}$	$2464.37_{\pm 222.27}$	$-26.78_{\pm 0.05}$	**$0.997_{\pm 0.001}$**	**$630.82_{\pm 55.22}$**	$-34.93_{\pm 0.25}$	$0.981_{\pm 0.004}$
CMCD-rKL-LD	**$94.16_{\pm 2.55}$**	**$-0.23_{\pm 0.01}$**	**$2426.40_{\pm 160.27}$**	**$-21.94_{\pm 0.10}$**	**$0.997_{\pm 0.000}$**	**$630.81_{\pm 55.44}$**	$-34.93_{\pm 0.25}$	$0.981_{\pm 0.004}$

Table 2: Results on synthetic learning benchmarks. The Sinkhorn distance (the lower the better) is reported for various methods and tasks. Runs that diverge after reaching the reported value marked with † and runs that do not converge at all are denoted as N/A. Sinkhorn distances are computed on Funnel using 2000 samples and on GMM and MoS using 16000 samples. Ground truth sinkhorn distances are computed by calculating the sinkhorn distance between two independent set of samples from the target distribution.

Task	GMM (5d)		MoS (10d)	
Metric	Sinkhorn (↓)	ELBO (↑)	Sinkhorn (↓)	ELBO (↑)
CMCD-rKL-LD $\sigma_{\text{prior,init}} = 1$	3083.75	-3.68	311.11	-2.39
CMCD-rKL-LD tune $\sigma_{\text{prior,init}}$	1160.30	**-1.44**	222.9	-1.16
CMCD-rKL-LD annealing	**146.23**	-2.95	**42.13**	**-0.50**
CMCD-rKL-LD off-policy $\sigma_{\text{prior,init}} = 1$	3069	-3.84	44.68	-0.65
CMCD-rKL-LD off-policy tune $\sigma_{\text{prior,init}}$	996.84	-1.64	44.68	-0.65

Table 3: Comparison of different exploration methods on GMM 5d and MoS 10d. $\sigma_{\text{prior,init}} = 1$ stands for runs that were initialized with a prior with standard deviation of 1. Tune $\sigma_{\text{prior,init}}$ denotes that the standard deviation of the prior was tuned to the problem. Annealing and off-policy denote the exploration methods presented in Sec. 3.3.

initial choices of σ_{prior} and σ_{diff} (see App. C). When we train the diffusion parameters σ_{diff} with the LV loss (denoted as CMCD-LV), we consistently observe divergent behavior across all hyperparameter choices. For comparison, we study our proposed loss function in two variants: CMCD-rKL-LD with trainable σ_{diff}, and CMCD-rKL-LD ⋆ with frozen σ_{diff}. For CMCD-rKL-LD ⋆ we performed hyperparameter tuning on the initial value of σ_{diff} which we call $\sigma_{\text{diff,init}}$. The experimental results demonstrate that the rKL-LD loss consistently outperforms the LV loss across all tested configurations and that incorporating learnable diffusion coefficients further enhances model performance when using this loss function. The results in Tab. 1 and Tab. 2 indicate that the divergent behavior is present on most investigated benchmarks.

In Fig. 1 (middle), we analyze the learned diffusion coefficients σ_{diff} across dimensions, showing their values in ascending order along with standard deviations computed from three independent seeds. The results reveal that σ_{diff} systematically adopts different scales across dimensions while maintaining consistency between seeds. To test the hypothesis, whether different values of σ_{diff} in each dimension are indeed beneficial, we additionally train CMCD-rKL-LD with frozen sigma parameters initialized at the average σ_{diff} after training of the best run (CMCD-rKL-LD ⋆ $\sigma_{\text{diff,init}} = \sigma_{\text{diff,avg}}$ in Fig. 1 (left)). The results show that this uniform choice of σ_{diff} across dimensions yields inferior results.

Figure 1 (right) shows how the initial value of $\sigma_{\text{diff,init}}$ affects performance by comparing CMCD-rKL-LD with and without learned diffusion on the Seeds dataset. We track the convergence using ΔELBO, defined as $|\text{ELBO}_{\text{opt}} - \text{ELBO}|$, where we estimate $\text{ELBO}_{\text{opt}} = -73\,\text{nats}$ to enable visualization on a logarithmic scale. The results demonstrate that when CMCD learns $\sigma_{\text{diff,init}}$, it successfully approximates the target distribution regardless of the initial parameter choice. Without learned diffusion parameters, the model struggles to efficiently approximate the target distribution.

Benchmarks Similarly to Chen et al. (2024), we evaluate our model on two types of tasks: Bayesian learning problems, where we report the ELBO due to the absence of ground truth data (see App. B), and synthetic targets, where we can report the Sinkhorn distance between model samples and the target distribution (see App. B) together with the ELBO and Entropic Mode Coverage (EMC) Blessing et al. (2024) at the same training iteration. On multimodal tasks, a combination of high ELBO and low Sinkhorn distance indicate good performance. Detailed descriptions of both task types are provided in App. B.1. Our experimental setup mirrors Chen et al. (2024), i.e. training

CMCD for 40.000 training iterations with a batch size of 2.000 on all tasks and a batch size of 300 for LGCP. Models are trained using 128 diffusion steps. We use a simple architecture for all diffusion-based methods on Bayesian tasks as described in App. C.3. On GMM-40 50d and MoS-10 50d, we use the PISgradnet architecture from Vargas et al. (2024) as we observed that this architecture is less prone to mode-collapse when hyperparameters are carefully tuned. In our experiments, we compare different variations of CMCD trained with different loss functions, denoted as rKL (Eq. 5), rKL-LD (Eq. 9), and LV (Eq. 7). For each loss, we report the results of a variation, where σ_{diff} is not learned which we denote with a $\star$. For rKL-LD and LV we also report the results of the method when σ_{diff} is learned. For each variation, we perform a grid search over the learning rate, $\sigma_{\text{diff,init}}$ and $\sigma_{\text{prior,init}}$ as described in App. C. Runs that diverge, i.e. the ELBO decreases to high magnitudes after the reported best metric value is reached, are denoted with a †. Tab. 1 and Tab. 2 present our results on Bayesian and synthetic targets, respectively. Our results show that on Bayesian tasks, CMCD-rKL-LD $\star$ significantly outperforms CMCD-LV $\star$ in 5 out of 5 tasks and CMCD-rKL on 4 out of 5 tasks. If we additionally train the diffusion coefficients, we observe that CMCD-rKL-LD improves upon CMCD-rKL-LD $\star$ on 4 out of 5 tasks. In contrast to that, we observe that learning diffusion coefficients in combination with the LV loss deteriorate the performance of CMCD-rKL-LD in 4 out of 5 cases. In fact, this often leads to unstable learning dynamics as the runs diverge in 3 out of 5 cases. On synthetic tasks, CMCD-rKL-LD significantly achieves the best Sinkhorn distance on MoS 50d and insignificantly better Sinkhorn distance than CMCD-LV $\star$ and CMCD-rKL-LD $\star$ on Funnel and GMM40 50d. In terms of ELBO, CMCD-rKL-LD $\star$ and CMCD-rKL-LD achieve better results than CMCD-LV $\star$ and CMCD-LV in 3 out of 3 tasks. All methods except CMCD-rKL $\star$ achieve an EMC value close to $1.$, indicating that all modes are covered.

Exploration Stratergies In the following, we compare different methods to prevent the mode collapse in multimodal target distributions such as GMM 5d and MoS 10d. Here, we compare variational annealing (CMCD-rKL-LD Annealing) and off-policy learning (CMCD-rKL-LD off-policy) as introduced in Sec. 3.3 to careful tuning of $\sigma_{\text{prior,init}}$ (CMCD-rKL-LD tune). In variational annealing, we tune the starting inverse temperature β_{start}, and in off-policy learning the mixing probability α_{start} is tuned. For off-policy and variational annealing, we assume no prior knowledge of the problem and set $\sigma_{\text{prior,init}}$ = 1, except for CMCD-rKL-LD off-policy where we also show results of the method where $\sigma_{\text{prior,init}}$ is tuned. Results in Tab. 3 show that variational annealing obtains the best results in terms of Sinkhorn distance on both datasets and the best ELBO value on MoS 10d. While off-policy learning exhibits strong performance on MoS 10d it yields no improvement on GMM 5d when $\sigma_{\text{prior,init}}$ is not tuned. However, when we additionally tune $\sigma_{\text{prior,init}}$ the method improves and helps to prevent mode collapse. Only tuning $\sigma_{\text{prior,init}}$ does not help against mode collapse and yields the worst results.

6 Conclusion

In this work, we introduced a novel training approach for diffusion bridge-based samplers using gradients of the reverse Kullback-Leibler divergence estimated with the log derivative trick (rKL-LD). Our analysis reveals a critical insight: while the Log Variance (LV) loss and reverse KL loss are equivalent when training only the reverse diffusion process, this equivalence breaks down in two important scenarios - when working with diffusion bridges or learning diffusion coefficients. Our theoretical consideration show that in general the LV loss does not satisfy the data processing inequality, questioning its soundness in the context of diffusion bridges. Empirical results demonstrate the superiority of the proposed rKL-LD loss over the recently proposed LV loss. Notably, when using the rKL-LD loss, we can further improve diffusion bridges by learning the diffusion coefficients, which also reduces the sensitivity to hyperparameter choices. In contrast, the LV loss often exhibits unstable behavior during coefficient learning. Additionally, we demonstrated that exploration methods can effectively prevent mode collapse when dealing with multimodal target distributions. These findings open new avenues for improving the stability and performance of diffusion-based generative models.

Acknowledgments

The ELLIS Unit Linz, the LIT AI Lab, the Institute for Machine Learning, are supported by the Federal State Upper Austria. We thank the projects INCONTROL-RL (FFG-881064), PRIMAL (FFG-873979), S3AI (FFG-872172), DL for GranularFlow (FFG-871302), EPILEPSIA

(FFG892171), FWF AIRI FG 9-N (10.55776/FG9), AI4GreenHeatingGrids (FFG-899943), INTEGRATE (FFG-892418), ELISE (H2020-ICT-2019-3 ID: 951847), Stars4Waters (HORIZON-CL6-2021- CLIMATE-01-01). We thank NXAI GmbH, Audi.JKU Deep Learning Center, TGW LOGISTICS GROUP GMBH, Silicon Austria Labs (SAL), FILL Gesellschaft mbH, Anyline GmbH, Google, ZF Friedrichshafen AG, Robert Bosch GmbH, UCB Biopharma SRL, Merck Healthcare KGaA, Verbund AG, GLS (Univ. Waterloo), Software Competence Center Hagenberg GmbH, Borealis AG, TÜV Austria, Frauscher Sensonic, TRUMPF and the NVIDIA Corporation. We acknowledge EuroHPC Joint Undertaking for awarding us access to Meluxina, Vega, and Karolina at IT4Innovations.

REFERENCES

Abien Fred Agarap. Deep learning using rectified linear units (relu). *CoRR*, abs/1803.08375, 2018. URL http://arxiv.org/abs/1803.08375.

Christophe Andrieu and Johannes Thoms. A tutorial on adaptive mcmc. *Statistics and computing*, 18:343–373, 2008.

Michael Arbel, Alex Matthews, and Arnaud Doucet. Annealed flow transport monte carlo. In *International Conference on Machine Learning*, pp. 318–330. PMLR, 2021.

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016. URL http://arxiv.org/abs/1607.06450.

Yoshua Bengio, Tristan Deleu, Edward J. Hu, Salem Lahlou, Mo Tiwari, and Emmanuel Bengio. Gflownet foundations. *CoRR*, abs/2111.09266, 2021. URL https://arxiv.org/abs/2111.09266.

Julius Berner, Lorenz Richter, and Karen Ullrich. An optimal control perspective on diffusion-based generative modeling. *arXiv preprint arXiv:2211.01364*, 2022.

Denis Blessing, Xiaogang Jia, Johannes Esslinger, Francisco Vargas, and Gerhard Neumann. Beyond elbos: A large-scale evaluation of variational methods for sampling. *arXiv preprint arXiv:2406.07423*, 2024.

Junhua Chen, Lorenz Richter, Julius Berner, Denis Blessing, Gerhard Neumann, and Anima Anandkumar. Sequential controlled langevin diffusions. *arXiv preprint arXiv:2412.07081*, 2024.

Imre Csiszár. On information-type measure of difference of probability distributions and indirect observations. *Studia Sci. Math. Hungar.*, 2:299–318, 1967.

Marco Cuturi, Laetitia Meng-Papaxanthos, Yingtao Tian, Charlotte Bunne, Geoff Davis, and Olivier Teboul. Optimal transport tools (ott): A jax toolbox for all things wasserstein. *arXiv preprint arXiv:2201.12324*, 2022.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709, 2021.

Tomas Geffner and Justin Domke. Langevin diffusion variational inference. In *International Conference on Artificial Intelligence and Statistics*, pp. 576–593. PMLR, 2023.

P Glasserman. Monte carlo methods in financial engineering, 2004.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Jeremy Heng, Adrian N Bishop, George Deligiannidis, and Arnaud Doucet. Controlled sequential monte carlo. *The Annals of Statistics*, 48(5):2904–2929, 2020.

Mohamed Hibat-Allah, Estelle M. Inack, Roeland Wiersema, Roger G. Melko, and Juan Carrasquilla. Variational neural annealing. *Nat. Mach. Intell.*, 3(11):952–961, 2021. doi: 10.1038/s42256-021-00401-3. URL https://doi.org/10.1038/s42256-021-00401-3.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014. URL https://arxiv.org/abs/1312.6114.

Fenglin Liu, Xuancheng Ren, Zhiyuan Zhang, Xu Sun, and Yuexian Zou. Rethinking skip connection with layer normalization in transformers and resnets. *CoRR*, abs/2105.07205, 2021. URL https://arxiv.org/abs/2105.07205.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=rkgz2aEKDr.

Jesper Møller, Anne Randi Syversveen, and Rasmus Plenge Waagepetersen. Log gaussian cox processes. *Scandinavian journal of statistics*, 25(3):451–482, 1998.

Kevin P Murphy. *Probabilistic machine learning: Advanced topics*. MIT press, 2023.

Radford M Neal. Slice sampling. *The annals of statistics*, 31(3):705–767, 2003.

Frank Noé and Hao Wu. Boltzmann generators - sampling equilibrium states of many-body systems with deep learning. *CoRR*, abs/1812.01729, 2018. URL http://arxiv.org/abs/1812.01729.

Nikolas Nüsken and Lorenz Richter. Solving high-dimensional hamilton–jacobi–bellman pdes using neural networks: perspectives from the theory of controlled diffusions and measures on path space. *Partial differential equations and applications*, 2(4):48, 2021.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning (ICML)*, 2014.

Lorenz Richter, Ayman Boustati, Nikolas Nüsken, Francisco Ruiz, and Omer Deniz Akyildiz. Vargrad: a low-variance gradient estimator for variational inference. *Advances in Neural Information Processing Systems*, 33:13481–13492, 2020.

Lorenz Richter, Julius Berner, and Guan-Horng Liu. Improved sampling via learned diffusions. *arXiv preprint arXiv:2307.01198*, 2023.

Sebastian Sanokowski, Sepp Hochreiter, and Sebastian Lehner. A diffusion model framework for unsupervised neural combinatorial optimization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 43346–43367. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/sanokowski24a.html.

Sebastian Sanokowski, Wilhelm Berghammer, Sebastian Sanokowski, P. Wang Haoyu, Sepp Martin Ennemoserand Hochreiter, and Sebastian Lehner. Scalable discrete diffusion samplers: Combinatorial optimization and statistical physics. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=peNgxpbdxB.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Andy Shih and Stefano Ermon. Probabilistic circuits for variational inference in discrete graphical models. *Advances in neural information processing systems*, 33:4635–4646, 2020.

Francisco Vargas, Will Grathwohl, and Arnaud Doucet. Denoising diffusion samplers. *arXiv preprint arXiv:2302.13834*, 2023.

Francisco Vargas, Shreyas Padhy, Denis Blessing, and N Nüsken. Transport meets variational inference: Controlled monte carlo diffusions. In *The Twelfth International Conference on Learning Representations*, 2024.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256, 1992. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Dian Wu, Lei Wang, and Pan Zhang. Solving statistical mechanics using variational autoregressive networks. *Phys. Rev. Lett.*, 122:080602, Feb 2019. doi: 10.1103/PhysRevLett.122.080602. URL https://link.aps.org/doi/10.1103/PhysRevLett.122.080602.

Qinsheng Zhang and Yongxin Chen. Path integral sampler: A stochastic control approach for sampling. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=_uCb2ynRu7Y.

A Derivations and Proofs

A.1 Gradient of the Log Variance Loss

In the following we derive the gradient of the log variance loss:

$$
\begin{aligned}
\nabla_\theta D^{\omega}_{LV}(q_\theta, p) &= \nabla_\theta \mathbb{E}_{X_{0:T}\sim\omega}\left[\left(\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})} - \mathbb{E}_{X_{0:T}\sim\omega}\left[\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})}\right]\right)^2\right] \\
&= \mathbb{E}_{X_{0:T}\sim\omega}\left[2\left(\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})} - \mathbb{E}_{X_{0:T}\sim\omega}\left[\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})}\right]\right)\cdot\left(\nabla_\theta\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})} - \mathbb{E}_{X_{0:T}\sim\omega}\left[\nabla_\theta\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})}\right]\right)\right] \\
&= \mathbb{E}_{X_{0:T}\sim\omega}\left[2\left(\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})} - \mathbb{E}_{X_{0:T}\sim\omega}\left[\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})}\right]\right)\cdot\nabla_\theta\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})}\right]
\end{aligned}
$$

Where we have used that $\mathbb{E}_{X_{0:T}\sim\omega}\left[2\left(\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})} - \mathbb{E}_{X_{0:T}\sim\omega}\left[\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})}\right]\right)\cdot\left(\mathbb{E}_{X_{0:T}\sim\omega}\left[\nabla_\theta\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})}\right]\right)\right] = 0$

A.2 Gradient of the Reverse Kullback-Leibler Divergence Loss

In the following the gradient of the rKL divergence is derived, when it is optimized with the usage of the log derivative trick:

$$
\begin{aligned}
\nabla_\theta D_{KL}(q_\theta(X_{0:T})\,||\,p_\theta(X_{0:T})) &= \nabla_\theta \mathbb{E}_{X_{0:T}\sim q_\theta}\left[\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})}\right] \\
&= \mathbb{E}_{X_{0:T}\sim q_\theta}\left[\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})}\nabla_\theta\log q_\theta(X_{0:T})\right] + \mathbb{E}_{X_{0:T}\sim q_\theta}\left[\nabla_\theta\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})}\right] \\
&= \mathbb{E}_{X_{0:T}\sim q_\theta}\left[\left(\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})} - \mathbb{E}_{X_{0:T}\sim q_\theta}\left[\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})}\right]\right)\nabla_\theta\log q_\theta(X_{0:T})\right] + \mathbb{E}_{X_{0:T}\sim q_\theta}\left[\nabla_\theta\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})}\right] \\
&= \mathbb{E}_{X_{0:T}\sim q_\theta}\left[\left(\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})} - \mathbb{E}_{X_{0:T}\sim q_\theta}\left[\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})}\right]\right)\nabla_\theta\log q_\theta(X_{0:T})\right] - \mathbb{E}_{X_{0:T}\sim q_\theta}\left[\nabla_\theta\log p_\theta(X_{0:T})\right]
\end{aligned}
$$

where we use in lines two to three the fact that $b\,\mathbb{E}_{X_{0:T}\sim q_\theta}\left[\nabla_\theta\log q_\theta(X_{0:T})\right] = 0$ and that $b = \mathbb{E}_{X_{0:T}\sim q_\theta}\left[\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})}\right]$ is a baseline that leads to gradient updates with lower variance.

A.3 Off-policy Gradient of the Reverse Kullback-Leibler Divergence Loss

In off-policy optimization, the gradient of the rKL can be adapted by changing the expectations from the on-policy distribution $q_\theta(X_{0:T})$ to an expectation over samples from the off-policy distribution $\tilde{q}_\theta(X_{0:T})$.

Therefore we first rewrite

$$
\nabla_\theta D_{KL}(q_\theta(X_{0:T})\,||\,p_\theta(X_{0:T})) = \mathbb{E}_{X_{0:T}\sim q_\theta}\left[\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})}\nabla_\theta\log q_\theta(X_{0:T})\right] + \mathbb{E}_{X_{0:T}\sim q_\theta}\left[\nabla_\theta\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})}\right]
$$

to

$$
\begin{aligned}
\nabla_\theta D_{KL}(q_\theta(X_{0:T})\,||\,p_\theta(X_{0:T})) = {} & \mathbb{E}_{X_{0:T}\sim\tilde{q}_\theta}\left[w(X_{0:T})\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})}\nabla_\theta\log q_\theta(X_{0:T})\right] \\
& + \mathbb{E}_{X_{0:T}\sim\tilde{q}_\theta}\left[w(X_{0:T})\nabla_\theta\log\frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})}\right],
\end{aligned} \tag{13}
$$

where $w(X_{0:T}) = \frac{q_\theta(X_{0:T})}{\tilde{q}_\theta(X_{0:T})}$. In practice, we use self-normalized importance sampling for numerical stability.

Variance reduction can is applied by computing the baseline with $b = \mathbb{E}_{X_{0:T} \sim \tilde{q}_\theta}\left[\log \frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})}\right]$. The baseline is then subtracted from Eq 13 yields the final gradient update formula as:

$$\begin{aligned}\nabla_\theta D_{KL}(q_\theta(X_{0:T}) \,||\, p_\theta(X_{0:T})) = \mathbb{E}_{X_{0:T} \sim \tilde{q}_\theta}\left[w(X_{0:T})\left(\log \frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})} - b\right)\nabla_\theta \log q_\theta(X_{0:T})\right] \\ + \mathbb{E}_{X_{0:T} \sim \tilde{q}_\theta}\left[w(X_{0:T})\nabla_\theta \log \frac{q_\theta(X_{0:T})}{p_\theta(X_{0:T})}\right].\end{aligned}$$

A.4 Entropy of joint variational reverse diffusion process

In the following, we derive that:

$$H(q_\theta(X_{0:T})) = H(\pi_T) + \frac{N}{2}\sum_{t=1}^{T}\left[1 + \frac{1}{2}\log(2\pi\sigma_t^2)\right]$$

To show this we compute

$$H(q_\theta(X_{t-1}|X_t)) = -\mathbb{E}_{X_{t-1} \sim q_\theta(X_{t-1}|X_t)}\left[\log(q_\theta(X_{t-1}|X_t))\right] \tag{14}$$

by using $q_\theta(X_{t-1}|X_t) = \mathcal{N}(X_{t-1}; X_t + u_\theta(X_t), \sigma_t)$ and with the usage of the parametrization trick $X_{t-1} = X_t + u_\theta(X_t) + \epsilon\sigma_t$ we can show that:

$$\begin{aligned}\log \mathcal{N}(X_{t-1}; X_t + u_\theta(X_t), \sigma_t) &= -\frac{(X_{t-1} - (X_t + u_\theta(X_t)))^2}{2\sigma^2} - \frac{N}{2}\log(2\pi\sigma_t^2) \\ &= -\epsilon_t^2/2 - \frac{N}{2}\log(2\pi\sigma_t^2)\end{aligned}$$

Due to the parametrization trick, we can rewrite the expectation over X_t to an expectation over σ_t.

Therefore, we have

$$H(q(X_{t-1}|X_t)) = -\mathbb{E}_{\epsilon_t}\left[-\epsilon_t^2/2 - \frac{N}{2}\log(2\pi\sigma_t^2)\right] = \frac{N}{2}\left[1 + \log(2\pi\sigma_t^2)\right] \tag{15}$$

We can now show with

$$\begin{aligned}H(q_\theta(X_{t-1}|X_t)) &= -\mathbb{E}_{X_{0:T} \sim q_\theta(X_{0:T})}\left[\log q_\theta(X_{0:T})\right] = -\mathbb{E}_{X_{0:T} \sim q_\theta(X_{0:T})}\left[\log \pi_T(X_T) + \sum_{t=1}^{T}\log q_\theta(X_{t-1}|X_t)\right] \\ &= H(\pi_T) - \sum_{t=1}^{T}\mathbb{E}_{X_{T:t-1} \sim q_\theta(X_{T:t-1})}\left[\log q_\theta(X_{t-1}|X_t)\right] \\ &= H(\pi_T) + \sum_{t=1}^{T}\mathbb{E}_{X_{T:t} \sim q_\theta(X_{T:t})}\left[H(q_\theta(X_{t-1}|X_t))\right] \\ &= H(\pi_T) + \frac{N}{2}\sum_{t=1}^{T}\mathbb{E}_{X_{T:t} \sim q_\theta(X_{T:t})}\left[1 + \log(2\pi\sigma_t^2)\right] \\ &= H(\pi_T) + \frac{N}{2}\sum_{t=1}^{T}\left[1 + \log(2\pi\sigma_t^2)\right]\end{aligned}$$

A.5 Counterexample to Data Processing Inequality for the Log Variance Loss

Let p, q be probability distributions, similar as in Eq. 6 but without learnable parameters, with respective marginal and conditional distributions and $(X, Y) \sim q$. Then the data processing inequality for the LV loss is

$$\mathrm{Var}_{(X,Y)\sim q}\left[\log\frac{q(X)}{p(X)}\right] \leq \mathrm{Var}_{(X,Y)\sim q}\left[\log\frac{q(X,Y)}{p(X,Y)}\right] \tag{16}$$

which can be decomposed such that

$$\begin{aligned}\mathrm{Var}_{(X,Y)\sim q}\left[\log\frac{q(X,Y)}{p(X,Y)}\right] =& \mathrm{Var}_{(X,Y)\sim q}\left[\log\frac{q(X)}{p(X)}\right] + \mathrm{Var}_{(X,Y)\sim q}\left[\log\frac{q(Y|X)}{p(Y|X)}\right] \\ &+ 2\,\mathrm{Cov}_{(X,Y)\sim q}\left[\log\frac{q(Y|X)}{p(Y|X)}, \log\frac{q(X)}{p(X)}\right].\end{aligned}$$

Therefore, if we find distributions p, q such that

$$\mathrm{Var}_{(X,Y)\sim q}\left[\log\frac{q(Y|X)}{p(Y|X)}\right] + 2\,\mathrm{Cov}_{(X,Y)\sim q}\left[\log\frac{q(Y|X)}{p(Y|X)}, \log\frac{q(X)}{p(X)}\right] \leq 0$$

we disprove inequality Eq. 16. For this, let $\mathcal{X} := \{0, 1\}$, p, q be defined on $\mathcal{X} \times \mathcal{X} := \mathcal{X}^2$ with marginal probabilities

$$p(X=0) = 0.1,\ p(X=1) = 0.9, \quad q(X=0) = 0.9,\ q(X=1) = 0.1, \quad q(Y=0) = 1,\ q(Y=1) = 0$$

and conditional probabilities

$$p(Y=0|X=0) = 0.5,\ p(Y=0|X=1) = 0.1.$$

From $q(Y = 1) = 0$ we get $q(X, Y = 1) = 0$. By standard arguments e.g., as used for KL-divergence we can interpret

$$q(X, Y=1)\log q(Y=1|X) = q(X)q(Y=1|X)\log q(Y=1|X)$$

as being zero since $\lim_{x\to 0^+} x\log x = 0$ which results in the following simplifications

$$\begin{aligned}\mathrm{Var}_{(X,Y)\sim q}\left[\log\frac{q(Y|X)}{p(Y|X)}\right] &= \sum_{(x,y)\in\mathcal{X}^2} q(x,y)\left(\log\frac{q(y|x)}{p(y|x)} - \sum_{(x',y')\in\mathcal{X}^2} q(x',y')\log\frac{q(y'|x')}{p(y'|x')}\right)^2 \\ &= \sum_{x\in\mathcal{X}} q(x,0)\left(\log\frac{q(0|x)}{p(0|x)} - \sum_{x'\in\mathcal{X}} q(x',0)\log\frac{q(0|x')}{p(0|x')}\right)^2 \\ &= \sum_{x\in\mathcal{X}} q(x)\left(\log p(0|x) - \sum_{x'\in\mathcal{X}} q(x')\log p(0|x')\right)^2\end{aligned}$$

by using $q(x, 0) = q(x)$. Analogously we get for

$$\begin{aligned}&\mathrm{Cov}_{(X,Y)\sim q}\left[\log\frac{q(Y|X)}{p(Y|X)}, \log\frac{q(X)}{p(X)}\right] \\ &= -\sum_{x\in\mathcal{X}} q(x)\left(\log p(0|x) - \sum_{x'\in\mathcal{X}} q(x')\log p(0|x')\right)\left(\log\frac{q(x)}{p(x)} - \sum_{x'\in\mathcal{X}} q(x')\log\frac{q(x')}{p(x')}\right).\end{aligned}$$

By inserting the corresponding probability values we have

$$\mathrm{Cov}_{(X,Y)\sim q}\left[\log\frac{q(Y|X)}{p(Y|X)}, \log\frac{q_X(X)}{p_X(X)}\right] \approx -0.6365, \quad \mathrm{Var}_{(X,Y)\sim q}\left[\log\frac{q(Y|X)}{p(Y|X)}\right] \approx 0.2331$$

which demonstrates that the data processing inequality does not always hold for the LV loss.

B Metrics

Evidence Lower Bound The evidence lower bound is a lower bound on $\log \mathcal{Z}$ and is computed with:

$$\text{ELBO} = \mathbb{E}_{X_{0:T} \sim q_\theta(X_{0:T})} \left[\frac{p_\theta(X_{0:T})}{q_\theta(X_{0:T})} \right] \leq \log \mathcal{Z} \tag{17}$$

Sinkhorn Distance The Sinkhorn distance is an entropic regularization of the 2-Wasserstein ($\mathcal{W}_2$) optimal transport (OT) distance Peyré et al. (2019), providing a principled alternative to ELBO for evaluating sample quality. Unlike ELBO, which is often insensitive to mode collapse Blessing et al. (2024), the Sinkhorn distance measures the discrepancy between generated and target distributions, offering better insights into sample diversity and multimodal coverage. As it requires access to ground-truth samples, its use is limited to synthetic tasks Chen et al. (2024). Following Blessing et al. (2024); Chen et al. (2024), we compute the Sinkhorn distance using the `ott` package Cuturi et al. (2022) and report it as a primary metric for the appropriate benchmarks.

Entropic Mode Coverage The Entropic Mode Coverage Blessing et al. (2024) is given by

$$\text{EMC} := \mathbb{E}_{X_0 \sim q_\theta} \left[\mathcal{H}(p(\xi, X_0)) \right] = -\frac{1}{N} \sum_{X_0 \sim q_\theta} \sum_{i=1}^{M} p(\xi, X_0) \log_M p(\xi, X_0)$$

where $i \in \{1, ..., M\}$ and $p(\xi_i, X_0)$ is the probability corresponding to the mixture component with the highest likelihood at X_0. The optimal value of EMC is 1, i.e. every mode from the target distribution is covered and it is 0. in the worst case.

B.1 Benchmarks

B.1.1 Bayesian Learning tasks

These tasks involve probabilistic inference where the true underlying parameter distributions are unknown, requiring Bayesian approaches for estimation.

Bayesian Logistic Regression (Sonar and Credit). We consider Bayesian logistic regression for binary classification on two well-established benchmark datasets, frequently used for evaluating variational inference and Markov Chain Monte Carlo (MCMC) methods. The model's posterior distribution is given by:

$$\rho_{\text{target}}(x) = p(x) \prod_{i=1}^{n} \text{Bernoulli}\left(y_i; \text{sigmoid}(x \cdot u_i)\right)$$

where the dataset consists of standardized input-output pairs $((u_i, y_i))_{i=1}^n$. Our evaluation includes the Sonar dataset ($d = 61, n = 208$) and the German Credit dataset ($d = 25, n = 1000$). The prior distribution is chosen as a standard Gaussian $p = \mathcal{N}(0, I)$ for Sonar, whereas for German Credit, we follow the implementation of Blessing et al. (2024), which omits an explicit prior by setting $p \equiv 1$.

Random Effect Regression (Seeds). The Seeds dataset ($d = 26$) is modeled using a hierarchical random effects regression framework, which captures both fixed and random effects to account for variability across different experimental conditions. The generative process is specified as:

$$\begin{aligned}
&\tau \sim \text{Gamma}(0.01, 0.01) \\
&a_0, a_1, a_2, a_{12} \sim \mathcal{N}(0, 10) \\
&b_i \sim \mathcal{N}\left(0, \frac{1}{\sqrt{\tau}}\right), \quad i = 1, \ldots, 21, \\
&\text{logits}_i = a_0 + a_1 x_i + a_2 y_i + a_{12} x_i y_i + b_1, \quad i = 1, \ldots, 21, \\
&r_i \sim \text{Binomial}\left(\text{logits}_i, N_i\right), \quad i = 1, \ldots, 21.
\end{aligned}$$

The inference task involves estimating the posterior distributions of τ, a_0, a_1, a_2, a_{12}, and the random effects b_i, given observed values of x_i, y_i, and N_i. This model is particularly relevant for analyzing seed germination proportions, where the inclusion of random effects accounts for heterogeneity in experimental conditions; see Geffner & Domke (2023) for further details.

Time Series Models (Brownian). The Brownian motion model ($d = 32$) represents a discretized stochastic process commonly used in time series analysis, with Gaussian observation noise. The generative model follows:

$$
\begin{aligned}
\alpha_{\text{inn}} &\sim \text{LogNormal}(0,2), \\
\alpha_{\text{obs}} &\sim \text{LogNormal}(0,2), \\
x_1 &\sim \mathcal{N}(0,\alpha_{\text{inn}}), \\
x_i &\sim \mathcal{N}(x_{i-1},\alpha_{\text{inn}}), \quad i = 2,\ldots,30, \\
y_i &\sim \mathcal{N}(x_i,\alpha_{\text{obs}}), \quad i = 1,\ldots,30.
\end{aligned}
$$

The inference objective is to estimate α_{inn}, α_{obs}, and the latent states $\{x_i\}_{i=1}^{30}$ based on the available observations $\{y_i\}_{i=1}^{10}$ and $\{y_i\}_{i=20}^{30}$, with the middle observations missing. This missing-data structure increases the difficulty of inference, making it a useful benchmark for probabilistic time series modeling; see Geffner & Domke (2023).

Spatial Statistics (LGCP). The *Log-Gaussian Cox Process* (LGCP) is a widely used spatial model in statistics (Møller et al., 1998), which describes spatially distributed point processes such as the locations of tree saplings. The target density is defined over a discretized spatial grid of size $d = 40 \times 40 = 1600$, and follows:

$$
\rho_{\text{target}} = \mathcal{N}(x;\mu,\Sigma)\prod_{i=1}^{d}\exp\left(x_i y_i - \frac{\exp(x_i)}{d}\right),
$$

where y represents an observed dataset, and μ and Σ define the mean and covariance of the prior distribution. This formulation leads to a complex spatial dependency structure. We focus on the more challenging *unwhitened* variant of the model, which retains the full covariance structure and thus introduces stronger dependencies between grid locations, as described in Heng et al. (2020); Arbel et al. (2021).

B.1.2 Synthetic targets

For these tasks, ground-truth samples are available, allowing for direct evaluation of inference accuracy.

Mixture distributions (GMM and MoS). We consider mixture models where the target distribution consists of m mixture components, defined as:

$$
p_{\text{target}} = \frac{1}{m}\sum_{i=1}^{m} p_i.
$$

The *Gaussian Mixture Model* (GMM), adapted from Blessing et al. (2024), is constructed with $m = 40$ Gaussian components:

$$
\begin{aligned}
p_i &= \mathcal{N}(\mu_i, I), \\
\mu_i &\sim \mathcal{U}_d(-40,40),
\end{aligned}
$$

where $\mathcal{U}_d(l,u)$ denotes a uniform distribution over $[l,u]^d$. We set the dimensionality to $d = 50$ in the experiments in Tab. 2 and to $d = 5$ in Tab. 3.

The *Mixture of Student's t-distributions* (MoS) follows a similar construction but uses Student's t-distributions with two degrees of freedom (t_2) as the mixture components:

$$
\begin{aligned}
p_i &= t_2 + \mu_i, \\
\mu_i &\sim \mathcal{U}_d(-10,10),
\end{aligned}
$$

where μ_i denotes the translation of each component. We set the dimensionality to $d = 50$ in the experiments in Tab. 2 and to $d = 10$ in Tab. 3.

For both the GMM and MoS tasks, the component locations μ_i remain fixed across experiments using a predefined random seed to ensure reproducibility.

Funnel The *Funnel* distribution, originally introduced in Neal (2003), serves as a challenging benchmark due to its highly anisotropic shape. It is defined as:

$$p_{\text{target}}(x) = \mathcal{N}(x_1; 0, \sigma^2)\mathcal{N}(x_2, \ldots, x_{10}; 0, \exp(x_1)I), \tag{18}$$

where $\sigma^2 = 9$ for any number of dimensions $d \geq 2$. In our main experiments, we consider the case $d = 10$. To maintain consistency with prior benchmarks Blessing et al. (2024), we apply a hard constraint by clipping all sampled values to the interval $[-30, 30]$.

C Experimental Details

C.1 Evaluation

In the Bayesian learning task, we compute the average of the ELBO over the previous 10 estimations, each estimated using 2000 samples. For the LGCP task, the evaluation is performed using 300 samples. The ELBO values reported in Tab. 1 represent the best ELBO achieved during training. For synthetic tasks, we additionally compute the Sinkhorn distance 100 times throughout the training process. The ELBO and Sinkhorn distance reported in Tab. 2 correspond to the checkpoint at the end of training. On MoS-40 50D and GMM-40 50D, we use 16000 samples to estimate the sinkhorn distance and use 2000 samples on all other synthetic targets. In Tab. 1 and Tab. 2 we report the average metric value together with the standard error over three seeds.

C.2 Hyperparameter tuning

Benchmarks In benchmark experiments in Sec. 5 we perform for each method a grid search over σ_{diff}, σ_{prior}, the learning rate of the model. The learning rate of the diffusion parameters such as σ_{prior} and σ_{diff} is always chosen to be equal to the model learning rate. On all Bayesian learning tasks, we perform a grid search over $\sigma_{\text{diff,init}} = \{0.1, 0.3\}$, $\sigma_{\text{prior,init}} = \{0.5, 1.0\}$ and the learning rate $\lambda_{\text{model,SDE}} \in \{0.005, 0.002, 0.001\}$. On Brownian and German Credit, we found that if σ_{diff} is not learned a finer grid-search over σ_{diff} is necessary. Therefore on Brownian, we additionally add $\sigma_{\text{diff}} = 0.05$ and on German Credit $\sigma_{\text{diff}} = 0.01$ to the grid search.

On MoS 50D and GMM 50D, we follow Chen et al. (2024) and fix $\sigma_{\text{prior,init}}$ to a high initial value. We found that $\sigma_{\text{prior,init}} = 80$ yielded the best results. We found that smaller learning rates are necessary and we also search over the learning rate of the interpolation parameters between the prior and the target distribution. Therefore we adapt the grid search to $\sigma_{\text{diff,init}} = \{1., 1.5\}$, $\lambda_{\text{itnerpol}} = \{0.01, 0.001\}$ and the learning rate $\lambda_{\text{model,SDE}} \in \{0.0001, 0.00005, 0.00001\}$.

Grid searches are performed over 8000 training iterations on all targets except MoS 50d and GMM 50d, where 20000 training iterations are performed. The best run is chosen according to the best ELBO value at the end of training on Bayesian tasks and on Synthetic targets according to the best Sinkhorn distance. The best hyperparameters are then run for 40000 training iterations. On MoS 50d and GMM 50d in Tab. 2 the best Sinkhorn distance is sometimes achieved at initialization. In this case, the checkpoint is excluded as it has only slightly better Sinkhorn values but much worse ELBOs than the runs at the end of training.

Ablations In ablation experiments in Sec. 5 we iteratively tuned hyperparameters such as $\sigma_{\text{prior,init}}$ and $\sigma_{\text{diff,init}}$ and learning rates for each method. For CMCD-LV and CMCD-LV $\star$ we found it hard to find a good-performing diffusion coefficient. Therefore, we used the learned average diffusion coefficient of CMCD-rKL-LD σ_{diff} as a starting point for iterative hyperparameter tuning which resulted in a decent performance of CMCD-LV and CMCD-LV $\star$.

Exploration Experiments In variational annealing we first approximate the target distribution at $\beta_{\text{start}} = \frac{1}{\mathcal{T}_{\text{start}}}$, where $\mathcal{T}$ is the temperature. In our experiments, we use a linear schedule, where the temperature is decreased linearly from $\mathcal{T}_{\text{start}}$ to 1.

A grid search is performed according to:

GMM-40 (5d) - CMCD-rKL-LD off-policy:

- $\alpha \in \{1.1, 1.3, 1.5, 1.7, 2.0\}$
- $\sigma_{\text{prior,init}} \in \{1, 10, 20, 40, 60\}$
- $\beta_{\max} \in \{0.1, 0.3, 0.5\}$

GMM-40 (5d) - CMCD-rKL-LD annealing:

- $T_{\text{start}} \in \{40, 60, 80, 100\}$
- learning rate $\in \{0.001, 0.005, 0.008\}$
- $\beta_{\max} \in \{0.1, 0.3, 0.5, 1\}$

GMM-40 (5d) - CMCD-rKL-LD tune $\sigma_{\text{prior,init}}$:

- $\sigma_{\text{prior,init}} \in \{10, 20, 40, 60\}$
- learning rate $\in \{0.001, 0.005, 0.008\}$
- $\beta_{\max} \in \{0.1, 0.3, 0.5, 1\}$

MoS-10 (10d) - CMCD-rKL-LD off-policy:

- $\alpha \in \{1.1, 1.5\}$
- $\sigma_{\text{prior,init}} \in \{1, 5, 10, 20\}$

MoS-10 (10d) - CMCD-rKL-LD annealing:

- $T_{\text{start}} \in \{1, 2, 5, 10\}$

MoS-10 (10d) - CMCD-rKL-LD tune $\sigma_{\text{prior,init}}$:

- $\sigma_{\text{prior,init}} \in \{1, 2, 5, 10, 20, 40\}$

C.3 Architecture

Score parametrization We parameterize the score in the following way:

$$u_\theta(X_t) = \text{clip}(\tilde{u}_\theta(X_t, t) + \hat{u}_\theta(x_t, t) \odot \text{clip}(\nabla_{X_t} \log \pi_t, -10^2, 10^2), -10^4, 10^4) \tag{19}$$

where $\tilde{u}_\theta(X_t, t)$ and $\hat{u}_\theta(x_t, t)$ are parameterized with backbone MLPs with two hidden layers and 64 neurons each, where we use skip connections Liu et al. (2021), layer normalization Ba et al. (2016), and Relu activation layers Agarap (2018) with He initialization He et al. (2015). The input of the backbone MLPs is a shared embedding, where X_t is processed by a single activation layer and t is processed by sine and cosine embedding with an overall dimension of 128. The parameter count of this architecture is similar to PISgradnet from (Vargas et al., 2024).

Parametrization of prior distribution: Similarly to Chen et al. (2024) and Blessing et al. (2024) and parameterize the prior distribution π_T in the following way:

$$\pi_T = \mathcal{N}(\mu_\theta, \text{diag}(\exp(l_\theta))$$

where $\mu_\theta \in \mathbb{R}^d$ and logarithmic standard deviations $l_\theta \in \mathbb{R}^d$ are learnable parameters. In contrast to Chen et al. (2024) and Blessing et al. (2024) we do not update μ_θ and l_θ via the reparameterization trick as training progresses but also with the usage of the log derivative trick.

Parametrization of interpolation parameters: For each diffusion time step $t \in \{0, ..., T-1\}$ we parameterize the interpolation parameter $\beta_\theta(t)$ in the following way:

$$\beta_\theta(t) = \frac{\mathrm{softplus}(\theta_t)}{\sum_{t=0}^{T} \mathrm{softplus}(\theta_t)}, \tag{20}$$

where $\theta \in \mathbb{R}^T$ are learnable parameters. Each variable of $\theta \in \mathbb{R}^T$ is initialized to zero.

Parametrization diffusion coefficient: We keep diffusion coefficients constant across time steps and parameterize it as $\sigma_t = \exp \gamma$, where $\gamma = \log \sigma_{\mathrm{init}}$. In principle, one could parameterize it similarly as the interpolation parameters, which would allow for a time-adaptive schedule. However, we leave this up for future work.

Training All parameters are trained with the usage of the RAdam Liu et al. (2020) optimizer. We use gradient clipping by norm at the value of 1. The learning rates are decayed with a cosine learning rate schedule from λ_{start} to $\lambda_{\mathrm{start}}/10$.