

A Appendix

A.1 Implementation Details

Preconditioning and Architecture. Following [61], SCDS uses a network of the form

$$u_\theta(X_t, t, d) = f_\theta^{(1)}(X_t, t, d) + f_\theta^{(2)}(t) \nabla \log \rho(X_t).$$

They demonstrated that incorporating the gradient $\nabla \log \rho$ can mitigate mode collapse and improve overall performance. This design choice is also supported by the experiments of [56, 43, 10].

The base model $f_\theta^{(1)}$ is a multi-layer perceptron that uses Fourier embeddings [53] for the current time t and the step size d . The additional score component $f_\theta^{(2)}$ also employs Fourier embeddings to process the time variable t .

Discretizing the Radon–Nikodym derivative. With $v = 0$, the log-density ratio between the controlled process and its time-reversed uncontrolled reference reduces to

$$\log \frac{d\mathbb{P}^{u, p_{\text{prior}}}}{d\mathbb{P}^{0, p_{\text{target}}}} = \int_0^T u_s \cdot dW_s - \frac{1}{2} \int_0^T \|u_s\|^2 ds + \log \frac{p_{\text{prior}}(X_0)}{\rho(X_T)}.$$

A simple Monte-Carlo estimator of its log-variance (the LV-divergence [5]) is given in Algorithm 5.

Algorithm 5 Monte-Carlo estimator of the LV-divergence

```

1: Input: trained control  $u_\theta$ ; time grid  $\{t_n\}_{n=0}^N$ ; batch size  $m$ 
2: for  $i = 1, \dots, m$  do
3:   Sample  $X_0 \sim p_{\text{prior}}$ 
4:    $\text{rnd}^{(i)} \leftarrow \log p_{\text{prior}}(X_0)$ 
5:   for  $n = 1, \dots, N$  do
6:      $\Delta t \leftarrow t_n - t_{n-1}$ 
7:     Compute the drift  $\mu \leftarrow \mu(X_{t_{n-1}}, t_{n-1})$ 
8:     Compute the diffusion coefficient  $\sigma \leftarrow \sigma(t_{n-1})$ 
9:     Compute the control  $u \leftarrow u_\theta(X_{t_{n-1}}, t_{n-1})$ 
10:    Sample  $Z \sim \mathcal{N}(0, I)$ 
11:     $X_{t_n} \leftarrow X_{t_{n-1}} + (\mu + \sigma u) \Delta t + \sigma \sqrt{\Delta t} Z$ 
12:     $\text{rnd}^{(i)} \leftarrow \text{rnd}^{(i)} + u \cdot Z \sqrt{\Delta t} - \frac{1}{2} \|u\|^2 \Delta t$ 
13:   end for
14:    $\text{rnd}^{(i)} \leftarrow \text{rnd}^{(i)} - \log \rho(X_{t_N})$ 
15: end for
16:  $\bar{r} \leftarrow \frac{1}{m} \sum_{i=1}^m \text{rnd}^{(i)}$ 
17:  $\mathcal{L}_{\text{LV}} \leftarrow \frac{1}{m-1} \sum_{i=1}^m (\text{rnd}^{(i)} - \bar{r})^2$ 
18: return  $\mathcal{L}_{\text{LV}}$ 

```

Lines 1–4 initialize the log-likelihood ratio with the prior term. Lines 5–13 implement Euler–Maruyama simulation of the controlled SDE together with the discrete Girsanov increment $u \cdot Z \sqrt{\Delta t} - \frac{1}{2} \|u\|^2 \Delta t$. Lines 14–18 add the terminal likelihood ratio $-\log \rho(X_{t_N})$ and form the unbiased sample variance.

A.2 Experiment Details

Hyperparameters. All models are trained using the Adam optimizer [28]. We adopt the learning-rate strategy of [10], which is summarized as follows:

- **GMM:** Learning rate 1×10^{-4} for DDS, DIS, SCDS, and CDDS; 1×10^{-3} for PIS
- **Funnel:** Learning rate 1×10^{-3} for all models
- **MW54, MW52:** Learning rate 1×10^{-3}
- **Ionosphere and LGCP:** Learning rate 1×10^{-4}

For the synthetic tasks and the Ionosphere dataset, we use a batch size of 512, while for LGCP we use a batch size of 64. We generate samples using 128 diffusion steps for the synthetic tasks and 256 for the real-world experiments. Although the synthetic tasks typically converge after about 10,000 iterations, we train for up to 60,000 iterations on the real-world tasks. Each experiment is repeated five times, and we report the average results across these trials.

Tasks.

- **Gaussian Mixture Model (GMM):** Following [61, 8], we define the target distribution as

$$\rho(X) = \sum_{m=1}^M \alpha_m \mathcal{N}(X; \mu_m, \Sigma_m).$$

In our setup, we set $M = 9$, $\Sigma_m = 0.3 I$, and $\{\mu_m\}_{m=1}^M$ to be a 3×3 grid in \mathbb{R}^2 with coordinates $\{-5, 0, 5\} \times \{-5, 0, 5\}$.

- **Funnel:** We use the funnel distribution introduced by [36] and follow the methodology of [8]. The target distribution is

$$\rho(X) = \mathcal{N}(X_1; 0, v^2) \prod_{i=2}^d \mathcal{N}(X_i; 0, e^{X_1}),$$

where $d = 10$ and $v = 3$.

- **Many-Well:** We follow [8] and define the many-well distribution by

$$\rho(X) = \exp\left(-\sum_{i=1}^m (X_i^2 - \delta)\right) \exp\left(-\frac{1}{2} \sum_{i=m+1}^d X_i^2\right).$$

For the MW54 target, we use $d = 5$, $m = 5$, and $\delta = 4$. For the MW52 target, we use $d = 50$, $m = 5$, and $\delta = 2$.

- **Log Cox Gaussian Process (LGCP):** As discussed in [61, 12], the LGCP distribution is

$$\rho(X) = \mathcal{N}(X; \mu, \Sigma) \prod_{i=1}^d \exp\left(X_i Y_i - \frac{\exp(X_i)}{d}\right),$$

where Y is a given dataset, and μ, Σ specify a chosen Gaussian prior. We follow [61, 4] for both the dataset construction and prior selection.

- **Ionosphere:** We evaluate a Bayesian logistic regression model where

$$X \sim \mathcal{N}(0, \sigma^2 I), \quad Y^{(i)} \sim \text{Bernoulli}(\text{sigmoid}(X^\top U^{(i)})),$$

using the 35-dimensional Ionosphere dataset $\{U^{(i)}, Y^{(i)}\}_{i=1}^{351}$. This task was introduced in [49], and we follow the implementation details from [10].

Evaluation Metrics.

- **Sinkhorn Distance:** The Sinkhorn distance \mathcal{W}_γ^2 [14] is an entropy-regularized optimal transport metric computed between samples from the model and samples from the target distribution. This requires direct sampling from the target density, so for real-world tasks we rely on other metrics when such samples are unavailable.
- **Effective Sample Size (ESS):** We compute the normalized ESS via

$$\text{ESS} = \frac{\left(\sum_{i=1}^m w^{(i)}\right)^2}{m \sum_{i=1}^m (w^{(i)})^2},$$

where $w^{(i)}$ are importance weights approximated by discretizing the Radon–Nikodym derivative (6), see Algorithm 5.

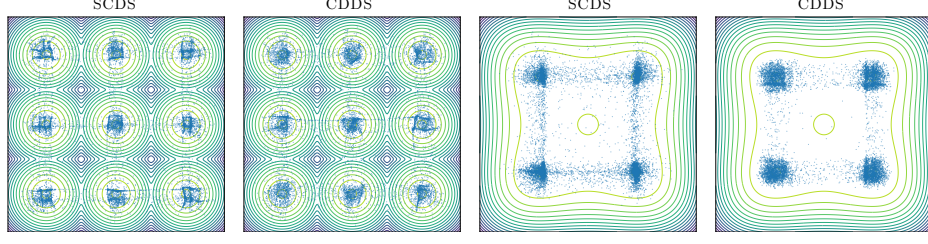


Figure 4: Left single-step sampling for the GMM tasks. Right for the MW54 task.

- **Estimation of the Normalization Constant:** When the ground-truth partition function Z is known, we report $|\Delta \log Z| = |\log Z - \log \hat{Z}|$. We estimate $\log \hat{Z}$ by rearranging (6), giving

$$\log \hat{Z} \approx \log \left(\frac{1}{m} \sum_{i=1}^m w^{(i)} \right).$$

- **Evidence Lower Bound (ELBO):** We follow the extended ELBO estimator

$$\text{ELBO} \approx \frac{1}{m} \sum_{i=1}^m \log w^{(i)},$$

where the weights $w^{(i)}$ again come from (6).

Hardware. All experiments were trained and evaluated on a single NVIDIA RTX A6000 GPU with CUDA version 12.2. We implemented our methods in PyTorch.

A.3 Additional Results

Figure 4 visualizes samples generated in a single step using our proposed methods, SCDS and CDDS, on the GMM and MW54 benchmark tasks. The results demonstrate that both methods successfully recover the modes even when sampling with only a single diffusion step.

Table 3 provides a comparison of single-step sampling methods (CDDS and SCDS) against Sequential Monte Carlo (SMC) [15], where SMC employs 128 steps. The comparison illustrates that both CDDS and SCDS outperform SMC despite using only a single step.

Tables 4 and 5 replicate the results presented in Tables 1 and 2, respectively, but include standard deviations to provide additional statistical context and completeness.

Table 3: Sinkhorn distance \mathcal{W}_γ^2 for SMC with 128 steps against single-step CDDS and SCDS.

Target	SMC	CDDS	SCDS
Funnel	149.31 \pm 2.97	8.88 \pm 0.09	5.49 \pm 0.04
MW54	0.61 \pm 0.15	0.28 \pm 0.00	0.39 \pm 0.00
MW52	46.44 \pm 0.13	6.18 \pm 0.00	7.11 \pm 0.00

A.4 Proofs

Theorem 1. Let $f_\theta(X_t, t)$ be a consistency function parameterized by θ , and let $f(X_t, t; u)$ denote the consistency function of the PF ODE defined by the control u . Assume that f_θ is L -Lipschitz continuous. Additionally, assume that for each step $n \in \{1, 2, \dots, N-1\}$, the ODE solver called at t_n has a local error bounded by $O((t_{n+1} - t_n)^{p+1})$ for some $p \geq 1$. If $\mathcal{L}_{CD}(\theta, \theta'; u) = 0$, then:

$$\sup_{n, X_{t_n}} \|f_\theta(X_{t_n}, t_n) - f(X_{t_n}, t_n; u)\|_2 = O((\Delta t)^p), \quad (12)$$

where $\Delta t := \max_{n \in \{1, 2, \dots, N-1\}} |t_{n+1} - t_n|$.

Table 4: Synthetic-benchmark results (mean (std)). Each task reports the Sinkhorn distance ($\mathcal{W}_\gamma^2 \downarrow$), effective sample size (ESS \uparrow), and absolute log-normalisation error ($|\Delta \log Z| \downarrow$). Shaded cells denote single-step methods (NFE = 1).

	PIS	DDS	DIS	DIS	CDDS (ours)	SCDS (ours)
NFE \downarrow	128	128	128	1	1	1
GMM (2D)						
\mathcal{W}_γ^2	1.795(0.076)	0.090(0.001)	0.020(0.000)	0.056(0.001)	0.031(0.000)	0.048(0.001)
ESS	0.00082(0.00053)	0.00654(0.00101)	0.805(0.001)	0.00057(0.00050)	0.340(0.201)	0.050(0.005)
$ \Delta \log Z $	2.181(0.002)	1.682(0.016)	0.090(0.013)	14.767(0.041)	1.572(0.004)	1.074(0.009)
MW54 (5D)						
\mathcal{W}_γ^2	0.138(0.000)	0.137(0.000)	0.123(0.000)	6.280(0.006)	0.282(0.001)	0.391(0.001)
ESS	0.066(0.016)	0.005(0.006)	0.268(0.012)	0.00002(0.00001)	0.044(0.060)	0.00002(0.00001)
$ \Delta \log Z $	1.997(0.005)	2.415(0.019)	1.206(0.021)	6766.289(17.303)	1.097(0.015)	11.252(0.077)
Funnel (10D)						
\mathcal{W}_γ^2	6.073(0.096)	5.860(0.035)	5.175(0.071)	10.522(0.057)	8.885(0.085)	5.492(0.043)
ESS	0.067(0.058)	0.058(0.034)	0.130(0.037)	0.00005(0.00003)	0.00005(0.00003)	0.00014(0.00014)
$ \Delta \log Z $	0.438(0.014)	0.564(0.008)	0.641(0.020)	$1.28 \times 10^7 (9.20 \times 10^6)$	1.332(0.006)	10.356(0.033)
MW52 (50D)						
\mathcal{W}_γ^2	6.804(0.002)	6.783(0.002)	6.881(0.002)	31.253(0.008)	6.176(0.002)	7.111(0.001)
ESS	0.370(0.031)	0.441(0.007)	0.003(0.002)	0.00001(0.00000)	0.00006(0.00003)	0.00001(0.00000)
$ \Delta \log Z $	42.450(0.008)	42.424(0.013)	39.781(0.031)	9116.071(10.956)	63.724(0.014)	87.709(0.127)

Table 5: ELBO (mean (std)); higher is better. Shaded columns correspond to single-step inference.

Method	Ionosphere (35D)	LGCP (1600D)
PIS (256)	−39.316 (2.378)	397.539 (5.413)
DDS (256)	−1510.319 (25.052)	314.760 (3.676)
DIS (256)	−77.395 (0.657)	365.606 (1.291)
DIS (1)	−3252.689 (40.147)	$-3.09 \times 10^6 (1.02 \times 10^6)$
CDDS (1)	−27.506 (0.018)	1118.019 (0.746)
SCDS (1)	−567.271 (13.572)	−4579.654 (8.424)

878 *Proof.* The proof is similar to the one presented by [52], with the key difference that we must account
879 for the global integration error introduced by the ODE solver.

880 If the ODE solver, when called at t_{n+1} , has a local error uniformly bounded by $O((t_n - t_{n-1})^{p+1})$,
881 then the cumulative error across all steps is approximately the sum of $n + 1$ local errors and is
882 bounded by $O((\Delta t)^p)$.

883 We are interested in e_n , the error between the learned consistency function and the consistency
884 function of the PF ODE defined by the control u at $X_{t_n} \sim p_{t_n}(X_{t_n})$,

$$e_n := f_\theta(X_{t_n}, t_n) - f(X_{t_n}, t_n; u).$$

885 If $\mathcal{L}(\theta, \theta; u) = 0$, we deduce that

$$\lambda(t_n)d(f_\theta(\hat{X}_{t_{n+1}}, t_{n+1}), f_\theta(\hat{X}_{t_n}, t_n)) = 0.$$

886 Since $\lambda(t_n) > 0$, this implies:

$$f_\theta(\hat{X}_{t_{n+1}}, t_{n+1}) = f_\theta(\hat{X}_{t_n}, t_n). \quad (13)$$

887 We can derive a recurrence relation for e_n :

$$\begin{aligned}
e_n &\stackrel{(i)}{=} f_\theta(X_{t_n}, t_n) - f_\theta(\hat{X}_{t_n}, t_n) + f_\theta(\hat{X}_{t_n}, t_n) - f(X_{t_{n+1}}, t_{n+1}; u) \\
&\stackrel{(ii)}{=} f_\theta(X_{t_n}, t_n) - f_\theta(\hat{X}_{t_n}, t_n) + f_\theta(\hat{X}_{t_{n+1}}, t_{n+1}) - f(X_{t_{n+1}}, t_{n+1}; u) \\
&= f_\theta(X_{t_n}, t_n) - f_\theta(\hat{X}_{t_n}, t_n) + f_\theta(\hat{X}_{t_{n+1}}, t_{n+1}) - f_\theta(X_{t_{n+1}}, t_{n+1}) \\
&\quad + f_\theta(X_{t_{n+1}}, t_{n+1}) - f(X_{t_{n+1}}, t_{n+1}; u) \\
&= f_\theta(X_{t_n}, t_n) - f_\theta(\hat{X}_{t_n}, t_n) + f_\theta(\hat{X}_{t_{n+1}}, t_{n+1}) - f_\theta(X_{t_{n+1}}, t_{n+1}) + e_{n+1} \\
&\dots \\
&\stackrel{(iii)}{=} f_\theta(X_{t_n}, t_n) - f_\theta(\hat{X}_{t_n}, t_n) + f_\theta(X_T, T) - f_\theta(\hat{X}_T, T) + e_T.
\end{aligned}$$

888 Here, step (i) follows from the definition of the consistency function, step (ii) is due to Eq. (13), and
889 step (iii) leverages the telescoping nature of the sum.

890 Furthermore, since f_θ is parameterized such that $f_\theta(X_T, T) = X_T$, we have

$$e_T = f_\theta(X_T, T) - f(X_T, T; u) = X_T - X_T = 0.$$

891 Finally, given that f_θ is Lipschitz and considering the bound on the global error of the ODE solver:

$$\|e_n\|_2 \leq \|e_T\|_2 + L\|X_{t_n} - \hat{X}_{t_n}\|_2 + L\|X_T - \hat{X}_T\|_2 = O((\Delta t)^p).$$

892

□

893 A.5 Limitations

894 The primary limitation of CDDS is its reliance on a pre-trained diffusion sampler. In contrast,
895 SCDS overcomes this limitation by simultaneously learning both the sampling process and self-
896 distillation within a single training phase using a single model. However, SCDS incurs a slightly
897 higher computational cost during training compared to methods like PIS, DDS, and DIS, requiring an
898 additional three network function evaluations per training iteration.

899 Nevertheless, compared to conventional diffusion-based samplers, which often demand hundreds of
900 function evaluations per iteration during training, the additional computational overhead introduced
901 by SCDS is minimal. Section 6 provides a detailed analysis of combined training and sampling costs,
902 demonstrating that the training overhead associated with SCDS is modest and quickly amortized in
903 realistic scenarios.

904 A.6 Broader Impact

905 This paper introduces consistent diffusion samplers, a new class of samplers designed to generate
906 high-fidelity samples in a single step. It offers significant promise for reducing the iterative steps to
907 produce high-quality samples. Its broader impact will likely be positive, fostering innovation and
908 new applications of diffusion samplers.