Ex$^2$MCMC: Sampling through Exploration Exploitation

Evgeny Lagutin$^{1,2,5}$ Danil Selikhanovych$^{1,5}$ Achille Thin$^4$ Sergey Samsonov$^2$
Alexey Naumov$^2$ Denis Belomestny$^{2,3}$ Maxim Panov$^1$ Eric Moulines$^{2,4}$

$^1$ Skoltech $^2$ HSE University $^3$ Duisburg-Essen University
$^4$ Ecole Polytechnique $^5$ Moscow Institute of Physics and Technology

Abstract

We develop an Explore-Exploit Markov chain Monte Carlo algorithm (Ex$^2$MCMC) that combines multiple global proposals and local moves. The proposed method is massively parallelizable and extremely computationally efficient. We prove $V$-uniform geometric ergodicity of Ex$^2$MCMC under realistic conditions, and compute explicit bounds on the mixing rate showing the improvement brought by the multiple global moves. We show that Ex$^2$MCMC allows fine-tuning of exploitation (local moves) and exploration (global moves) via a novel approach to proposing dependent global moves. Finally, we develop an adaptive scheme, FlEx$^2$MCMC, that learns the distribution of global moves using normalizing flows. We illustrate the efficiency of Ex$^2$MCMC and its adaptive versions on many classical sampling benchmarks. We also show that these algorithms improve the quality of sampling GANs as energy-based models.

1 Introduction

Suppose one is interested in sampling from a probability distribution $\Pi$ that is known up to a scaling factor. A Markov chain Monte Carlo algorithm (MCMC) consists of simulating a realization of a time-homogeneous Markov chain $\{Y_k, k \in \mathbb{N}\}$ with the Markov kernel $K$, with the property that the distribution of $Y_n$ becomes arbitrarily close to $\Pi$ as $n \to \infty$, irrespective of the distribution of $Y_0$. A property that the kernel $K$ must satisfy is that it leaves the distribution $\Pi$ invariant, i.e., $\Pi$ should be a fixed point of the Markov kernel. Instead, one can consider the stronger detailed balance condition or reversibility, a property that is easier to handle due to its local character. In particular, it leads to the famous Metropolis-Hastings (MH) kernel, the cornerstone of MCMC simulations, and a number of its successful variants.

To improve the available samplers, a number of authors have tried to optimize the usual MH algorithm by generating a pool of proposals at each iteration, e.g., Multiple-Try Metropolis algorithm (MTM; Liu et al. (2000); Craiu and Lemieux (2007)). The use of multiple proposals at each iteration, which can be efficiently implemented in parallel computing architectures, allows to increase the local search region without decreasing the acceptance ratio, which leads to an improvement in the mixing rate. This property has been theoretically supported by results on the high-dimensional scaling limit (see Bédard et al. (2012)). At the same time, MCMC algorithms based on multiple independent proposals suffer from the fact that their acceptance rate decreases dramatically in large dimension. In this work, we follow the idea of generating multiple proposals for MH-based MCMC but give a new prospective on it and provide a computationally attractive alternative to MTM.

Contributions  The main contributions of the paper are as follows:

- We propose an Explore-Exploit MCMC algorithm (Ex$^2$MCMC), that retains most of the desirable properties of MTM, in particular, high degree of parallelization and improved mixing rate, while reducing the computational cost. We prove $V$-uniform geometric convergence of Ex$^2$MCMC and evaluate its mixing rate;
- We propose an original method to construct dependent proposals for Ex$^2$MCMC, which allows

Copyright 2022 by the author(s).
to fine-tune the exploration-exploitation trade-off. Moreover, we propose an adaptive algorithm to learn the proposal distribution (FIE\textsuperscript{x}MCMC);

- We provide numerical evaluation of Ex\textsuperscript{x}MCMC and FIE\textsuperscript{x}MCMC on various sampling problems, including sampling from GANs as energy-based model. The results clearly show the benefits of the proposed approaches compared to standard MCMC methods.

2 Ex\textsuperscript{x}MCMC

2.1 From Importance Sampling to Sampling Importance Resampling

Importance Sampling (IS) is widely used for estimating integrals of the function $f$ w.r.t. of a target distribution $\Pi$ on a state-space $(X, \mathcal{X})$, known up to a normalizing factor $Z_\Pi$, $\Pi(dx) = \hat{\Pi}(dx)/Z_\Pi$; see, e.g. (Robert and Casella, 2013). IS consists of weighting samples from a proposal $\Lambda$. Assume that $\Pi(dx) = \hat{\Pi}(x)\Lambda(dx)$, and that the importance weight function $\hat{w}$ is positive, i.e. $\hat{w}(x) > 0$ for all $x \in \mathbb{X}$. We often assume that $\hat{\Pi}$ (hence, $\Pi$) and $\Lambda$ have positive densities w.r.t. a common dominating measure, denoted by $\hat{\pi}$, $\pi$, and $\lambda$, respectively. If this is the case, $\hat{w}(x) = \hat{\pi}(x)/\lambda(x)$. The self-normalized importance sampling (SNIS) estimator of $\pi(f)$ is then given by $\hat{\Pi}_N(f) = \sum_{i=1}^N \hat{\omega}_N f(X_i)$, where $X_1^N \sim \Lambda$ and $\hat{\omega}_N = \hat{w}(X_i)/\sum_{j=1}^N \hat{w}(X_j)$, $i \in \{1, \ldots, N\}$ are the self-normalized importance weights.

Provided that $\Lambda(\omega^2) = \int \hat{w}(x)^2\Lambda(dx) < \infty$, the bias and the mean square error MSE of the SNIS estimator are inversely proportional to $N$.

Although importance sampling is primarily intended to approximate integrals of the form $\Pi(f)$, it can also be used to (approximately) sample from $\Pi$. The latter can be achieved by the Sampling Importance Resampling (SIR, Rubin, 1987). SIR is a two-stage procedure. In the first stage, an i.i.d. sample $X_1^N = X_1, \ldots, X_N$ is sampled from $\Lambda$ and the importance weights $\omega_1^N = \omega_1^{(1)}$ are computed. In a second step, a sample of size $M$, $Y_1^M$ is obtained by sampling with replacement with the weights $\omega_1^N$, denoted $\text{Cat}(\omega_1^N)$. In other words, given a $N$-iid sample $X_1^N$ from $\Lambda$, SIR draws samples from the empirical distribution $\hat{\Pi}(dx) = \sum_{i=1}^N \omega_i^N \delta_{X_i}(dx)$ where $\delta_{X_i}(dx)$ denotes the Dirac mass at $y$. As $N \to \infty$, a sample $Y_1^M \sim \Pi$ will be distributed according to $\Pi$; see (Smith and Gelfand, 1992; Skare et al., 2003). The main issue of SIR method is that it is only asymptotically valid.

2.2 From SIR to iterated Sampling Importance Resampling (i-SIR)

A closely related algorithm is the iterated SIR (i-SIR), a term coined in (Andrieu et al., 2010) and later investigated more deeply in (Andrieu et al., 2018). For i-SIR, the sample size $N$ is not necessarily large, but the process of sampling from the proposal, computing the normalized importance weights and picking a candidate is iterated. The $j$-th i-SIR iteration is defined as follows. Given the current state $Y_j \in \mathbb{X}$, (i) Set $X_{j+1}^1 = Y_j$ and draw $X_{j+1}^{2:N}$ independently from the proposal distribution $\Lambda$. (ii) Compute the normalized importance weights $\hat{\omega}_{N,j+1} = \hat{w}(X_{j+1}^i)/\sum_{i=1}^N \hat{w}(X_{j+1}^i)$, $i \in \{1, \ldots, N\}$. (iii) Draw $Y_{j+1}$ from the proposal set $X_{j+1}^{1:N}$, choosing $X_{j+1}^i$ with probability $\omega_{N,j+1}$. The Markov kernel associated with i-SIR is given, for $(x, A) \in \mathbb{X} \times \mathcal{X}$, by

$$P_N(x, A) = \int \delta_x(dx) \frac{\sum_{i=1}^N \hat{w}(x_i)}{\sum_{j=1}^N \hat{w}(x_j)} 1_A(x) \prod_{j=2}^N \Lambda(dx_j).$$

\textbf{Lemma 1.} $P_N$ admits $\Pi$ as its invariant distribution.

This result is proven in (Andrieu et al., 2018) (see also Appendix B.1). To go further, we now establish the V-geometric convergence for i-SIR samples to $\Pi$. To state the results, we introduce, for a function $V(x) : \mathbb{X} \to [1, \infty)$, the V-norm of two probability measures $\mu$ and $\nu$ on $(\mathbb{X}, \mathcal{X})$, $\|\mu - \nu\|_V := \sup_{f \in V(x)} |\int f d\mu - \int f d\nu|$. If $V \equiv 1$, $\|\cdot\|_1$ corresponds to the total variation distance (rather denoted $\|\cdot\|_{TV}$.)

\textbf{Definition 2 (Geometric Ergodicity).} A Markov kernel $Q$ with invariant probability measure $\Pi$ is V-geometrically ergodic if there exist constants $\rho \in (0, 1)$ and $M < \infty$ such that, for all $k \in \mathbb{N}$,

$$\|Q^k(x, \cdot) - \Pi\|_V \leq M V(x)\rho^k \quad \text{for all } x \in \mathbb{X}. \quad (2)$$

In particular, geometric ergodicity results ensure that the distribution of the $n$-th step of a Markov chain converges geometrically fast to the invariant probability in V-norm, for all starting points $x \in \mathbb{X}$. Here the dependence on the initial state $x$ appears on the right-hand side only in $V(x)$. Uniform geometric ergodicity of i-SIR is established in (Andrieu et al., 2018) (Theorem 1) under the assumption that the normalized importance weight function $w$ (that is, $\Pi(dx) = w(x)\Lambda(dx)$) is uniformly bounded.

This result is extended below for arbitrary V-norm satisfying $\Pi(V) = \int V(x)\Pi(dx) < \infty$ under the following condition.

\textbf{A1.} For any $x \in \mathbb{X}$, $w(x) \leq L$, with $L < \infty$. The result is up to our best knowledge new.

\textbf{Theorem 3.} Assume $A$. Set $\epsilon_N = \frac{N-1}{2L + N-2}$ and $\kappa_N = 1 - \epsilon_N$. Then,

(i) For any $x \in \mathbb{X}$ and $k \in \mathbb{N}$, $\|P_N^k(x, \cdot) - \Pi\|_{TV} \leq \kappa_N^k$.
by very many MCMC kernels, typically under super-exponential tail conditions for the target distribution; see (Roberts and Rosenthal, 2004) and (Douc et al., 2018) Chapter 2) and the references therein. (iii) states that the importance weights are upper bounded on level sets. This is a mild condition: if $X = \mathbb{R}^d$, and $V$ is norm-like, then the level sets $V_d$ are compact and $\tilde{w}$ is bounded as soon as $\pi$ and $\lambda$ are positive and continuous. We emphasize that we do not need to identify the small sets of the rejuvenation kernel. We can now present the main theoretical result of this paper.

Theorem 4. Assume A[2]. Then, for all $x \in \mathbb{X}$ and $k \in \mathbb{N}$,

$$\|K_N^k(x, \cdot) - \Pi V \|_V \leq c_{N,R}(\Pi(V) + V(x)) \tilde{k}_N^k,$$

where the constant $c_{N,R}$, $\tilde{k}_N$ are given in \eqref{eq:norm-conditions}.

The proof is postponed to Appendix B.3. The main steps are (i) establishing that the level sets $V_d$ are small for the Markov kernel $K_N$ (with a constant which depends only on $N$ and $w_{\infty,d}$); see Lemma 7; (ii) checking that $K_N$ also satisfies a Foster-Lyapunov function $V$ with constant depending only on $\lambda_{R,d} \in [0,1]$, $b_R$, $d$, and $N$; see Lemma 9. Note that $\tilde{k}_N$ typically decreases when the number of proposals $N$ grows. In many situations, the mixing rate $\tilde{k}_N$ is significantly better than the corresponding mixing rate of $K_N$, provided that $N$ is large enough. This is illustrated in Appendix C with the Metropolis Adjusted Langevin Algorithm (MALA) kernel (see e.g. (Besag 1994, Roberts and Tweedie 1996)).

2.3 Dependent proposals for i-SIR and Ex2MCMC algorithms

We now extend Ex2MCMC by relaxing the assumptions that the proposals at each stage are independent. The possibility of using dependent particles (and in particular local moves around the conditioning point) allows Ex2MCMC algorithm to fine-tune Exploration v.s. Exploitation. Denote by $A_N(dx^{1:N})$ the joint distributions of the proposed particles. The key property to satisfy is that the marginal distribution of the proposed particles is $A$, that is for each $i \in [N] = \{1, \ldots, N\}$ there exists a Markov kernel denoted $Q_i$, satisfying

$$A(dx^i)Q_i(x^i, dx^{1:N}(\cdot))/A_N(dx^{1:N}) = \Lambda(dx^{1:N}).$$

The Markov kernel $Q_i(x^i, \cdot)$ defines the conditional distribution of the particles $X^{1:N}(\cdot)$ given the “conditioning” particle $X^i$. In the simple i-SIR case, $Q_i(x^i, dx^{1:N}(\cdot)) = \prod_{j \neq i} \Lambda(dx^j)$ so that $A_N(dx^{1:N}) = \prod_{i=1}^N \Lambda(dx^i)$. Using these conditional distributions, we now adapt Algorithm 5 by changing line 1 by

1. Draw $U_{j+1}$ from the uniform distribution in $[N]$ and set $X^{(j)}_{j+1} = Y_j$.

**Algorithm 1: Single stage of Ex2MCMC algorithm with independent proposals**

| Procedure | Ex2MCMC ($Y_j, \Lambda, R$): |
|-----------|-------------------------------|
| Input     | Previous sample $Y_j$; proposal distribution $\Lambda$; rejuvenation kernel $R$; |
| Output    | New sample $Y_{j+1}$; |
|           | Set $X_{j+1} \sim Y_j$; draw $X^{2:N}_{j+1} \sim \Lambda$; |
|           | for $i \in [N]$ do |
|           | compute the normalized weights |
|           | $\omega_{i,j+1} = \frac{\tilde{w}(X_{j+1}^{i})}{\sum_{k=1}^N \tilde{w}(X_{j+1}^{k})}$; |
|           | Set $I_{j+1} = \text{Cat}(\omega_{1,j+1}, \ldots, \omega_{N,j+1})$; |
|           | Draw $Y_{j+1} \sim R(X_{j+1}^{I_{j+1}}, \cdot)$. |
We now describe a general method for constructing 

density, which is a key to efficient implementation. The 

result establishes the validity of the proposed algorithm, 

Theorem 5. For any $N \geq 2$, the Markov kernel $C_N$ 

admits $\Pi$ as its unique invariant distribution.

We now describe a general method for constructing joint proposals $\Lambda_N$ with the proper marginal $\Lambda$. The 

main motivation for this construction is to seize the 
opportunity of massive parallelization of proposal 
sampling, which is a key to efficient implementation. The 

basic idea is to introduce a hierarchical latent variable 
model that allows flexible control of the dependency 
between proposals while preserving the desired sym- 

metry property. Denote by $\Xi$ the latent distribution and $C_N$ the corresponding Markov kernel. The following re-

sult establishes the validity of the proposed algorithm, 

which holds virtually without any assumption.

\[
\Xi(d\xi)R(\xi, dx) = \Lambda(dx)R(x, d\xi). 
\]  

(8)

The Markov kernel $\tilde{R}$ can be computed using the Bayes 

rule. This is transparent when the Markov kernel $R$ has density, i.e. there exists a function $r$ and a measure $\mu$ 
on $\mathbb{X}$ such that $R(\xi, dx) = r(x | \xi)\mu(dx)$. In this case 

$r$ is the conditional p.d.f. of the proposal $X$ given the 

latent variable $\xi$. In such case, $\Lambda(dx) = \lambda(x)\mu(dx)$, 

with $\lambda(x) = \int_\mathbb{X} \Xi(d\xi)/r(x | \xi)\Xi(d\xi)\Lambda(dx)$. This shows that the reverse 

kernel has a density w.r.t. $\Xi$ given by $\tilde{r}(\xi | x) = r(x | \xi)/\lambda(x)$. Using (8), it is easily seen that for all $i \in [N]$

\[
Q_i(x^i, dx^{1:N\setminus\{i\}}) = 
\int_{\mathbb{X}} \tilde{R}(x^i, d\xi^i)S_i(\xi^i, d\xi^{1:N\setminus\{i\}}) \prod_{j \neq i} R(\xi^j, dx^j). 
\]  

(9)

We can then simply update Algorithm 3 by replacing line 1 by the following three steps:

1. Draw $U_{j+1}$ from the uniform distribution in $[N]$ and set $X_{j+1} = Y_{j+1} \sim S_{j+1}(\xi_{j+1}^{1:N\setminus\{j+1\}})$.
2. Draw latents $\xi_{j+1}^{1:N\setminus\{j+1\}} \sim U_{j+1}^{1:N\setminus\{j+1\}}$.
3. Draw proposals $X_{j+1} \sim R(\xi_{j+1}^{1:N\setminus\{j+1\}})$.

Below we provide an example of Ex$^2$MCMC algorithm with dependent proposal distributions.
2.4 Dependent Gaussian proposals

Let us denote by $g(x; \mu, \Sigma)$ the Gaussian p.d.f. with mean $\mu \in \mathbb{R}^d$ and covariance $\Sigma \in \mathbb{R}^{d \times d}$. Assume that the proposal distribution is Gaussian, i.e., $\lambda(x) = g(x; 0, \Sigma_2^2 I_d)$. As we will see below, this example is important because it is often used when sampling generative models. We give a step-by-step construction that closely follows the presentation given above.

We first define the latent space $\mathbb{E} = \mathbb{R}^d \times \mathbb{R}$, the latent variable $\xi = (\eta, \alpha)$. We assume that the latent distribution is a product $\mathbb{E} = \Lambda \times \nu$, with $\nu(\alpha) = \delta_{a}(\alpha) + (1 - \epsilon) \delta_{0}(\alpha)$, with $\epsilon \in [0, 1]$ and $a \in [0, 1]$. Expressed with random variables, $X = \alpha \eta + \sigma(\alpha) W$, $\sigma(\alpha) = \alpha \lambda(1 - \alpha^2)^{1/2}$ where $(\alpha, W)$ are two independent random variables, $\alpha$ takes two values $a$ and $0$ with probability $\nu(\alpha = a) = \epsilon$, $W \sim N(0, \sigma_2^2 I_d)$. Under these assumptions, the conditional p.d.f of the proposal sample $x$ given the latent variable $(\eta, \alpha)$ is given by $r(x | (\eta, \alpha)) = g(x; \alpha \eta, \sigma^2(\alpha) I_d)$. The reverse kernel $\tilde{R}$ defined by (9) can be written as $\tilde{R}(x, d\alpha, \eta) = \epsilon g(\eta; a x, \sigma^2(\alpha) I_d) \delta_0(\alpha) d\eta + (1 - \epsilon) g(\eta; 0, \sigma_2^2 I_d) \delta_0(\alpha)$. We now specify the joint law of the latent variables to be, for $i \in \mathbb{N}$,

$$S_i(\xi^i, \delta^{1:N \backslash \{i\}}) = \prod_{j \neq i} \delta_{\eta^j}(d\eta^j) \nu(d\alpha^j),$$

(10)

that is, the latent variables $\eta^i, i \in \mathbb{N}$ are all equal, whereas that the random variables $\alpha^{1:N \backslash \{i\}}$ are conditionally independent with the same distributions $\nu$. The graphical model for the resulting latent variables approach for dependent proposals generation is given in Figure 2. Now we can specify the modification of Algorithm 5 in this case by substituting its line 1 by Algorithm 6.

The value $\epsilon$ controls the exploration-exploitation ratio of Ex^2MCMC. When $\epsilon = 0$ we recover the independent i-SIR, and for $\epsilon = 1$ i-SIR is bound only to local proposals. In our experiments, we find in practice that a value $\epsilon \in (0, 1)$ is relevant. To illustrate the need for introducing dependencies between proposal sions, consider the toy problem of sampling a high-dimensional Gaussian distribution: $\pi(x) = g(x; 0, I_d)$ and $\lambda(x) = g(x; 0, 2I_d)$. This is an artificially broken problem, but it provides a diagnosis of why i-SIR fails in high dimensions and why Ex^2MCMC (here with $\epsilon = 1$ and without rejuvenation steps at all) works well (see Figure 1). Details of the experiment can be found in Supplementary Material, Appendix F.4.

2.5 Related Work

i-SIR has been proposed by Andrieu et al. (2010) and further developed in (Andrieu et al. 2018); see also Lee et al. (2010) and Lee (2011). Andrieu et al. (2018) highlights the links of i-SIR with particle Gibbs methods, the main difference being that the proposal distribution is defined on the “path space” used in sequential Monte Carlo methods; see also Doucet et al. (2015). Using this analogy, the rejuvenation kernel plays a role similar to the backward sampling kernel in the particle Gibbs with Forward Backward Sampling (PGFS, Lindsten and Schön 2013). The idea of making the moves dependent of the conditioning particle has been suggested in the PGBS context by Shestopaloff and Neal (2018).

Ex^2MCMC algorithm can also be seen as a collapsed version of the Gibbs sampler proposed in (Jelmeland 2004). The algorithm also has similarities with the Multiple Tries Metropolis (MTM) algorithm, but the Ex^2MCMC is both computationally simpler and displays more favorable mixing properties. In the MTM algorithm, $N$ i.i.d. trial proposals $\{X^{i}_j\}_{i=1}^{N}$ are drawn from a kernel $T(y, \cdot)$ in each iteration: this is similar to Ex^2MCMC sampling step, except that we do not require $Q_j(x^2, dx^1:\Lambda\backslash\{i\}) = \prod_{k=1}^{N} T(x^2, dx^2)$. In a second step, a sample $Y^{*}_{j+1}$ is selected with probability proportional to the weights (the exact expression of the importance weights differs from ours, but this does not change the complexity of the algorithm). In a third step (see Liu et al. (2000) section 2), $N - 1$ i.i.d. proposals are drawn from the kernel $T(Y^{*}_{j+1}, \cdot)$ and the move is assumed to be $Y^{j+1} = Y^{*}_{j+1}$ with a generalized $M$-$H$ ratio, see Liu et al. (2000) eq. 3. This step is bypassed in Ex^2MCMC, which reduces the computational complexity by a factor of 2.

3 Adaptive Ex^2MCMC algorithm

As mentioned earlier, the success of IS methods lies in an appropriate choice of proposal distribution. A classical approach is to define families of proposal distributions $\{\lambda_{\theta}\}$ parameterized by some parameters $\theta$ chosen to match the target distribution $\pi$. Such families can be obtained using a sequence of invertible transformations called normalizing flow (Papamakarios et al. 2021). Let $T$: $\mathbb{X} \rightarrow \mathbb{X}$ be a $C^1$ diffeomorphism. We denote the push-forward of measure $\Lambda$ under $T$, that is, the distribution of $Y = T(X)$ with $X \sim \lambda$, by $T\#\Lambda$. 

---

**Algorithm 2: Ex^2MCMC for Gaussian proposal**

**Input**: $Y^j$ from previous iteration  
**Output**: Set of proposals for the current iteration $X_{j+1}$

1. Draw $U_{j+1} \sim \text{Unif}(\{N\})$ and set $X_{j+1} = Y^j$
2. Draw $\alpha_{j+1} \sim \nu$ and $\eta_{j+1} \sim N(\alpha_{j+1} X_{j+1}, \sigma^2(\alpha_{j+1}) I_d)$
3. For $i \in \{N\} \setminus \{U_{j+1}\}$, draw $W_{i_{j+1}} \sim N(0, I_d)$, $\alpha_{i_{j+1}} \sim \nu$, and set $X^j_{j+1} = \alpha_{i_{j+1}} \eta_{j+1} + \sigma(\alpha_{j+1}) W_{j+1}$. 

---
Algorithm 3: Single stage of FlEx2MCMC.
Steps of Ex²MCMC are done in parallel with common values of proposal parameters \( \theta_j \).
Step 4 updates the parameters using the gradient estimate obtained from all the chains.

| Input | Output: new weights \( \theta_j = Y_j[1 : K] \) |
|-------|-------------------------------------------------|
| \( Y_j[k] = \text{Ex}^2 \text{MCMC}(Y_j[k], T_\theta, \# \Lambda, R) \) | \( \lambda \) |
| Draw \( Z[1 : K] \sim \Lambda \) | \( \theta_j \) |
| Update \( \theta_j = -\gamma \nabla L(Y_j[k], Z, \theta_j) \) | \( \lambda \) |

The corresponding push-forward density is given by \( \lambda_T(y) = \lambda(T^{-1}(y)) J_T(y) \), where \( J_T \) denotes the Jacobian determinant of \( T \); see (Rezende and Mohamed, 2015) and references therein. The parameterized family of diffeomorphisms \( \{ T_\theta \} \) defines a family of distributions \( \{ \lambda_T \} \), denoted for conciseness \( \lambda \). The parameter \( \theta \) is then chosen to minimize a discrepancy between the target \( \pi \) and \( \lambda \). This discrepancy can be e.g. a forward or backward divergence KL or another \( f \)-divergence; see e.g. (Papamakarios et al., 2021). The forward KL objective and its gradient are given by

\[
L^f(\theta) = \int \log \frac{\pi(x)}{\lambda_\theta(x)} \pi(x) dx, \quad (11)
\]

\[
\nabla L^f(\theta) = - \int \nabla \log \lambda_\theta(x) \pi(x) dx. \quad (12)
\]

The backward KL divergence and its gradient are given by

\[
L^b(\theta) = \int \log \frac{\lambda(T_\theta(x)) J_{T_\theta}(x)}{\pi(T_\theta(x))} \lambda(x) dx, \quad (13)
\]

\[
\nabla L^b(\theta) = - \int \nabla \log (\pi(T_\theta(x)) J_{T_\theta}(x)) \lambda(x) dx. \quad (14)
\]

Note that \( \nabla L^b(\theta) \) does not depend on the normalizing constant of \( \pi \). Thus we can compute unbiased estimates of \( L^f(\theta) \) and \( L^b(\theta) \), given a sample \( Y[1] \sim \pi \) and \( Z[1] \sim \lambda \) for \( k \in [K] \), by

\[
\hat{\nabla L^f}(Y[1 : K], \theta) = - \frac{1}{K} \sum_{k=1}^K \nabla \log \lambda_\theta(Y[k]), \quad (15)
\]

\[
\hat{\nabla L^b}(Z[1 : K], \theta) = - \frac{1}{K} \sum_{k=1}^K \nabla \log (\pi(T_\theta(Z[k])) J_{T_\theta}(Z[k])) \lambda(Z[k]). \quad (16)
\]

We introduce in the following a novel adaptive MCMC scheme, FlEx²MCMC, which combines normalizing flows and Ex²MCMC; see (Andrieu and Thoms, 2008; Liang et al., 2011) for a background on adaptive MCMC. The importance weights for FlEx²MCMC become \( \hat{w}_\theta(x) = \hat{\pi}(x)/\lambda_\theta(x) \). The dependence between proposals can be introduced by first correlating samples with kernels \( \{ Q_i \}_{i=1}^\infty \) which satisfy (4), and passing them through the flow \( T_\theta \) afterwards in order to enhance the initial proposal distribution \( \lambda \). The \( j \)-th step of the algorithm is given in Algorithm 7. We essentially perform \( K \) independent Ex²MCMC steps with the same values of flow parameters \( \theta \) and then update parameters based on the gradient estimate obtained from all the chains.

4 Experiments

In this section we illustrate the efficiency of Ex²MCMC and FlEx²MCMC compared to standard MCMC methods. In all our methods we use the MALA rejuvenation kernel, but it is also possible to use other kernels, e.g. HMC (Neal, 2011) or NUTS (Hoffman et al., 2014). For FlEx²MCMC proposals we use the RealNVP (Dinh et al., 2017) normalizing flows trained with the Adam optimizer (Kingma and Ba, 2015). Additional details on the experimental setup, including hyperparameters can be found in Supplementary material, Appendix F and Table 6. We assess sampling performance using the empirical Sliced Total Variation distance (STV, Kolouri et al., 2019), Effective Sample Size (ESS, Kish, 1965) and the empirical Euclidean Earth Mover’s distance (EMD, Monge, 1781). These metrics are defined in Supplementary Material, Appendix F.1. The code reproducing the experiments can be found at https://github.com/stat-ml/ex2mcmc

4.1 Sampling experiments

Distributions with complex geometry We study the ability of Ex²MCMC to sample from the banana-shaped distributions and funnel distributions in high dimensions. Analytical expressions for target distributions and experimental details can be found in Appendix F.5. We use the HMC-based NUTS sampler (Hoffman et al., 2014) to generate reference samples and compute STV, ESS and EMD for samples obtained from single run of MALA, Ex²MCMC and FlEx²MCMC. The numerical results for asymmetric banana distribution are shown on Figure 3 and for funnel distribution on Figure 10. We clearly observe the benefits of Ex²MCMC over MALA in approximation quality (see also Figure 4), while FlEx²MCMC allows to dramatically improve sampling efficiency.

Allen-Cahn equation We use FlEx²MCMC to sample from the invariant distribution of the Allen-Cahn stochastic differential equation; see (Allen and
Bayesian Logistic Regression

Consider the training set $\mathcal{D} = \{(x_j, y_j)\}_{j=1}^M$ consisting of pairs $(x_j, y_j)$, where $x_j = (x_j^{(0)}, \ldots, x_j^{(d-1)}) \in \mathbb{R}^d$ and labels $y_j \in \{-1, 1\}$. Without loss of generality, we assume that $x_j^{(0)} = 1$. The likelihood for a pair $(x, y)$ is given by $p(y \mid x, \theta) = \logit(y(x, \theta))$, $\theta \in \mathbb{R}^d$. Given a prior distribution $p_0(\theta)$, we sample the posterior distribution $p(\theta \mid \mathcal{D})$ and compute the posterior predictive distribution $p(y \mid x, \mathcal{D})$. We approximate $p(y \mid x, \mathcal{D})$ as $p(y \mid x, \mathcal{D}) \simeq \frac{1}{n} \sum_{i=1}^{n} p(y \mid x, \mathcal{D}, \theta_i)$ for $\theta_i \sim p(\cdot \mid \mathcal{D})$, $i = 1, \ldots, n$. We display boxplots of the posterior predictive distribution averaged over the dataset based on 30 independent runs of the samplers, see Figure 6. The results show that Ex2MCMC achieves much higher values compared to i-SIR and MALA while FiEx2MCMC allows to further improve the average values and also simultaneously decrease the variance. The datasets and the implementation details are given in Supplementary Material, Appendix F.6.

4.2 Sampling from GAN as Energy-based model (EBM)

Generative adversarial networks (GANs) represent a class of generative models defined by a pair of a generator network $G$ and a discriminator network $D$. The generator $G$ takes a latent variable $z$ from a prior density $p_0(z)$, $z \in \mathbb{R}^d$, and produces an observation $G(z) \in \mathbb{R}^d$ in the observation space. The discriminator takes a sample in the observation space and aims to distinguish between real examples and fake ones, produced by the generator. Recently, it has been advocated that considering GAN as an energy-based model increases the quality of the generated samples (Turner et al., 2019; Che et al., 2020). Following Che et al. (2020), we consider the EBM model induced by the GAN on the latent space, contrary to (Turner et al., 2019) which works in the observation space. Recall that an EBM is defined by a Boltzmann-Gibbs distribution $p(z) = e^{-E(z)/Z}$, $z \in \mathbb{R}^d$, where $E(z)$ is the energy function and $Z$ is the normalizing constant.
We set \( E(z) = -\log p_0(z) - \logit(D(G(z))) \), where \( \logit(y), y \in (0, 1) \) is the inverse of the sigmoid function and \( p_0(z) = g(z; 0; \text{Id}_d) \). The distribution \( p(z) \) would perfectly reproduce the target one even for the imperfect generator, provided that the discriminator is optimal. In most EBMs, samples are generated from \( p(z) \) by an MCMC algorithm, either using the Unadjusted Langevin Dynamics (ULA) or standard MCMC algorithms like MALA or HMC; see (Xie et al., 2018; Nijkamp et al., 2020; Song and Kingma, 2021) and the references therein. We advocate using Ex MCMC algorithm instead.

**GANs on synthetic data.** Following the setting used in (Che et al., 2020, Section 5.1), we apply Ex\(^2\)MCMC to a WGAN model (Arjovsky et al., 2017) trained on synthetic datasets. Implementation details and additional experiments are provided in Supplement Material, Appendix F.11. Our first example is a mixture of \( M = 243 \) Gaussians in \( \mathbb{R}^5 \), that is, the target distribution is \( M^{-1} \sum_{i=1}^{M} \mathcal{N}(\mu_i, \sigma^2 \text{Id}_5) \) with \( \sigma^2 = 1 \) and centers \( \{\mu_i\}_{i=1}^{M} \) equally spaced on a uniform grid at \( \{-2; 0; 2\}^5 \). To assess the sampling performance, we first assign each point \( x \) to its closest mode \( \mu_i(x) = \arg\min_{j \in [M]} \| x - \mu_j \|^2 \). The point \( x \) is tagged as an outlier if \( \| x - \mu_i(x) \|^2 \geq t \), where \( t \) is the 95\% quantile of the \( \chi^2 \)-distribution with 5 degrees of freedom. We compute mode-std as the sample variance of points in the neighborhood of each mode and number of captured modes as number of modes which were assigned with at least one point from the sample. We also compute the empirical Earth Mover’s distance (EMD) between the target and the empirical distribution. When applicable, we provide results both for single-start and multi-start regimes (see Appendix F.11 for more details). Results are summarized in Tables 1 and 2. This example illustrates that Langevin-based methods explore the support of distribution only in the multi-start regime. At the same time, each particular chain tends to stick in one of the modes of the latent distribution \( p(z) \). At the same time, Ex\(^2\)MCMC allows to achieve high sampling quality even for the single chain.

**GANs for CIFAR-10.** In this experiment we investigate performance of Ex\(^2\)MCMC algorithm for sampling from EBM for GAN on the CIFAR10 dataset. As a GAN model we consider two popular architectures, DC-GAN (Radford et al., 2016) and SN-GAN (Miyato et al., 2018).

We compare ULA, i-SIR, and Ex\(^2\)MCMC (both with correlated and independent proposals) methods. To evaluate sampling quality, we report the values of the energy function \( E(z) \), averaged over 1000 independent runs of each sampler. We present the results on Figure 7 together with the dynamics of the Frechet Inception Distance (FID, Heusel et al. (2017)), computed over the first 600 sampler iterations. Additional implementation details are provided in Appendix A. Note that all methods except for i-SIR improve in terms of average energy function values, and Ex\(^2\)MCMC algorithms allows for the best exploration.

```
Table 1: GAN sampling from 243 Gaussians

| Model             | mode std | # captured modes | EMD  |
|-------------------|----------|------------------|------|
| Vanilla GAN       | 0.042    | 44               | 4.36 |
| ULA (Che et al., 2020) | 0.035    | 35               | 4.04 |
| MALA              | 0.040    | 85               | 3.56 |
| Ex\(^2\)MCMC     | 0.039    | 69               | 3.60 |

Table 2: Results for Swiss Roll dataset

| Model | EMD    | STV |
|-------|--------|-----|
| WGAN-GP | 0.011  | 0.053 |
| ULA (Che et al., 2020) | 0.010  | 0.055 |
| MALA   | 0.011  | 0.043 |
| Ex\(^2\)MCMC | 0.011  | 0.063 |
```
5 Conclusions

We propose a new MCMC algorithm, Ex²MCMC, which allows to improve over the competitors due to the efficiency both at exploration and exploitation steps. We analyze its theoretical properties and suggest an adaptive version of the algorithm, FlEx²MCMC, based on normalizing flows. It allows to overcome the issues of MTM and i-SIR algorithms, which are caused by low acceptance rate when sampling from high dimensional distributions. Further studies of FlEx²MCMC, in particular its mixing rate, is an interesting direction for the future work.

Acknowledgement

The publication was supported by the grant for research centers in the field of AI provided by the Analytical Center for the Government of the Russian Federation (ACRF) in accordance with the agreement on the provision of subsidies (identifier of the agreement 000000D730321P5Q0002) and the agreement with HSE University No. 70-2021-00139.

References

Agapiou, S., Papaspiiliopoulos, O., Sanz-Alonso, D., and Stuart, A. M. (2017). Importance sampling: Intrinsic dimension and computational cost. Statistical Science, 32(3):405–431.

Allen, S. M. and Cahn, J. W. (1975). Coherent and incoherent equilibria in iron-rich iron-aluminum alloys. Acta Metallurgica, 23(9):1017–1026.

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. Journal of the Royal Statistical Society: Series B, 72(3):269–342.

Andrieu, C., Lee, A., Vihola, M., et al. (2018). Uniform ergodicity of the iterated conditional SMC and geometric ergodicity of particle Gibbs samplers. Bernoulli, 24(2):842–872.

Andrieu, C. and Thoms, J. (2008). A tutorial on adaptive MCMC. Statistics and computing, 18(4):343–373.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In International conference on machine learning, pages 214–223. PMLR.

Azadi, S., Olsson, C., Darrell, T., Goodfellow, I., and Odena, A. (2018). Discriminator Rejection Sampling. In International Conference on Learning Representations.

Azadi, S., Olsson, C., Darrell, T., Goodfellow, I., and Odena, A. (2019). Discriminator rejection sampling. arXiv:1810.06758.
metropolis algorithm. *Computational Statistics*, 14(3):375–395.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Hoffman, M. D., Gelman, A., et al. (2014). The no-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *ICLR 2015*.

Kish, L. (1965). *Survey sampling*. Chichester : Wiley New York.

Kobyzev, I., Prince, S., and Brubaker, M. (2020). Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Kolouri, S., Nadjahi, K., Simsekli, U., Badeau, R., and Gustavo, K. (2019). Generalized sliced Wasserstein distances. In *NeurIPS 2019*.

Laurent, B. and Massart, P. (2000). Adaptive estimation of a quadratic functional by model selection. *Ann. Statist.*, 28(5):1302–1338.

Lee, A. (2011). *On auxiliary variables and many-core architectures in computational statistics*. PhD thesis, University of Oxford.

Lee, A., Yau, C., Giles, M. B., Doucet, A., and Holmes, C. C. (2010). On the utility of graphics cards to perform massively parallel simulation of advanced Monte Carlo methods. *Journal of computational and graphical statistics*, 19(4):769–789.

Liang, F., Liu, C., and Carroll, R. (2011). *Advanced Markov chain Monte Carlo methods: learning from past samples*, volume 714. John Wiley & Sons.

Lindsten, F. and Schön, T. B. (2013). Backward simulation methods for Monte Carlo statistical inference. *Foundations and Trends® in Machine Learning*, 6(1):1–143.

Liu, J. S., Liang, F., and Wong, W. H. (2000). The multiple-try method and local optimization in Metropolis sampling. *Journal of the American Statistical Association*, 95(449):121–134.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. *arXiv:1802.05957*.

Monge, G. (1781). Mémoire sur la théorie des déblais et des remblais. *Histoire de l’Académie Royale des Sciences de Paris*.

Neal, R. M. (2011). MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, pages 113–162.

Nijkamp, E., Hill, M., Han, T., Zhu, S.-C., and Wu, Y. N. (2020). On the anatomy of MCMC-based maximum likelihood learning of energy-based models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5272–5280.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64.

Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv:1511.06434*.

Rezende, D. and Mohamed, S. (2015). Variational inference with normalizing flows. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France. PMLR.

Robert, C. and Casella, G. (2013). *Monte Carlo statistical methods*. Springer Science & Business Media.

Roberts, G. O. and Rosenthal, J. S. (2004). General state space Markov chains and MCMC algorithms. *Probability surveys*, 1:20–71.

Roberts, G. O. and Tweedie, R. L. (1996). Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms. *Biometrika*, 83(1):95–110.

Rubin, D. B. (1987). Comment: A noniterative Sampling/Importance Resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: The SIR algorithm. *Journal of the American Statistical Association*, 82(398):542–543.

Shestopaloff, A. Y. and Neal, R. M. (2018). Sampling latent states for high-dimensional non-linear state space models with the embedded HMM method. *Bayesian Analysis*, 13(3):797–822.

Skare, O., Bolviken, E., and Holden, L. (2003). Improved sampling-importance resampling and reduced bias importance sampling. *Scandinavian Journal of Statistics*, 30(4):719–737.

Smith, A. F. and Gelfand, A. E. (1992). Bayesian statistics without tears: a sampling–resampling perspective. *The American Statistician*, 46(2):84–88.
Song, Y. and Kingma, D. P. (2021). How to train your energy-based models. *arXiv preprint arXiv:2101.03288*.

Tanaka, A. (2019). Discriminator optimal transport. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 6816–6826.

Tjelmeland, H. (2004). Using all Metropolis–Hastings proposals to estimate mean values. Technical report.

Turner, R., Hung, J., Frank, E., Saatchi, Y., and Yosinski, J. (2019). Metropolis-Hastings generative adversarial networks. In *International Conference on Machine Learning*, pages 6345–6353. PMLR.

Wainwright, M. J. (2019). *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Xie, J., Lu, Y., Gao, R., and Wu, Y. N. (2018). Cooperative learning of energy-based model and latent variable model via MCMC teaching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
Supplementary Material

A Sampling GANs as energy-based model on CIFAR-10

We consider two popular GAN architectures, DC-GAN [Radford et al. (2016)] and SN-GAN [Miyato et al. (2018)]. Below we provide the details on experimental setup and evaluation for both of the models.

A.1 DC-GAN

For DC-GAN experiments, we took the pretrained GAN model after 200 epochs from the open repository [https://github.com/csinva/gan-vae-pretrained-pytorch] to compute FID we took code from the repository [https://github.com/abdulfatir/gan-metrics-pytorch].

For DC-GAN, the latent dimension equals $d = 100$. Following Che et al. (2020), we consider sampling from the latent space distribution

$$p(z) = e^{-E(z)/Z} \quad z \in \mathbb{R}^d, \quad E(z) = -\log p_0(z) - \logit(D(G(z))),$$

where $\logit(y) = \log(y/(1-y))$, $y \in (0,1)$ is the inverse of the sigmoid function, and $p_0(z) = g(z; 0; \text{Id}_d)$. For the step size $\gamma > 0$ we define $(k+1)$-th iteration of the Unadjusted Langevin Algorithm as

$$Z_{k+1} = Z_k - \gamma \nabla E(Z_k) + \sqrt{2\gamma} \epsilon_{k+1}, \epsilon_{k+1} \sim N(0, \text{Id}_d).$$

Note that the ergodic distribution of the corresponding continuous-time diffusion is given by $p(z) = e^{-E(z)/Z}$. We specify our choice of $\gamma$ in Table 6.

**Evaluation protocol** We run $n = 600$ iterations of the ULA, i-SIR, and Ex$^2$MCMC algorithm in its correlated and vanilla versions. For the vanilla Ex$^2$MCMC algorithm (Algorithm 5), we use the Markov kernel (50), corresponding to 1 MALA step, as the rejuvenation kernel. Step size $\gamma$, reported for Ex$^2$MCMC algorithm, corresponds to its rejuvenation MALA kernel. For the correlated Ex$^2$MCMC algorithm (see Algorithm 6), we bypass the rejuvenation step, since local exploration is guaranteed by selecting large $\alpha$. Additional details are provided at Table 3.

We run $N = 50000$ independent chains for each of the mentioned MCMC algorithms. Then, for $j$-th iteration, we calculate average value of the energy function $E(z)$ averaged over $M = 1000$ chains. Every 40 MCMC iterations we calculate FID based on 50000 images generated on the current MCMC step and 50000 images from the training data. We report FID results after 600 MCMC iterations at Table 5.

A.2 SN-GAN

For SN-GAN, we took an implementation available at [https://github.com/pfnet-research/sngan_projection] to reproduce results with the unconditional version of SN-GAN pretrained on CIFAR-10, to compute FID we took statistics from the repository [https://github.com/pfnet-research/chainer-gan-lib/blob/master/common/cifar-10-fid.npz] used in the SN-GAN repository. We additionally calibrated SN-GAN discriminator as suggested at Azadi et al. (2019), by replacing its top linear layer with fully-connected one consisting of 3 consecutive linear layers and finetuning them with binary cross entropy loss for 5k iterations.

| Method          | sampl. steps | $\gamma$ | $\#$ particles, $N$ | $\epsilon$ | $\alpha$ | num MALA steps |
|-----------------|--------------|----------|---------------------|-------------|----------|----------------|
| ULA             | 600          | 0.01     | -                   | -           | -        | -              |
| i-SIR           | 600          | -        | 5                   | -           | -        | -              |
| Vanilla Ex$^2$MCMC | 600     | 0.02     | 5                   | -           | -        | 1              |
| Correlated Ex$^2$MCMC | 600     | -        | 5                   | 0.9         | 0.99     | 0              |

Table 3: CIFAR-10 hyperparameters for DC-GAN architecture.
For SN-GAN, generator is a mapping $G(z) : \mathbb{R}^d \mapsto \mathbb{R}^D$ with latent dimension $d = 128$ and ambient space dimension $D = 784$. Following [Che et al. 2020], we performed an additional tempering of the distribution, that is, we sampled from the energy-based model

$$p(z) = e^{-E(z)/T} / Z, \quad z \in \mathbb{R}^d, \quad E(z) = -\log p_{\theta}(z) - \logit(D(G(z))).$$

In our experiments we observe, that simply setting $T = 1$ does not yield significant sampling quality improvement, compared to vanilla GAN sampling. At the same time, setting $T = 1/4$ allows to efficiently reduce the FID values, as reported in Figure 8.

Note that i-SIR and Ex2-MCMC with independent proposals fail to improve sampling quality compared to the vanilla GAN sampling, as it can be observed from Figure 8c-(a), and average discriminator scores $D(G(z))$, see Figure 8c-(c). Indeed, average discriminator score does not increase. At the same time, both ULA and Ex2MCMC with correlated particles shows considerable energy score improvement and better explore the regions of $p(z)$ with large values $D(G(z))$. Ex2MCMC shows more stable behaviour in terms of FID metrics, reported in Figure 8c-(b).

The evaluation protocol for SN-GAN follows the one for DC-GAN but with some adjustments, see Table 4. We run $N = 5000$ independent chains for each of the mentioned MCMC algorithms. Then, for $j$-th iteration, we calculate average value of the energy function $E(z)$ averaged over $N$ chains. Every 20 MCMC iterations we calculate FID based on 5000 images generated on the current MCMC step and 5000 images from the training data. We report FID results after 600 MCMC iterations at Table 5.

### B Proofs of main theoretical results of Section 2

**Notations** For $k \in \mathbb{N}$, $m, m' \in \mathbb{N}^*$ and $\Omega, \Omega'$ two open sets of $\mathbb{R}^m, \mathbb{R}^{m'}$ respectively, denote by $C^k(\Omega, \Omega')$, the set of $k$-times continuously differentiable functions. For $f \in C^2(\mathbb{R}^d, \mathbb{R})$, denote by $\nabla f$ the gradient of $f$ and by $\Delta f$ the Laplacian of $f$. For $k \in \mathbb{N}$ and $f \in C^k(\mathbb{R}^d, \mathbb{R})$, denote by $D^i f$ the $i$-th order differential of $f$ for $i \in \{0, \ldots, k\}$. For $x \in \mathbb{R}^d$ and $i \in \{1, \ldots, k\}$, define $\|D^i f(x)\| = |f(x)|$, $\|D^j f(x)\| = \sup_{u_1, \ldots, u_i \in \mathbb{R}^d} D^i f(x)[u_1, \ldots, u_i]$. For $k, p \in \mathbb{N}$ and $f \in C^k(\mathbb{R}^d, \mathbb{R})$, define

$$\|f\|_{k, p} = \sup_{x \in \mathbb{R}^d, i \in \{0, \ldots, k\}} \|D^i f(x)\|(1 + \|x\|^p).$$

| Method                  | sampl. steps | $\gamma$ | $\#$ particles, $N$ | $\epsilon$ | $\alpha$ | num MALA steps |
|------------------------|-------------|----------|---------------------|-----------|-------|----------------|
| ULA                    | 600         | 0.0025   | –                   | –         | –     | –              |
| i-SIR                  | 600         | –        | 4                   | –         | –     | –              |
| Vanilla Ex$^2$MCMC     | 600         | 0.005    | 4                   | –         | –     | 1              |
| Correlated Ex$^2$MCMC  | 600         | –        | 4                   | 0.5       | 0.995 | 0              |

Table 4: CIFAR-10 hyperparameters for SN-GAN architecture.
Define $C^k_{\text{poly}}(\mathbb{R}^d, \mathbb{R}) = \{ f \in C^k(\mathbb{R}^d, \mathbb{R}) : \inf_{p \in \mathbb{N}} \| f \|_{k,p} < +\infty \}$ and for any $f \in C^k_{\text{poly}}(\mathbb{R}^d, \mathbb{R})$, we consider the seminorm $\| f \|_k = \| f \|_{k,p}$ where $p = \min \{ q \in \mathbb{N} : \| f \|_{k,q} < +\infty \}$.

Finally, define $C^\infty_{\text{poly}}(\mathbb{R}^d, \mathbb{R}) = \cap_{k \in \mathbb{N}} C^k_{\text{poly}}(\mathbb{R}^d, \mathbb{R})$.

In the sequel, we denote by $w(x)$ the normalized weight function, that is,

$$\Pi(dx) = w(x) \Lambda(dx).$$  \hfill (22)

### B.1 Proof of Lemma 1

By symmetrisation, note that

$$P_N(x, A) = \int \delta_x(dx) \sum_{i=1}^N \frac{w(x^i)}{\sum_{j=1}^N w(x^j)} \Lambda(x^i) \prod_{j=2}^N \Lambda(dx^j)$$

$$= \frac{1}{N} \int \sum_{\ell=1}^N \delta_x(dx_\ell) \prod_{j \neq \ell} \Lambda(dx_\ell) \sum_{i=1}^N \frac{w(x^i)}{\sum_{\ell=1}^N w(x^\ell)} \Lambda(x^i).$$  \hfill (24)

Then,

$$\int \Pi(dx) P_N(x, A) = N^{-1} \int \Pi(dx) \sum_{\ell=1}^N \delta_x(dx_\ell) \prod_{j \neq \ell} \Lambda(dx_\ell) \sum_{i=1}^N \frac{w(x^i)}{\sum_{\ell=1}^N w(x^\ell)} \Lambda(x^i)$$

$$= N^{-1} \left( \sum_{\ell=1}^N w(x_\ell) \right) \prod_{j=1}^N \Lambda(dx_j) \sum_{i=1}^N \frac{w(x^i)}{\sum_{\ell=1}^N w(x^\ell)} \Lambda(x^i)$$

$$= N^{-1} \prod_{j=1}^N \Lambda(dx^j) \sum_{i=1}^N w(x^i) \Lambda(x^i) = \Pi(A).$$  \hfill (27)

### B.2 Proof of Theorem 3

We preface the proof by a technical lemma.

**Lemma 6.** Let $Y^{1:M}$ be i.i.d. random variables, such that $\mathbb{E}(Y_1) = 1$, and $\mathbb{P}(Y_1 \in [0, L]) = 1$. Then for $S = \sum_{i=1}^M Y_i$ and $a, b > 0$

$$\mathbb{E}\left[ (a + bS)^{-1} \right] \leq (a + bM/2)^{-1} + (1/a) \exp(-M/(2L^2)).$$  \hfill (28)

**Proof.** Let $K \geq 0$. Then we get

$$\frac{1}{a + bS} = \frac{1}{a + bS}1\{S < K\} + \frac{1}{a + bS}1\{S \geq K\}$$

$$\leq \frac{1}{a + bK}1\{S \geq K\} + \frac{1}{a + bS}1\{S < K\} \leq \frac{1}{a + bK} + \frac{1}{a}1\{S < K\}$$  \hfill (29)

and in particular, $\mathbb{E}[(a + bS)^{-1}] \leq (a + bK)^{-1} + a^{-1} \mathbb{P}(S < K)$. By Hoeffding’s inequality,

$$\mathbb{P}(S < K) = \mathbb{P}(S - M < -(M - K)) \leq \exp(-2(M - K)^2/(ML^2)).$$  \hfill (31)

In particular, for $K = M/2$, we have $\mathbb{P}(S < K) \leq \exp(-M/(2L^2))$. \hfill \qed
Proof of Theorem \textbf{3} (i) Under $A^1$ we have, for $(x,A) \in \mathbb{X} \times \mathcal{X}$,

$$
P_N(x,A) = \int \pi(dx) \sum_{i=1}^{N} \frac{w(x^i)}{\sum_{j=1}^{N} w(x^j)} 1_A(x^i) \prod_{j=2}^{N} \Lambda(dx^j) \tag{32}
$$

$$
= \int \frac{w(x)}{w(x) + \sum_{j=2}^{N} w(x^j)} 1_A(x) \prod_{j=2}^{N} \Lambda(dx^j) + \int \frac{w(x^j)}{w(x) + \sum_{j=2}^{N} w(x^j)} 1_A(x^j) \prod_{j=2}^{N} \Lambda(dx^j) \tag{33}
$$

$$
\geq \sum_{i=2}^{N} \int \frac{w(x^i)}{w(x) + w(x^i) + \sum_{j=2,j \neq i}^{N} w(x^j)} 1_A(x^i) \prod_{j=2}^{N} \Lambda(dx^j) \tag{34}
$$

Finally, since the function $f : z \mapsto (z + a)^{-1}$ is convex on $\mathbb{R}_+$ and $a > 0$, we get for $i \in \{2, \ldots, N\}$,

$$
\int \frac{1}{w(x) + w(x^i) + \sum_{j=2,j \neq i}^{N} w(x^j)} \prod_{j=2,j \neq i}^{N} \Lambda(dx^j) \tag{36}
$$

$$
\geq \int \frac{1}{w(x) + w(x^i)} \prod_{j=2}^{N} \Lambda(dx^j) \tag{37}
$$

$$
\geq \frac{1}{w(x) + w(x^i) + N - 2} \geq \frac{1}{2L + N - 2} \cdot \tag{38}
$$

We finally obtain the inequality

$$
P_N(x,A) \geq \Pi(A)^1 \times \frac{N - 1}{2L + N - 2} = \epsilon_N \Pi(A) \cdot \tag{39}
$$

This means that the whole space $\mathbb{X}$ is $(1, \epsilon_N \Pi)$-small (see \cite[Definition 9.3.5]{Douc et al. 2018}). Since $P_N(x,\cdot)$ and $\Pi$ are probability measures, \cite[39] implies

$$
\|P_N(x,\cdot) - \Pi\|_{TV} = \sup_{A \in \mathcal{X}} |P_N(x,A) - \Pi(A)| \leq 1 - \epsilon_N = \kappa_N \cdot \tag{40}
$$

Now the statement follows from \cite[Theorem 18.2.4]{Douc et al. 2018} applied with $m = 1$.

(ii) Let $V : \mathbb{X} \rightarrow [1, \infty)$ be a measurable function such that $\Pi(V) < \infty$ and $\Lambda(V) < \infty$. We aim to check first the drift condition

$$
P_N V(x) \leq \kappa_N V(x) + b_N \tag{41}
$$

with the constants $\kappa_N$ and $b_N$ defined in Theorem \textbf{3} and \textbf{51}, respectively. Setting $x^1 = x$, we obtain

$$
P_N V(x) = \int \sum_{i=1}^{N} \frac{w(x^i)}{\sum_{j=1}^{N} w(x^j)} V(x^i) \prod_{j=2}^{N} \Lambda(dx^j) \tag{42}
$$

$$
= V(x) \int \frac{w(x)}{\sum_{j=1}^{N} w(x^j)} \prod_{j=2}^{N} \Lambda(dx^j) + \int \sum_{i=2}^{N} \frac{w(x^i)}{\sum_{j=1}^{N} w(x^j)} V(x^i) \prod_{j=2}^{N} \Lambda(dx^j). \tag{43}
$$

We bound these two terms separately.

$$
V(x) \int \frac{w(x)}{\sum_{j=1}^{N} w(x^j)} \prod_{j=2}^{N} \Lambda(dx^j) = V(x) \int \left(1 - \frac{\sum_{j=2}^{N} w(x^j)}{\sum_{j=1}^{N} w(x^j)} \right) \prod_{j=2}^{N} \Lambda(dx^j) \tag{44}
$$

$$
= V(x) \left(1 - \sum_{k=2}^{N} \frac{w(x^k)}{w(x) + w(x^k) + \sum_{j=2,j \neq k}^{N} w(x^j)} \prod_{j=2}^{N} \Lambda(dx^j) \right) \tag{45}
$$

$$
= V(x) \left(1 - \sum_{k=2}^{N} \frac{\pi(dx^k)}{w(x) + w(x^k) + \sum_{j=2,j \neq k}^{N} w(x^j)} \prod_{j=2,j \neq k}^{N} \Lambda(dx^j) \right) \tag{46}
$$
From [36], we get that
\[
V(x) \int \frac{w(x)}{\sum_{j=1}^{N} w(x)} \prod_{j=2}^{N} \Lambda(dx^j) \leq V(x) \left(1 - \frac{N - 1}{2L + N - 2}\right) = \kappa_N V(x).
\] (47)

Moreover, we have
\[
\int \frac{w(x^i) V(x^i)}{\sum_{j=1}^{N} w(x^j)} \prod_{j=2}^{N} \Lambda(dx^j) = (N - 1) \int \frac{w(x^2) V(x^2) \Lambda(dx^2)}{w(x) + w(x^2) + \sum_{j=3}^{N} w(x^j)} \prod_{j=3}^{N} \Lambda(dx^j).
\] (48)

Since the function \( z \mapsto z/(z + a) \) is concave on \( \mathbb{R}_+ \) for \( a > 0 \), we have
\[
\int \frac{w(x^2)}{w(x) + w(x^2) + \sum_{j=3}^{N} w(x^j)} V(x^2) \Lambda(dx^2)
= \Lambda(V) \int \frac{w(x^2)}{w(x) + \sum_{j=3}^{N} w(x^j)} \frac{V(x^2) \Lambda(dx^2)}{\Lambda(V)}
\leq \Lambda(V) \frac{\int w(x^2) V(x^2) \Lambda(dx^2)/\Lambda(V) + w(x) + \sum_{j=3}^{N} w(x^j)}{\Pi(V)/\Lambda(V) + w(x) + \sum_{j=3}^{N} w(x^j)} \leq \Pi(V).
\] (49)

Using Lemma 6, with \( Y_i = w(x^i) \),
\[
\int \frac{\Pi(V)}{\Pi(V) + \sum_{j=3}^{N} w(x^j)} \prod_{j=3}^{N} \Lambda(dx^j) \leq \frac{\Pi(V)}{\Pi(V) + (N - 2)/2} + \Lambda(V) \exp(-(N - 2)/2L^2).
\] (50)

Writing in this case
\[
b_N = \frac{\Pi(V)(N - 1)}{\Pi(V) + (N - 2)/2} + \Lambda(V)(N - 1) \exp(-(N - 2)/2L^2).
\] (51)

concludes the proof.

Moreover, (i) implies that for any \( d_N > 1 \) the level sets \( \{ x \in \mathbb{X} : V(x) \leq d_N \} \) are \((1, \epsilon_N \Pi)\)-small. Let us choose \( d_N = 1 + 4b_N/(1 - \kappa_N) - 1 \vee 2/\kappa_N \). Then \( \kappa_N + 2b_N/(1 + d_N) < 1 \), and [Douc et al., 2018, Theorem 19.4.1] implies
\[
\|\mathcal{P}_N^\epsilon(x, \cdot) - \Pi\|_V \leq c_N \{V(x) + \Pi(V)\} \tilde{\kappa}_N^\alpha,
\] (52)

where the constants \( \tilde{\kappa}_N \) and \( c_N \) are given by\n\[
\log \tilde{\kappa}_N = \log \kappa_N \log 2b_N, \quad c_N = (\tilde{\lambda}_N + 1)(1 + \tilde{b}_N)/[\kappa_N(1 - \tilde{\lambda}_N)] \quad (53)
\]

\[
\tilde{\lambda}_N = \kappa_N + 2b_N/(1 + d_N), \quad \tilde{b}_N = \kappa_N b_N + d_N, \quad d_N = 1 + 4b_N/(1 - \kappa_N) - 1 \vee 2/\kappa_N.
\] (54)

In the expression above we used the fact that \( 1 - \epsilon_N = \kappa_N \). The choice of \( d_N \) in (53) implies \( 2b_N/(1 + d_N) \leq b_N \kappa_N \). Then the elementary calculations imply \( \kappa_N \leq \tilde{\lambda}_N \leq (b_N + 1)\kappa_N \) and
\[
\tilde{\kappa}_N \leq \kappa_N^{\theta_N},
\] (55)

where
\[
\theta_N = \frac{\log(1/\tilde{\lambda}_N)}{\log(1/\kappa_N) + \log(1/\tilde{\lambda}_N) + \log b_N} \geq \frac{\log(1/\kappa_N) - \log(b_N + 1)}{2\log(1/\kappa_N) + \log b_N} = \frac{1}{3} + \alpha_N.
\] (56)

In the expression above \( \alpha_N \to 0 \) when \( N \to \infty \), since \( b_N \leq 4\Pi(V) \) for \( N \geq 3 \). \( \square \)
B.3 Proof of Theorem Theorem 4

We preface the proof with some preparatory lemmas.

**Lemma 7.** Let $K \subset \mathbb{X}$, such that $\sup_{x \in K} w(x) < w_{\infty,K} < \infty$ and $\Pi(K) > 0$. Then, for all $(x, A) \in K \times \mathcal{X}$,

$$P_N(x, A) \geq \epsilon_{N,K} \Pi_K(A),$$

with $\epsilon_{N,K} = (N-1)\Pi(K)/[2w_{\infty,K} + N - 2]$ and $\Pi_K(A) = \Pi(A \cap K)/\Pi(K)$.

Note that if the weight function $w$ is continuous, then for any compact $K$, $\sup_{x \in K} w(x) < w_{\infty,K} < \infty$.

**Proof.** Let $(x, A) \in \mathbb{X} \times \mathcal{X}$. Then

$$P_N(x, A) = \int \frac{w(x)}{w(x) + \sum_{j=2}^N w(x')} 1_A(x) \prod_{i=2}^N \Lambda(dx') + \int \frac{w(x')}{w(x) + \sum_{j=2}^N w(x')} 1_A(x') \prod_{i=2}^N \Lambda(dx')$$

$$\geq \sum_{i=2}^N \int \frac{w(x')}{w(x) + w(x')} 1_A(x') \prod_{i=2}^N \Lambda(dx')$$

$$\geq \sum_{i=2}^N \frac{1}{w(x) + w(x')} \int w(x') + \sum_{j=2}^N w(x') \prod_{j=2,j\neq i}^N \Lambda(dx')$$

$$\geq (N - 1) \int \frac{1}{w(x) + w(x')} + \frac{N - 2}{2w_{\infty,K} + N - 2},$$

where the last inequality follows from Jensen’s inequality and the convexity of the function $z \mapsto (z + a)^{-1}$ on $\mathbb{R}_+$. Now,

$$P_N(x, A) \geq (N - 1) \int \frac{1}{w(x) + w(x')} + \frac{N - 2}{2w_{\infty,K} + N - 2} \Pi(dy) 1_{A \cap K}(y) = \frac{(N - 1)\Pi(K)}{2w_{\infty,K} + N - 2} \Pi_K(A).$$

**Lemma 8.** Let $P$ be a Markov kernel on $(\mathbb{X}, \mathcal{X})$, $\gamma$ be a probability measure on $(\mathbb{X}, \mathcal{X})$, and $\epsilon > 0$. Let also $C \in \mathcal{X}$ be an $(1, \epsilon\gamma)$-small set for $P$. Then for arbitrary Markov kernel $Q$ on $(\mathbb{X}, \mathcal{X})$, the set $C$ is an $(1, \epsilon\gamma_Q)$-small set for $PQ$, where $\gamma_Q(A) = \int \gamma(dy)Q(y, A)$ for $A \in \mathcal{X}$.

**Proof.** Let $(x, A) \in C \times \mathcal{X}$. Then it holds

$$PQ(x, A) = \int P(x, dy)Q(y, A) \geq \epsilon \int_C \gamma(dy)P(y, A) = \epsilon \gamma(A).$$

**Lemma 9.** Let $P$ and $Q$ be two irreducible Markov kernels with $\Pi$ as their unique invariant distribution. Let $V : \mathbb{X} \rightarrow [1, \infty)$ be a measurable function. Assume that there exists $\lambda_Q \in [0, 1)$ and $b_P, b_Q \in \mathbb{R}_+$ such that $PV(x) \leq V(x) + b_P$ and $QV(x) \leq \lambda_Q V(x) + b_Q$. Let $d_0 > 1$. Assume in addition, that for all $d \geq d_0$, there exist $\epsilon_d > 0$ and a probability measure $\gamma_d$ such that for all $(x, A) \in V_d \times \mathcal{X}$, $P(x, A) \geq \epsilon_d \gamma_d(A)$, where $V_d = \{x \in \mathbb{X} : V(x) \leq d\}$. Define $K = PQ$ and $\lambda_K = \lambda_Q$, $b_K = b_P + b_Q$. Then,

$$KV(x) \leq \lambda_K V + b_K \text{ and, for all } x \in V_d, K(x, A) \geq \epsilon_d \gamma_{Q,d}(A),$$

where $\gamma_{Q,d}(A) = \int \gamma_d(dy)Q(y, A)$.

Let $d \geq d_0$ be such that $\lambda_K + 2b_K/(1 + d) < 1$. Then, for any $x \in \mathbb{X}$ and $k \in \mathbb{N},$

$$\|K^k(x, \cdot) - \Pi_V\| \leq c_K \{V(x) + \Pi(V)\} \rho_K^k,$$
with \[ \rho_K = \frac{\log(1 - \epsilon_d) \log \bar{\lambda}_K}{\log(1 - \epsilon_d) + \log \bar{\lambda}_K - \log \bar{b}_K}, \] (65)

with \( \bar{\lambda}_K = \lambda_K + 2b_K/(1 + d), \) \( \bar{b}_K = \lambda_K b_K + d, \) and \( c_K = (\lambda_K + 1)(1 + \bar{b}_K/[(1 - \epsilon_d)/(1 - \bar{\lambda}_K)]). \)

Proof. By Lemma \[ \text{[8]} \], we have directly that for any \((x, A) \in \mathcal{V}_d \times \mathcal{X}, K(x, A) \geq \epsilon_d \gamma_{Q,d}(A). \) Moreover, for any \( x \in \mathcal{X}, KV(x) = P \mathcal{Q} V(x) \leq \lambda_Q \mathcal{P} V(x) + b_Q \leq \lambda_Q V(x) + b_Q + b_p. \) The proof is completed with \( \text{[Douc et al., 2018 Theorem 19.4.1].} \)

Proof of Theorem 10. Note first, that the Markov kernel of Ex\(^2\)MCMC algorithm can be represented as a composition \( K_N = P_N R \) with \( R \) being the rejuvenation kernel. Applying the same symmetrisation argument as \( \text{[53]}, \) we write \( K_N \) for \((x, A) \in \mathcal{X} \times \mathcal{X} \) as

\( K_N(x, A) = \frac{1}{N} \int \sum_{\ell = 1}^{N} \delta_x(dx_{\ell}) \prod_{i \neq \ell} \Lambda(dx^i) \sum_{i = 1}^{N} \frac{w(x^i)}{\sum_{i = 1}^{N} w(x^i)} R(x^i, A). \) (66)

Applying \( \text{[42]-[48]-[49]}, \) we get

\[ P_N V(x) = V(x) \int \frac{w(x)}{w(x) + \sum_{j = 2}^{N} w(x^j)} \prod_{j = 2}^{N} \Lambda(dx^j) + \int \sum_{i = 2}^{N} \frac{w(x^i)}{\sum_{i = 1}^{N} w(x^i)} V(x^i) \prod_{j = 2}^{N} \Lambda(dx^j) \] (67)

\[ \leq V(x) + (N - 1)U_N \] with \( U_N = \int \frac{\Pi(V)}{\Pi(V) + \lambda(V) + \sum_{j = 1}^{N} w(x^j)} \prod_{j = 3}^{N} \Lambda(dx^j) \) (68)

On the other hand, using \( \text{[29]} \) with \( K = (N - 2)/2 \) together with Markov inequality,

\[ U_N \leq \frac{\Pi(V)}{\Pi(V) + \lambda(V) + (N - 2)/2} + \lambda(V) \int \frac{1}{\sum_{j = 3}^{N} w(x^j)} \Pi(dx^j), \] (69)

\[ \leq \frac{\Pi(V) + 4 \lambda(V) \operatorname{Var}[w]}{N - 2}, \] (70)

where \( \operatorname{Var}[w] = \int (w(x) - 1)^2 \lambda(dx) \) is the variance of the normalized weight functions under the proposal distribution. Combining the above results, for any \( x \in \mathcal{X}, \)

\[ P_N V(x) \leq V(x) + b_p, \] where \( b_p_N = \frac{\Pi(V) - 1}{\Pi(V) + (N - 2)/2} + \frac{4(N - 1) \lambda(V) \operatorname{Var}[w]}{N - 2}. \) (71)

Assumption \( A^2 \text{(i)} \) implies \( RV(x) \leq \lambda_R V(x) + b_R. \) Assumption \( A^2 \text{together with Lemma [7]} \) implies that the level sets \( \mathcal{V}_d \) are \((1, \epsilon_{d,N} \gamma_d)\)-small for the Markov kernel \( P_N. \) Here the probability measure \( \gamma_d \) and \( \epsilon_{d,N} \) are given, for any \( A \in \mathcal{X}, \) by

\[ \gamma_d(A) = \int \Pi_{\mathcal{V}_d}(dy) R(y, A), \] where \( \Pi_{\mathcal{V}_d}(B) = \Pi(B \cap \mathcal{V}_d) / \Pi(\mathcal{V}_d), B \in \mathcal{X}, \) \( \epsilon_{d,N} = (N - 1) \Pi(\mathcal{V}_d)/[2 \epsilon_{w,d} + N - 2]. \) (72)

Hence all conditions of Lemma 9 are satisfied. Choose \( d_N = 1 + 4(b_R + b_{p_N})/(1 - \lambda_R) - 1. \) Then \( \lambda_R + 2(b_R + b_{p_N})/(1 + d_N) < 1, \) and Lemma 9 implies for any \( x \in \mathcal{X} \) and \( k \in \mathbb{N}, \)

\[ ||K_N^k(x, \cdot) - \Pi||_V \leq c_{N,R} \{ V(x) + V(V) \} \tilde{K}_{N,R}, \]

where the constants \( c_{N,R} \) and \( \tilde{K}_{N,R} \) are given by

\[ \log \tilde{K}_{N,R} = \frac{\log(1 - \epsilon_{d,N}) \log \bar{\lambda}_R}{\log(1 - \epsilon_{d,N}) + \log \bar{\lambda}_R - \log \bar{b}_R}, \] \( c_{N,R} = (\lambda_R + 1)(1 + \bar{b}_R/[(1 - \epsilon_{d,N})(1 - \bar{\lambda}_R)] \) (73)

\[ \bar{\lambda}_R = \lambda_R + 2b_R/(1 + d), \] \( \bar{b}_R = \lambda_R b_R + d_N, \) \( d_N = 1 + 4(b_R + b_{p_N})/(1 - \lambda_R) - 1, \) (74)

and \( \epsilon_{d,N} \) defined in \( \text{[71]}. \) It is easy to see from this expression that, for \( d_N \) being fixed, \( \tilde{K}_{N,R} \) decreases with \( N \rightarrow \infty. \)
B.4 Proof of Theorem Theorem 5

The algorithm Ex2MCMC defines a Markov chain \( \{ Y_j, \ j \in \mathbb{N} \} \) with Markov kernel

\[
C_N(x, A) = \frac{1}{N} \int \sum_{j=1}^{N} \delta_x(dx^j)Q_j(x^j, dx^{1..N}\setminus\{j\}) \times \sum_{i=1}^{N} \frac{w(x^i)}{\sum_{\ell=1}^{N} w(x^\ell)} R(x^i, A). \tag{75}
\]

Let \( f \) be a nonnegative measurable function. Using that \( \Pi(dy)\delta_y(dx^j) = \Pi(dx^j)\delta_x(dy), \Pi(dx^j) = \Lambda(dx^j)x \), and \( \Lambda(dx^j)Q_j(x^j, dx^{1..N}\setminus\{j\}) = \tilde{\Lambda}_N(dx^{1..N}) \), we get

\[
\int \Pi(dy)C_N(y, dy')f(y') = N^{-1} \int \sum_{j=1}^{N} w(x^j)x \tilde{\Lambda}_N(dx^j) \sum_{i=1}^{N} \frac{w(x^i)}{\sum_{\ell=1}^{N} w(x^\ell)} Rf(x^i)
= N^{-1} \int \sum_{i=1}^{N} w(x^i)Rf(x^i)\tilde{\Lambda}_N(dx^{1..N}). \tag{76}
\]

Using that

\[
w(x^i)\tilde{\Lambda}_N(dx^{1..N}) = w(x^i)\Lambda(dx^i)Q_j(x^i, dx^{1..N}\setminus\{i\}) = \Pi(dx^i)Q_i(x^i, dx^{1..N}\setminus\{i\}), \tag{78}
\]

we obtain

\[
N^{-1} \int \sum_{i=1}^{N} w(x^i)Rf(x^i)\tilde{\Lambda}_N(dx^{1..N}) = N^{-1} \int \sum_{i=1}^{N} \Pi(dx^i)Rf(x^i)Q_i(x^i, dx^{1..N}\setminus\{i\})
= \sum_{i=1}^{N} \int \Pi(dx^i)f(x^i) \int Q_i(x^i, dx^{1..N}\setminus\{i\}) = \int \Pi(dy)f(y), \tag{79}
\]

where we have used \( \int Q_i(x^i, dx^{1..N}\setminus\{i\}) = 1 \).

C Metropolis-Adjusted Langevin rejuvenation kernel

Assume that \( \mathbb{X} = \mathbb{R}^d \) and the target distribution \( \Pi \) is absolutely continuous with \( \Pi(dx) = \pi(x)dx \), where \( \pi(x) = \exp\{-U(x)\} \) with continuously differentiable function \( U(x) \). Then the MALA kernel is given, for \( \gamma > 0 \), \( x, z \in \mathbb{R}^d \), and \( A \in \mathcal{B}(\mathbb{R}^d) \), by

\[
R_{\gamma}^{\text{MALA}}(x, A) = \int_{\mathbb{R}^d} 1_A(x - \gamma \nabla U(x) + \sqrt{2} \gamma z) \min(1, e^{-\frac{\gamma}{\tau_{\gamma}^{\text{MALA}}}(x, z)} \varphi(z)dz
\]

\[
= \delta_x(A) \int_{\mathbb{R}^d} \{1 - \min(1, e^{-\frac{\gamma}{\tau_{\gamma}^{\text{MALA}}}(x, z)}\} \varphi(z)dz,
\]

\[
\tau_{\gamma}^{\text{MALA}}(x, z) = U(x - \gamma \nabla U(x) + \sqrt{2} \gamma z) - U(x)
+ (1/2)\{ \|z - (\gamma/2)\}^{1/2} \{\nabla U(x) + \nabla U(x - \gamma \nabla U(x) + \sqrt{2} \gamma z)\}^2 - \|z\|^2 \}. \tag{82}
\]

Note that the MALA kernel leaves the target \( \pi \) invariant. For notation simplicity, in this section we simply write \( R_{\gamma} \) instead of \( R_{\gamma}^{\text{MALA}} \). Consider the following assumptions on the target distribution:

**H 1.** \( U(x) \in C^\infty_{\text{poly}}(\mathbb{R}^d, \mathbb{R}) \), and \( \nabla U \) is Lipschitz, i.e. there exists \( L \geq 0 \) such that \( \|\nabla U(x) - \nabla U(y)\| \leq L\|x - y\| \) for all \( x, y \in \mathbb{R}^d \).

**H 2.** There exist \( K_1 \geq 0 \) and \( m > 0 \) such that for any \( x \notin B(0, K_1) \), and \( y \in \mathbb{R}^d \), \( \langle D^2 U(x)y, y \rangle \geq m\|y\|^2 \). Moreover, there exists \( M \geq 0 \) such that for any \( x \in \mathbb{R}^d \), \( \|D^3 U(x)\| \leq M \).

**Theorem 10.** Assume **H1** [2], **H2** [3] and that the proposal distribution has a continuous and positive p.d.f. \( \lambda(x) \). Then \( A^{(2)} \) is satisfied with \( \tilde{V}(x) = \exp(\tilde{\eta}\|x\|^2) \), \( \tilde{\eta} \) given in [95].

**Proof.** Proposition [18] implies that **A2** [ii] holds. Assume **H1** [1] and **H2**. Then there exist \( \bar{\gamma} > 0 \), \( \varpi > 0 \), and \( K_2, \bar{b} \geq 0 \) such that for any \( \gamma \in (0, \bar{\gamma}] \) and \( x \in \mathbb{R}^d \),

\[
R_{\gamma} V_{\varpi}(x) \leq \lambda_{R_{\gamma} V_{\varpi}}(x) + \varpi \rho_{R_{\gamma}} \rho_{\bar{\gamma}}, \text{ where } \lambda_{R_{\gamma}} = 1 - \varpi, \varpi = \bar{b} \gamma, \tag{83}
\]
and the parameters $\varpi$ and $\tilde{b}$ are given in \cite{135}. Moreover, since $\Lambda$ and $\Pi$ are continuous and $\Lambda(x) > 0$, the weight function $w(x)$ is bounded on the compact sets and $A2$ (iii) holds. Hence, all conditions of Lemma 9 are satisfied.

Consider now the sampling problem described in Appendix 3.4. We state the following result.

**Proposition 11.** Let $\pi(x) = g(x; 0; \text{Id})$ and $\lambda(x) = g(x; 0, 2\text{Id})$. Let $V_\eta(x) = \exp(\eta \|x\|^2)$, where $\eta$ is given in (95). Then, for $N \geq e^{2\eta D}$, the Markov kernel of Ex$^2$MCMC algorithm with MALA rejuvenation kernel satisfies

$$
\|K_N^k(x, \cdot) - \Pi\|_V \leq c_K \{V_\eta(x) + \Pi(V_\eta)\} \kappa_{N,R}^k, \quad \text{where} \quad \log \kappa_{N,R} \leq \frac{\log(1 - \varpi \gamma)}{1 + 8\eta} + \alpha_N, \quad (84)
$$

and $\alpha_N \to 0$ for $N \to \infty$, and $c_K$ is given in (74).

**Proof.** From the proof of Theorem 10 it follows that

$$
\|K_N^k(x, \cdot) - \Pi\|_V \leq c_{N,R} \{V(x) + \Pi(V)\} \kappa_{N,R}^k,
$$

where the constants $c_{N,R}$ and $\kappa_{N,R}$ are given in (74). We first estimate the quantity $\epsilon_{d,N} = (N - 1)\Pi(V_{d,N})/[2w_{\infty,d,N} + N - 2]$. Note that

$$
w_{\infty,d,N} \leq \sup_{x \in \mathbb{R}^D} \frac{\pi(x)}{\lambda(x)} = 2D/2,
$$

which does not depend on the choice of $d_N$. Hence, for any choice $d_N$, $\epsilon_{d,N}/\Pi(V_{d,N}) \to 1$ as $N \to \infty$. Now we can lower bound $\Pi(V_{d,N})$ as follows. Note that $V_{d,N} = \{x \in \mathbb{R} : V_\eta(x) \leq d_N\} = \{x \in \mathbb{R} : \|x\|^2 \leq \log d_N/\eta\}$. Then, for $\log d_N/\eta \geq 2D$, we write

$$
\Pi(V_{d,N}) = \mathbb{P} \left(\|Z\|^2 \leq \left(\frac{\log d_N}{\eta} - D\right) + D\right) \geq 1 - \exp \left\{-\frac{\log d_N}{8\eta} + \frac{D}{8}\right\}, \quad (85)
$$

where $Z = (Z_1, \ldots, Z_D) \sim \Pi$. Here we use the fact that $\|Z\|^2$ is a chi-squared random variable with $D$ degrees of freedom and \cite[Proposition 2.2]{19}. Hence, $\log(1 - \epsilon_{d,N}) \leq -\left(\frac{\log d_N}{8\eta} - \frac{D}{8}\right)$. Then using the expression (74) and choosing

$$
d_N = 1 \lor 4(b_R + b_{\rho,N})/(1 - \lambda_R) - 1 \lor N \lor e^{2\eta D}, \quad (86)
$$

we have

$$
\log \kappa_{N,R} = \frac{\log(1 - \epsilon_{d,N}) \log \lambda_R}{\log(1 - \epsilon_{d,N}) + \log \lambda_R - \log b_R} = \frac{\log \lambda_R}{1 + \log \lambda_R/\log(1 - \epsilon_{d,N}) - \log b_R/\log(1 - \epsilon_{d,N})}. \quad (87)
$$

Note that $\log \lambda_R/\log(1 - \epsilon_{d,N}) \to 0$, $N \to \infty$. Now the statement follows from $\log b_R/\log(1 - \epsilon_{d,N}) = -8\eta + \beta_N$, $\beta_N \to 0$ for $N \to \infty$.

**D Technical lemmas for Metropolis-Adjusted Langevin kernel**

The goal of this section is to establish the Foster-Lyapunov drift condition for the Markov kernel $R_{\gamma}^{\text{MALA}}$ defined in (80). As an auxiliary result we need to establish the drift condition for the Markov kernel $Q_{\gamma}^{\text{ULA}}$, defined as

$$
Q_{\gamma}^{\text{ULA}}(x, A) = \int_{\mathbb{R}^d} 1_A \left(x - \gamma \nabla U(x) + \sqrt{2\gamma} z\right) \varphi(z) dz, \quad (88)
$$

where $\varphi$ is the $d$-dimensional standard Gaussian density $\varphi(z) = (2\pi)^{-d/2} e^{-\|z\|^2}$. Define for any $\eta > 0$, $V_\eta(x) : \mathbb{R}^d \to [1, +\infty)$ as

$$
V_\eta(x) = \exp(\eta \|x\|^2). \quad (89)
$$

For notation simplicity, we write in this section $Q_\gamma$ instead of $Q_{\gamma}^{\text{ULA}}$, and $R_\gamma$ instead of $R_{\gamma}^{\text{MALA}}$. We begin with the technical lemma.
Lemma 12. Assume H[1] and H[2]. Then there exists $K_2 \geq 0$ such that for any $x \not\in B(0,K_2)$, $\langle \nabla U(x), x \rangle \geq (m/2)\|x\|^2$ and in particular $\|\nabla U(x)\| \geq (m/2)\|x\|$.

Proof. Using H[1] and H[2] we have for any $x \in \mathbb{R}^d$, $\|x\| \geq K_1$,

$$\langle \nabla U(x), x \rangle = \int_0^{K_1/\|x\|} D^2 U(tx)[x^{\otimes 2}] dt + \int_{K_1/\|x\|}^1 D^2 U(tx)[x^{\otimes 2}] dt \geq m\|x\|^2 \{1 - K_1(1 + L/m)/\|x\|\} , \tag{90}$$

which proves the first statement. The second statement easily follows from the Cauchy-Schwartz inequality. \hfill \square

Lemma 13. Assume H[1] and H[2]. Then, for any $t \in [0, 1]$, $\gamma \in (0, 1/(4L)]$ and $x, z \in \mathbb{R}^d$, $\|z\| \leq \|x\|/(4\sqrt{2\gamma})$, it holds

$$\|x + t(-\gamma \nabla U(x) + \sqrt{2\gamma}z)\| \geq \|x\|/2 . \tag{92}$$

Proof. Let $t \in [0, 1]$, $\gamma \in (0, 1/(4L)]$ and $x, z \in \mathbb{R}^d$, $\|z\| \leq \|x\|/(4\sqrt{2\gamma})$. Using the triangle inequality and H[1] we have since $t \in [0, 1]$

$$\|x + t(-\gamma \nabla U(x) + \sqrt{2\gamma}z)\| \geq (1 - \gamma L)\|x\| - \sqrt{2\gamma}\|z\| . \tag{93}$$

The conclusion then follows from $\gamma \leq 1/(4L)$ and $\|z\| \leq \|x\|/(4\sqrt{2\gamma})$. \hfill \square

Now we establish the drift condition for $Q^\gamma_{ULA}$.

Lemma 14. Assume H[1] and H[2] and let $\bar{\gamma} \in (0, m/(4L^2)]$. Then, for any $\gamma \in (0, \bar{\gamma}]$,

$$Q_\gamma V_{\bar{\gamma}}(x) \leq \exp \{-\bar{\eta}m\gamma\|x\|^2/4\} V_{\bar{\gamma}}(x) + b_{\bar{\eta}} \gamma 1_{B(0,K_3)}(x) , \tag{94}$$

where $V_{\bar{\gamma}}$ is defined by $89$, $\bar{\eta} = \min(m/16, (8\bar{\gamma})^{-1})$, $K_3 = \max(K_2, 4\sqrt{d/m})$, and

$$b_{\bar{\eta}} = \left[ \bar{\eta} \left\{ m/4 + (1 + 16\bar{\eta}\bar{\gamma})(4\bar{\eta} + 2\bar{\gamma}L^2) \right\} K_3^2 + 4\bar{\eta}d \right] \times \exp \left[ \bar{\gamma}\bar{\eta} \left\{ m/4 + (1 + 16\bar{\eta}\bar{\gamma})(4\bar{\eta} + 2\bar{\gamma}L^2) \right\} K_3^2 + (d/2) \log(2) \right] . \tag{95}$$

Proof. Let $\gamma \in (0, \bar{\gamma}]$. For any $x \in \mathbb{R}^d$, we have

$$\bar{\eta}\|x - \gamma \nabla U(x) + \sqrt{2\gamma}z\|^2 - \|z\|^2/2 = - \frac{1 - 4\bar{\eta}\gamma}{2} \|z\|^2 - \frac{2(2\gamma)^{1/2}\bar{\eta}}{1 - 4\bar{\eta}\gamma} \|x - \gamma \nabla U(x)\|^2 + \frac{\bar{\eta}}{1 - 4\bar{\eta}\gamma} \|x - \gamma \nabla U(x)\|^2 . \tag{96}$$

Since $1 - 4\bar{\eta}\gamma > 0$, we get that

$$Q_\gamma V_{\bar{\gamma}}(x) = (2\pi)^{-d/2} \int_{\mathbb{R}^d} \exp \left( \bar{\eta}\|x - \gamma \nabla U(x) + \sqrt{2\gamma}z\|^2 - \|z\|^2/2 \right) dz = (1 - 4\bar{\eta}\gamma)^{-d/2} \exp \left( \bar{\eta}(1 - 4\bar{\eta}\gamma)^{-1}\|x - \gamma \nabla U(x)\|^2 \right) . \tag{97}$$

We now distinguish the case when $\|x\| \geq K_3$ and $\|x\| < K_3$. By H[2] and Lemma 12 for any $x \in \mathbb{R}^d$, $\|x\| \geq K_3 \geq K_2$, using that $\bar{\eta} \leq m/16$ and $\gamma \leq \bar{\gamma} \leq m/(4L^2)$, we have

$$(1 - 4\bar{\eta}\gamma)^{-1}\|x - \gamma \nabla U(x)\|^2 - \|z\|^2 \leq \gamma\|x\|^2(1 - 4\bar{\eta}\gamma)^{-1} (4\bar{\eta} - m + \gamma L^2) \leq -\gamma(m/2)\|x\|^2(1 - 4\bar{\eta}\gamma)^{-1} . \tag{98}$$

Therefore, (97) becomes

$$Q_\gamma V_{\bar{\gamma}}(x) \leq \exp \{-\bar{\eta}m(2)(1 - 4\bar{\eta}\gamma)^{-1}\|x\|^2 - (d/2) \log(1 - 4\bar{\eta}\gamma)\} V_{\bar{\gamma}}(x) \leq \exp \{\bar{\eta}(m/2)\|x\|^2 + (d/2) \log(1 - 4\bar{\eta}\gamma)\} V_{\bar{\gamma}}(x) , \tag{99}$$

where we have used for the last inequality that $-\log(1 - t) \leq 2t$ for $t \in [0, 1/2]$ and $4\bar{\eta}\gamma \leq 1/2$. The proof of the statement then follows since $\|x\| \geq K_3 \geq 4\sqrt{d/m}$. 

In the case $|x| < K_3$, by (97), (98) and since $(1 - t)^{-1} \leq 1 + 4t$ for $t \in [0, 1/2]$, we obtain
\begin{align}
(1 - 4\bar{\gamma})^{-1}\|x - \gamma \nabla U(x)\|^2 - \|x\|^2 &\leq \gamma (1 - 4\bar{\gamma})^{-1} \{4\bar{\gamma} + 2L + \gamma L^2\}\|x\|^2 \\
&\leq \gamma (1 + 6\bar{\gamma}) \{4\bar{\gamma} + 2L + \gamma L^2\}\|x\|^2 ,
\end{align}
which implies that
\begin{equation}
Q_\gamma V_\eta(x)/V_0(x) \leq e^{-\bar{\gamma}m\|x\|^2/4} + \exp \left[\gamma \bar{\eta} \{m/4 + (1 + 6\bar{\gamma}) (4\bar{\gamma} + 2L + \gamma L^2)\}\|x\|^2 - (d/2) \log(1 - 4\bar{\gamma})\}^{-1} .
\end{equation}
The proof is then completed using that for any $t \geq 0$, $e^t - 1 \leq te^t$, for any $s \in [0, 1/2]$, $-\log(1 - s) \leq 2s$ and $4\bar{\gamma} \leq 1/2$.

We now provide a decomposition in $\gamma$ of $\tau_\gamma^{MALA}$ defined in (81). For any $x, z \in \mathbb{R}^d$, by (Durmus et al., 2017, Lemma 24) we have that
\begin{equation}
\tau_\gamma^{MALA}(x, z) = \sum_{k=2}^{6} \gamma^{k/2} A_{k, \gamma}(x, z)
\end{equation}
where, setting $x_t = x + t\{-\gamma \nabla U(x) + \sqrt{2}\gamma z\}$,
\begin{align}
A_{2, \gamma}(x, z) &= \int_0^1 (1/2 - t) dt \\
A_{3, \gamma}(x, z) &= 2^{3/2} \int_0^1 (1/2 - t) dt \\
A_{4, \gamma}(x, z) &= -\int_0^1 (1/2 - t) dt \\
A_{5, \gamma}(x, z) &= -(1/2)^{1/2} \int_0^1 (1/2 - t) dt \\
A_{6, \gamma}(x, z) &= (1/4) \int_0^1 (1/2 - t) dt.
\end{align}

**Lemma 15.** Assume $A_1$ and $H_3$. Then, for any $\bar{\gamma} > 0$, there exists $C_{1, \bar{\gamma}} < \infty$ such that for any $x, z \in \mathbb{R}^d$, $\gamma \in (0, \bar{\gamma}]$, it holds
\begin{equation}
|\tau_\gamma^{MALA}(x, z)| \leq C_{1, \bar{\gamma}} \gamma^{3/2} \{1 + \|z\|^4 + \|x\|^2\} .
\end{equation}

**Proof.** Since $\int_0^1 (1/2 - t) dt = 0$, we get setting $x_t = x + t\{-\gamma \nabla U(x) + \sqrt{2}\gamma z\}$,
\begin{align}
A_{2, \gamma}(x, z) &= \int_0^1 \int_0^1 (1/2 - t) dt \\
&= \sqrt{\gamma} \int_0^1 \int_0^1 (1/2 - t) dt \\
&= \gamma \int_0^1 \int_0^1 (1/2 - t) dt .
\end{align}
The proof follows from $\sup_{x \in \mathbb{R}^d} \|D^2 U(x)\| \leq L$ and $\sup_{x \in \mathbb{R}^d} \|D^3 U(x)\| \leq M$.

**Lemma 16.** Assume $A_1$ and $H_3$. Then, for any $\bar{\gamma} \in (0, m^3/(4L^4)]$ there exists $C_{2, \bar{\gamma}} < \infty$ such that for any $\gamma \in (0, \bar{\gamma}]$, $x, z \in \mathbb{R}^d$ satisfying $\|x\| \geq \max(2K_1, K_2)$ and $\|z\| \leq \|x\|/(4\sqrt{2\gamma})$, where $K_2$ is defined in Lemma 12 it holds
\begin{equation}
\tau_\gamma^{MALA}(x, z) \leq C_{2, \bar{\gamma}} \gamma \|z\|^2 \{1 + \|z\|^2\} .
\end{equation}

**Proof.** Let $\gamma \in (0, \bar{\gamma}]$, $x, z \in \mathbb{R}^d$ satisfying $\|x\| \geq \max(2K_1, K_2)$ and $\|z\| \leq \|x\|/(4\sqrt{2\gamma})$. Using (104), we get setting
\begin{align}
A_{4, \gamma}(x, z) &= \int_0^1 \int_0^1 (1/2 - t) dt \\
&= \gamma \int_0^1 \int_0^1 (1/2 - t) dt .
\end{align}

Note that with the notation of (Durmus et al., 2017, MALA corresponds to HMC with only one leapfrog step and step size equals to $(2\gamma)^{1/2}$
Then, by Lemma 16 and Lemma 17, there exist $\gamma > 0$ and $\gamma \in (0, \gamma]$ such that for any $x \in \mathbb{R}^d$, $\|x\| \geq 20\sqrt{\gamma d}$,

$$
\int_{\mathbb{R}^d \setminus B(0,\|x\|/(4\sqrt{\gamma}))} \varphi(z) dz \leq \exp(-\|z\|^2/(128\gamma)) .
$$

**Proof.** Let $x > 0$. By [Laurent and Massart (2000)](Laurent and Massart (2000) Lemma 1),

$$
P(\|Z\|^2 \geq 2(\sqrt{d} + \sqrt{x})^2) \leq P(\|Z\|^2 \geq d + 2\sqrt{dx} + 2x) \leq e^{-x},
$$

where $Z$ is a $d$-dimensional standard Gaussian vector. Setting $t = 2(\sqrt{d} + \sqrt{x})^2$, we obtain

$$
P(\|Z\|^2 \geq t) \leq \exp\left(-\left\{d + t/2 - 2\sqrt{td}\right\}\right) ,
$$

and for $\sqrt{t} \geq 5\sqrt{d}$, we get $P(\|Z\| \geq \sqrt{t}) \leq e^{-t/4}$ which gives the result. \hfill \Box

**Proposition 18.** Assume $H1$ and $H2$. Then there exist $\bar{\gamma}_1 > 0$, $\varpi > 0$, and $K_2, \bar{b} \geq 0$ such that for any $\gamma \in (0, \bar{\gamma}_1]$ and $x \in \mathbb{R}^d$,

$$
R_{\gamma} V_\gamma(x) \leq (1 - \varpi \gamma) V_\gamma(x) + \bar{b}_1 \gamma_1 1_{(0,K_2)}(x) ,
$$

where $V_\gamma$ is defined by (80), $R_{\gamma}$ is the MALA kernel given in (80) and $\bar{\gamma}_1$ is given by (95). \hfill \Box

**Proof.** Let $\bar{\gamma}_1 = m/(4L^2)$. By Lemma 14 for any $\gamma \in (0, \bar{\gamma}_1]$ and $x \in \mathbb{R}^d$,

$$
R_{\gamma} V_\gamma(x) \leq Q_\gamma V_\gamma(x) + V_\gamma(x) \int_{\mathbb{R}^d} \{1 - \min(1, e^{-\gamma_{\text{MALA}}(x,z)})\} \varphi(z) dz
$$

$$
\leq e^{-\bar{\gamma} m_1 \|x\|^2/4V_\gamma(x)} + b_1 \gamma_1 1_{(0,K_2)}(x) + V_\gamma(x) \int_{\mathbb{R}^d} \{1 - \min(1, e^{-\gamma_{\text{MALA}}(x,z)})\} \varphi(z) dz ,
$$

where $K_3$ and $b_1$ are given in (95). Let

$$
\bar{\gamma}_2 = \min\left(1, \bar{\gamma}_1, \frac{m^3}{(4L^4)}\right) , \quad M_1 = \max\left(1, 2K_1, K_2, K_3, 20\sqrt{2\gamma_2 d}\right) .
$$

Then, by Lemma 16 and Lemma 17 there exist $C_1 \geq 0$ such that for any $x \in \mathbb{R}^d$, $\|x\| \geq M_1$ and $\gamma \in (0, \bar{\gamma}_2]$,

$$
R_{\gamma} V_\gamma(x) \leq e^{-\bar{\gamma} m_1 \|x\|^2/4V_\gamma(x)} V_\gamma(x) \left\{C_1 \gamma + \exp(-\|x\|^2/(128\gamma))\right\}
$$

$$
\leq e^{-\bar{\gamma} m_1 \|x\|^2/4V_\gamma(x)} + V_\gamma(x) \left\{C_1 \gamma + \exp(-1/(128\gamma))\right\} .
$$

E. Lagutin, D. Selikhannovich, A. Thin, S. Samsonov, A. Naumov, D. Belomestny, M. Panov, E. Moulines
Using that there exists $C_2 \geq 0$ such that $\sup_{t \in (0,1)} \{ t^{-1} \exp(-1/(128t)) \} \leq C_2$ we get for any $x \in \mathbb{R}^d$, $\|x\| \geq M_1$, $\gamma \in (0, \bar{\gamma}_2)$,

$$R_\gamma V_\eta(x) \leq \exp(-\bar{\gamma} \|x\|^2/4)V_\eta(x) + V_\eta(x)\gamma \{ C_1 + C_2 \}.$$ \hfill (129)

Let

$$M_2 = \max \left( M_1, 4(C_1 + C_2)^{1/2}(\bar{\gamma}m)^{-1/2} \right), \quad \bar{\gamma}_3 = \min \left( \bar{\gamma}_2, 4 \{ m\bar{\gamma}M_2^2 \}^{-1} \right).$$ \hfill (130)

Then, since for any $t \in [0,1]$, $e^{-t} \leq 1 - t/2$, we get for any $x \in \mathbb{R}^d$, $\|x\| \geq M_2$, $\gamma \in (0, \bar{\gamma}_3)$,

$$R_\gamma V_\eta(x) \leq \exp(-\gamma \bar{\gamma} M_2^2/4)V_\eta(x) + V_\eta(x)\gamma \{ C_1 + C_2 \} \leq \left[ 1 - \gamma \{ \bar{\gamma}mM_2^2/8 - C_1 - C_2 \} \right] V_\eta(x) \leq \left[ 1 - \bar{\gamma}\bar{\gamma} M_2^2/16 \right] V_\eta(x).$$ \hfill (131)

In addition, by Equation (110), using that for any $t \in \mathbb{R}$, $1 - \min(1, e^{-t}) \leq |t|$, there exists $C_3 \geq 0$ such that for any $x \in \mathbb{R}^d$, $\|x\| \leq M_2$ and $\gamma \in (0, \bar{\gamma}_3)$,

$$R_\gamma V_\eta(x) \leq V_\eta(x) + b_\eta \gamma_1 B_{(0,K_3)}(x) + C_3 \gamma^{3/2} \int_{\mathbb{R}^d} \{ 1 + \|x\|^2 + \|z\|^4 \} \varphi(z) dz \leq (1 - \gamma \bar{\gamma} m M_2^2/16)V_\eta(x) + \gamma \bar{\gamma} m M_2^2 e^{\bar{\gamma} M_2^2}/16 + b_\eta + C_3 \gamma^{1/2} \{ 1 + M_2^2 + C_4 \},$$ \hfill (132)

where $C_4 = \int_{\mathbb{R}^d} \|z\|^4 \varphi(z) dz$. Hence, (123) holds with

$$\varpi = \bar{\gamma} m M_2^2/16, \quad \bar{b} = \bar{\gamma} m M_2^2 e^{\bar{\gamma} M_2^2}/16 + b_\eta + C_3 \gamma^{1/2} \{ 1 + M_2^2 + C_4 \}.$$ \hfill (133)

\[\square\]

### E Algorithms

In this section, we have compiled the detailed description of all the algorithms we use in the text:

- Algorithm 4 for i-SIR;
- Algorithm 5 for ExMCMC with independent proposals;
- Algorithm 6 for ExMCMC with dependent proposals.

The unifying information for the hyperparameter selection in all the considered experiments is given in Table 6.

\begin{algorithm}
\textbf{Algorithm 4:} Single stage of i-SIR algorithm with independent proposals
\begin{algorithmic}[1]
\Require Sample $Y_j$ from previous iteration
\Ensure New sample $Y_{j+1}$
\State 1. Set $X^{1}_{j+1} = Y_j$ and draw $X_{j+1} \sim \Lambda$
\For{$i \in [N]$}
\State 2. Compute the normalized weights $\omega_{i,j+1} = \tilde{w}(X^i_{j+1}) / \sum_{k=1}^{N} \tilde{w}(X^k_{j+1})$.
\State 3. Set $I_{j+1} = \text{Cat}(\omega_{1,j+1}, \ldots, \omega_{N,j+1})$.
\State 4. Draw $Y_{j+1} = X^{I_{j+1}}_{j+1}$.
\EndFor
\end{algorithmic}
\end{algorithm}

### F Numerical experiments

We provide here additional information to the simulation problems of the main document and present results on new simulation problems.
Algorithm 5: Single stage of Ex^2MCMC algorithm with independent proposals

**Input**: Sample \( Y_j \) from previous iteration

**Output**: New sample \( Y_{j+1} \)

1. Set \( X_{j+1}^{1:N} = Y_j \) and draw \( X_{j+1}^{1:N} \sim \Lambda \).
2. for \( i \in [N] \) do
3.  compute the normalized weights \( \omega_{i,j+1} = \tilde{w}(X_{j+1}^i)/\sum_{k=1}^N \tilde{w}(X_{j+1}^k) \).
4. Set \( I_{j+1} = \text{Cat}(\omega_{1,j+1}, \ldots, \omega_{N,j+1}) \).
5. Draw \( Y_{j+1} \sim \text{R}(X_{j+1}^{I_{j+1}}, \cdot) \).

Algorithm 6: Proposal generation procedure for FlEx^2MCMC algorithm with dependent proposal

**Input**: Sample \( Y_j \) from previous iteration

**Output**: Set of proposals for the current iteration \( X_{j+1}^{1:N} \)

1. Draw \( U_{j+1} \sim \text{Unif}([N]) \) and set \( X_{j+1}^{U_{j+1}} = Y_j \)
2. Set \( Z_{j+1}^{U_{j+1}} = T_{\theta_j}^{-1}(X_{j+1}^{U_{j+1}}) \)
3. Draw \( \alpha_{j+1}^{U_{j+1}} \sim \nu \)
4. Draw \( \xi_{j+1} \sim N(\alpha_{j+1}^{U_{j+1}} \sigma_{U_{j+1}}^2, \sigma_{U_{j+1}}^2) \)
5. for \( i \in [N] \) \( \{U_{j+1}\} \) do
6.  draw \( W_{j+1}^i \sim N(0, I_d) \) and \( \alpha_j^i \sim \nu \) and set
7.  \( Z_{j+1}^i = \alpha_j^i \xi_{j+1} + \sqrt{1 - \{\alpha_j^i\}^2} W_{j+1}^i \)
8. Set \( X_{j+1}^{1:N \setminus U_{j+1}} = T_{\theta_j}(Z_{j+1}^{1:N \setminus U_{j+1}}) \)

Algorithm 7: Single stage of FlEx^2MC with independent proposals. Steps 1-7 are done in parallel for independent chains indexed by \( k \) but with common values of proposal parameters \( \theta_j \). Step 9 updates the parameters using the gradient estimate obtained from all the chains.

**Input**: weights \( \theta_j \), batch \( Y_j[1 : K] \)

**Output**: new weights \( \theta_{j+1} \), batch \( Y_{j+1}[1 : K] \)

1. for \( k \in [K] \) do
2.  Set \( X_j^{1:N}[k] = Y_j[k] \).
3.  Draw \( Z_{j+1}^{1:N \setminus \{k\}}[k] \sim \Lambda \).
4.  Set \( X_{j+1}^{1:N \setminus \{k\}}[k] = T_{\bar{\theta}_j}(Z_{j+1}^{1:N \setminus \{k\}}[k]) \).
5. for \( i \in [N] \) do
6.  compute the unnormalized weights \( \tilde{w}_{i,j+1}[k] = \tilde{w}_{\theta_j}(X_{j+1}^i[k]) \).
7. Compute \( \Omega_N[j + 1, k] = \sum_{i=1}^N \tilde{w}_{i,j+1}[k] \) and the normalized weights \( \omega_{i,j+1}[k] = \tilde{w}_{i,j+1}[k]/\Omega_N[j + 1, k] \).
8. Set \( I_{j+1}[k] = \text{Cat}(\omega_{1,j+1}[k], \ldots, \omega_{N,j+1}[k]) \).
9. Draw \( Y_{j+1}[k] \sim \text{R}(X_{j+1}^{I_{j+1}[k]}, \cdot) \).
10. Draw \( Z[1 : K] \sim \Lambda \).
11. Update \( \theta_{j+1} = \theta_j - \gamma \nabla \tilde{C}(Y_{j+1}, \bar{Z}, \theta_j) \).

F.1 Metrics

**ESTD**: To compute ESTD, we perform 10 random one-dimensional projections and then perform Kernel Density Estimation there for reference and produced samples, and take TV-distance between two distributions over 1D grids of 1000 points. We consider the value averaged over the projections to show the divergence between the MCMC distribution and the reference distribution.

**EEMD**: We compute the EEMD as the transport cost between sample and reference points in \( L_2 \) using the algorithm proposed in Bonneel et al. (2011).
## Table 6: Hyperparameters used in experiments.

| Experiment                        | Method            | burn-in steps | sampl. steps | $\gamma$ | $N$ | $\epsilon$ | $\alpha$ | num MALA steps | num flows | batch size | num train steps | lr   |
|-----------------------------------|-------------------|---------------|--------------|----------|-----|------------|----------|----------------|-----------|------------|----------------|------|
| Funnel, Banana                    | MALA              | 50k           | 1k           | .01      | 10.5| .9         | 50       | 6              | 200       | 200        | .005           |      |
|                                   | Ex$^2$MCMC        |               | 1k           | .01      | 10.5| .9         |          |                |           |            |                |      |
|                                   | FIE$^2$MCMC       |               | 1k           | .01      | 10.5| .9         |          |                |           |            |                |      |
| Bayesian Logistic Regression      | i-SIR             | 3k            | 1k           | .01      | 5   |            |          |                |           | 1          |                |      |
|                                   | MALA              | 3k            |              | .01      | 5   |            |          |                |           |            |                |      |
|                                   | Ex$^2$MCMC        | 3k            |              | .01      | 5   | .5         | .9       |                | 4         | 100        | 200            | .01  |
|                                   | FIE$^2$MCMC       | 3k            |              | .01      | 5   | .5         | .9       |                | 1         | 100        | 200            | .01  |
| Gaussian                          | i-SIR             | 15k           | 15k          | .005     | 10  | 1          | .95      |                | 0         |            |                |      |
|                                   | Ex$^2$MCMC        |               |              |          |     |            |          |                |           |            |                |      |
| Mixture of two Gaussian distributions | i-SIR             | 100           | 100          | .01      | 10  |            |          |                |           |            |                |      |
|                                   | MALA              | 100           |              | .01      |     |            |          |                |           |            |                |      |
|                                   | Ex$^2$MCMC        | 100           |              | .01      | 5   | .5         | .9       |                | 1         | 2          | 100            | 200  |
|                                   | FIE$^2$MCMC       | 100           |              | .01      | 5   | .5         | .9       |                | 1         | 2          | 100            | 200  |
| Mixture of 25 Gaussian distributions | MALA              | 1k            | 1k           | .001     | 10  | 1          | .9       |                | 1         |            |                |      |
|                                   | Ex$^2$MCMC        |               | 1k           | .001     |     |            |          |                |           |            |                |      |
| Mixture of 243 Gaussian distributions | MALA              | 1.5k          | 1.5k         | .001     | 10  | .9         | .99      |                | 1         |            |                |      |
|                                   | Ex$^2$MCMC        |               | 1.5k         | .001     |     |            |          |                |           |            |                |      |
| Swissroll                         | MALA              | 5k            | 5k           | .001     | 10  | .5         | .9       |                | 1         |            |                |      |
|                                   | Ex$^2$MCMC        |               | 5k           | .001     |     |            |          |                |           |            |                |      |
| Allen-Cahn equation               | [Gabrié et al. (2021)] |               |              | .001     | 10  | .5         | .9       |                | 10        | 2          | 100            | 10k  |
|                                   | Ex$^2$MCMC        | .001          |              |          | 10  | .5         | .9       |                | 10        | 2          | 100            | 10k  |
| Ill-conditioned Gaussian distribution | MALA              | .01           | 10           | 10       |     |            |          |                |           |            |                |      |
|                                   | Ex$^2$MCMC        | .01           |              | 10       |     |            |          |                |           |            |                |      |
|                                   | FIE$^2$MCMC       | .01           |              | 10       |     |            |          |                |           |            |                |      |

Ex$^2$MCMC: Sampling through Exploration Exploitation
ESS The ESS is computed as follows: for sample \( \{Y_i\}_{i=1}^M, Y_i \in \mathbb{R}^d \) of size \( M \), we compute ESS component-wise.

To be specific, for \( i = 1, \ldots, d \), we compute

\[
\text{ESS}_i = M_1 + \sum_{k=1}^{M} \rho_k^{(i)},
\]

where \( \rho_k^{(i)} = \frac{\text{Cov}(Y_{i+k}, Y_{i})}{\text{Var}(Y_{i})} \) is the autocorrelation at lag \( k \) for \( i \)-th component. We replace \( \rho_k \) by its sample counterpart. Then we compute

\[
\text{ESS} = d^{-1} \sum_{i=1}^{d} \text{ESS}_i.
\]

F.2 Normalizing flow RealNVP

We use RealNVP architecture (Dinh et al. (2017)) for our experiments with adaptive MCMC. The key item of RealNVP is a coupling layer, defined as transformation \( f : \mathbb{R}^D \to \mathbb{R}^D \):

\[
\begin{align*}
    y_1 : d &= x_1 : d & (136) \\
    y_{d+1 : D} &= x_{d+1 : D} \odot \exp(s(x_1 : d) + t(x_1 : d)) & (137)
\end{align*}
\]

where \( s \) and \( t \) are some functions from \( \mathbb{R}^D \) to \( \mathbb{R}^D \). It is clear then that the Jacobian of such transformation is triangular matrix with nonzero diagonal terms. We use fully connected neural networks to parameterize the functions \( s \) and \( t \).

In all experiments with normalizing flows, we use optimizer Adam (Kingma and Ba (2015)) with \( \beta_1 = 0.9, \beta_2 = 0.999 \) and weight decay 0.01 to avoid overfitting.

F.3 Adaptive strategy for tuning the stepsize in the MALA algorithm

In all our experiments, except for the Allen-Cahn equation, we use adaptive strategy for the stepsize in MALA, Ex^2MCMC and FIEx^2MCMC algorithms to keep acceptance rate close to \( \alpha = 0.5 \): we measure the average acceptance rate \( \mu_{\text{acc}} \) during sampling, and increase the stepsize if \( \mu_{\text{acc}} > \alpha \) or decrease it if \( \mu_{\text{acc}} < \alpha \) with some tolerance threshold. The scheme is described in Algorithm 8. In practice, we set tolerance \( \delta = 0.03 \), factor \( s = 1.05 \).

Algorithm 8: Adaptive strategy for stepsize.

| Input: | average acc. rate \( \mu_{\text{acc}} \), target acc. rate \( \alpha \), current stepsize \( \gamma \), tolerance \( \delta \), factor \( s \) |
| Output: | new stepsize \( \gamma' \) |
| 1 if \( \mu_{\text{acc}} - \alpha > \delta \) then |
| 2 \( \gamma' = \gamma s \) |
| 3 else if \( \alpha - \mu_{\text{acc}} > \delta \) then |
| 4 \( \gamma' = \gamma / s \) |
| 5 else |
| 6 \( \gamma' = \gamma \) |

Note that in all experiments, wherever it is not said otherwise, we set noise scale coefficient in MALA kernel to \( \sqrt{2\gamma} \), where \( \gamma \) is step size.

F.4 High-dimensional Gaussian distribution sampling

We consider the problem of sampling from a high-dimensional standard Normal distribution \( \mathcal{N}(0, \text{Id}_d) \) and the proposal distribution \( \mathcal{N}(0, 2 \text{Id}_d) \) for different problem dimension \( d \in [30; 300] \). The goal of the experiment is to show the efficiency of correlated proposals for i-SIR. We apply Ex^2MCMC algorithm with \( \epsilon = 1 \) and without rejuvenation kernel, as it is described in Algorithm 6. We compute empirical estimates of mean and variance, and report confidence intervals for them based on 20 independent runs of each algorithm. We report empirical estimates of mean and variance, and the ESS. We perform \( 10^3 \) burn-in steps for each of the algorithms, and then compute the metrics over the next \( 5 \times 10^3 \) samples. Other experimental details are provided in Table 6.
F.5 Distributions with complex geometry

In this section, we study the sampling quality from high-dimensional distributions, whose density levels have high curvature (Banana shaped and Funnel distributions, details below). With such distributions, standard MCMC algorithms like MALA or i-SIR, fail to explore fully the density support. In the examples below, we sample reference points with HMC-type algorithm NUTS (Hoffman et al. (2014)). For each of examples we produce 1000 reference points from the target distribution and consider them as reference points to compute ESTV and EMD for samples from MALA, Ex^2MCMC and FlEx^2MCMC. For every target distribution, we set proposal to be the standard normal distribution $\mathcal{N}(0, \text{Id}_d)$. As we do 50 rejuvenation steps in Ex^2MCMC and FlEx^2MCMC algorithms, we take every 50th point from MALA samples to compute ESS. We set proposal as standard normal distribution for all target distributions.

**Symmetric banana-shaped distribution** Following Haario et al. (1999), we sample from the so-called “Banana-shape” distribution. For $x \in \mathbb{R}^d$, the density of the symmetric banana-shaped distribution is given by

$$p(x) = \frac{1}{Z} \exp \left( \sum_{i=1}^{d} -\frac{(x_{2i-1} - x_{2i}^2)^2}{\nu} - \frac{(x_{2i+1} - 1)^2}{2} \right), \quad (138)$$

with $Z$ being a normalizing constant and $\nu > 0$. In our examples we set $\nu = 0.2$.

**Asymmetric banana-shaped distribution** For $x \in \mathbb{R}^d$, the density of the asymmetric banana-shaped distribution is given by

$$p(x) = \frac{1}{Z} \exp \left( \sum_{i=1}^{d} -\frac{(x_{2i-1} - x_{2i}^2)^2}{\nu} - \frac{(x_{2i+1} - 1)^2}{2} \right), \quad (139)$$

with $Z$ being a normalizing constant and $\nu > 0$. In our examples we set $\nu = 0.2$.

**Funnel distribution** For $x \in \mathbb{R}^d$ the density of the funnel distribution is given by

$$p(x) = \frac{1}{Z} \exp \left( -\frac{x_1^2}{2a^2} - e^{-2bx_1} \frac{2d}{2} \sum_{i=2}^{d} x_i^2 - \log d + 2bx_1 \right), \quad (140)$$

where $Z$ is a normalizing constant. We set $a = 1$, $b = 0.5$.

Results for both types of banana-shaped distributions and for the funnel distribution are summarized in Figure 14 and in Figure 10 respectively. We report the values of ESTV, EEMD, and ESS metrics and their dependence on the problem dimension $d$. Clearly, FlEx^2MCMC algorithm outperforms both MALA and Ex^2MCMC with independent proposals. We visualize projections of generated samples on first two coordinates for the banana-shaped and funnel examples in Figure 13 and Figure 13c respectively. They illustrate how well Ex^2MCMC and FlEx^2MCMC can explore the support compared to MALA.

F.6 Bayesian Logistic regression

The training set $\mathcal{D}$ consists of pairs $(x, y)$ where $x = (x^{(0)}, \ldots, x^{(d-1)}) \in \mathbb{R}^d$ and labels $y \in \{ -1, 1 \}$. In practice, the first coordinate of $x$ represents the bias term, i.e. we have $x^{(0)} = 1$. The likelihood for a pair is $p(y \mid x, \theta) = \logit (y \mid x, \theta)$. Given a prior distribution $p(\theta)$, we sample from the posterior distribution $p(\theta \mid \mathcal{D})$ and compute the predictive posterior $p(y \mid x, \mathcal{D}) = \int p(y \mid x, \theta)p(\theta \mid \mathcal{D})d\theta$ for $(x, y) \in \mathcal{D}^{\text{test}}$. We approximate $p(y \mid x, \mathcal{D})$ using the Monte Carlo estimate $\frac{1}{n} \sum_{i=1}^{n} p(y \mid x, \mathcal{D}, \theta_i)$, where $\theta_i$ is a sample of $p(\theta \mid \mathcal{D})$ obtained using different MCMC samplers. We take a normal prior distribution $p_0(\theta) = \mathcal{N}(0, \sigma^2 \text{Id})$ with $\sigma^2 = 20$. We present results for the following datasets:

Covertype dataset consists of 581k instances of dimension 54, and we arbitrarily classes 3 and 5 from the original 7 classes to build a binary classification task. We used 1.5k steps for burn-in and sampling phases.

EEG dataset consists of 15k instances of dimension 15. We used 1.5k steps for burn-in and sampling phases.
F.7 Mixture of two Gaussian distributions

We consider here the task of sampling from a mixture of two Gaussian distributions in different dimensions. The target density is

\[ p(x) = \frac{1}{2} g(x; \mu, \sigma^2 I_d) + \frac{1}{2} g(x; -\mu, \sigma^2 I_d), \]

where \( \mu = (1.5, 1.5, \ldots, 1.5) \in \mathbb{R}^d \), \( \sigma^2 = 1 \), and we set the proposal distribution to be \( N(0, 2 I_d) \). We perform 200 independent starts for each method for each dimension in \([10, 12, \ldots, 30]\) to compute metrics and average them across different starts. We put more details on hyperparameters used in table 6. We compute ESS per second (ESS/s) as a product of ESS and size of obtained sample (900) divided by sampling time in seconds. The result of this experiment is presented in Figure 9. Note that, despite the ESS of MALA is good, MALA always explores only one mode of the mixture.

F.8 Allen-Cahn equation

In this experiment, we consider the task of sampling from the invariant distribution of the Allen-Cahn stochastic differential equation (Allen and Cahn (1975)). We borrow the setting from the paper Gabrié et al. (2021) in which they propose to use normalizing flows to enhance MCMC sampling from distributions with meta-stable states. Allen-Cahn equation is a SDE defined in terms of a random field \( \varphi : [0, 1] \to \mathbb{R} \) that satisfies

\[ \partial_t \varphi = a \partial_s^2 \varphi + a^{-1}(\varphi - \varphi^3) + \sqrt{2\beta^{-1}} \eta(t, s), \]

where \( a > 0, \beta \) are parameters, \( s \in [0, 1] \), \( \eta \) is a spatio-temporal white noise. Following Gabrié et al. (2021), we impose boundary conditions on \( \varphi : \varphi(s = 0) = \varphi(s = 1) = 0 \). The invariant measure of the stochastic Allen-Cahn equation is the Gibbs measure associated with the Hamiltonian

\[ U(\varphi) = \beta \int_0^1 \left[ \frac{a}{2} (\partial_s \varphi)^2 + \frac{1}{4a} (1 - \varphi(s)^2)^2 \right] ds. \]
Figure 11: Symmetric banana-shaped distribution, dimension $d = 50$: projection of resulted samples on first two coordinates.

The Hamiltonian has two global well separated minima $\varphi^+$ and $\varphi^-$, making it very challenging to sample from the invariant distribution using MCMC methods. Therefore authors propose to use informed base measure with the following Hamiltonian as a proposal

$$U_B(\varphi) = \beta \int_0^1 \left[ \frac{a}{2} (\partial_s \varphi)^2 + \frac{1}{2a} \varphi^2 \right] ds . \quad (143)$$

To perform sampling, we discretize the field on a uniform grid taking 100 points of spatial variable $s$. Therefore the dimension of space we sample on is 100.

We set $a = 0.1$, $\beta = 20$. We use RealNVP with 2 flows and hidden layer size 50. Other details are provided in the Table 6.

F.9 Sampling from Ill-Conditioned Gaussian distribution

In this experiment, we consider the task of sampling from zero-centered Gaussian distribution with ill-conditioned covariance being a diagonal matrix with elements spaced log-linearly between $10^{-2}$ and $10^2$. We set the dimension to 50. Figure 12 shows autocorrelation time versus sampling iterations. We perform 50 independent runs of each method to compute autocorrelations and average across different chains. The experimental details are provided in the Table 6.

F.10 Sampling from GAN as an Energy-Based Model

In our experiments, we closely follow the exposition of (Che et al., 2020, Section 5), albeit we were not able to exactly reproduce the numbers reported in this paper. We consider the task of sampling in the latent space of learnt GAN model, as presented in Che et al. (2020). The crux of the approach is that under some assumptions on the discriminator being close to the optimal one (given by Bayes rule, see Che et al. (2020)), sampling in latent space from the distribution $p(z) = \frac{p_0(z)}{Z} \exp(\text{logit}(D(G(z))))$ is equivalent to sampling from the data distribution, where $p_0(z)$ is a proposal distribution for Generator, $G$ is a Generator and logit is the inverse sigmoid function.

F.11 GANs as energy-based models: artificial datasets

We tackle in this section the task of sampling artificial datasets. We set the proposal distribution to the standard normal distribution for all datasets.

We pay particular attention to mixture of Gaussian distributions and Swissroll examples. Similar setup is considered at (Azadi et al., 2018; Turner et al., 2019; Tanaka, 2019). We follow the same experimental setting as provided at Tanaka (2019) and Che et al. (2020, Section 5.1). The main difference compared to Tanaka (2019) is that the prior distribution $p_0$ is a multivariate Gaussian (instead of a uniform).
Table 7: GAN sampling from mixture of Gaussian distributions

| Model                  | Single chain, 25G | Multiple chains, 25G | Single chain, 243G | Multiple chains, 243G |
|------------------------|-------------------|----------------------|--------------------|-----------------------|
|                        | std, # modes, EMD | std, # modes, EMD    | std, # modes, EMD  | std, # modes, EMD     |
| Goodfellow et al. (2014) | .058, 25          | .064, 25             | .064, 34.8, 3.86   | .040, 34.8, 3.86      |
| Che et al. (2020)      | .036, 1           | 6.97, .043, 25       | .062, 1, 30.96     | .039, 98              |
| MALA                   | .050, 25          | .2, .058, 25         | .030, 5.4, 24.03   | .040, 85.8, 3.80      |
| Ex^2MCMC              | .055, 25          | .18, .056, 25        | .18, 62.9, 3.78    | .039, 89.4, 3.60      |

Figure 12: Ill-conditioned Gaussian: Autocorrelations vs. sampling iteration.

**Mixture of 25 Gaussian distributions** We build a dataset of points sampled from a mixture of 25 Gaussian distributions in dimension 2, as in Turner et al. (2019). All Gaussian distributions have scale parameter $\sigma = 0.05$, and the coordinates of the means of the Gaussian distributions $\{\mu_i\}_{i=1}^{25}$ are distributed on a uniform grid: $\mu_i \in \{-2, -1, 0, 1, 2\}^2$. From this dataset, we train a WGAN (Arjovsky et al. 2017). We use 4-layer MLP architectures with hidden dimension 128 and 256 for Generator and Discriminator respectively. We set the dimension of the latent space to $d = 2$. We perform 50 independent runs for each method and average metrics across runs. The other experimental details are presented in the Table.

**Mixture of 243 Gaussian distributions** To train GAN we construct a dataset from points distributed according to a mixture of 243 Gaussian distributions in dimension 5. All Gaussian distributions have scale parameter $\sigma = 0.05$, and the means of the Gaussian distributions $\{\mu_i\}_{i=1}^{243}$ are distributed on a uniform grid: $\mu_i \in \{-2, 0, 2\}^5$. We set the dimension of the latent space to $d = 5$. We perform 50 independent runs for each method and average metrics across runs. We use 3-layer MLP architectures for both Generator and Discriminator with hidden dimension 256 and 512 respectively.

**Swissroll** Swissroll is a popular dataset of points in $2d$ distributed along the spiral with some noise. We set the scale factor of noise $\sigma = 0.05$. We set the dimension of the latent space to $d = 2$. To train WGAN we use 3-layer MLP architectures for both Generator and Discriminator with hidden dimension 128 and 256 respectively.
Figure 13: Funnel distribution, dim 15: projection of resulted samples on first two coordinates.

Figure 14: Sampling from symmetric banana-shaped distribution.