Local-Global MCMC kernels: the best of both worlds

Sergey Samsonov[1] **Evgeny Lagutin**[1] **Marylou Gabrié**[2] **Alain Durmus**[3]

Alexey Naumov[1] **Eric Moulines**[2]
[1]HSE University [2]Ecole Polytechnique [3]ENS Paris-Saclay
{svsamsonov,elagutin,anaumov}@hse.ru
{eric.moulines,marylou.gabrie}@polytechnique.edu
alain.durmus@ens-paris-saclay.fr

Abstract

Recent works leveraging learning to enhance sampling have shown promising results, in particular by designing effective non-local moves and global proposals. However, learning accuracy is inevitably limited in regions where little data is available such as in the tails of distributions as well as in high-dimensional problems. In the present paper we study an Explore-Exploit Markov chain Monte Carlo strategy ($\mathrm{Ex}^2\mathrm{MCMC}$) that combines local and global samplers showing that it enjoys the advantages of both approaches. We prove V-uniform geometric ergodicity of $\mathrm{Ex}^2\mathrm{MCMC}$ without requiring a uniform adaptation of the global sampler to the target distribution. We also compute explicit bounds on the mixing rate of the Explore-Exploit strategy under realistic conditions. Moreover, we also analyze an adaptive version of the strategy ($\mathrm{FlEx}^2\mathrm{MCMC}$) where a normalizing flow is trained while sampling to serve as a proposal for global moves. We illustrate the efficiency of $\mathrm{Ex}^2\mathrm{MCMC}$ and its adaptive version on classical sampling benchmarks as well as in sampling high-dimensional distributions defined by Generative Adversarial Networks seen as Energy Based Models.

1 Introduction

We consider the setting where a target distribution π on a measurable space $(\mathbb{X}, \mathcal{X})$ is known up to a normalizing constant and one tries to estimate the expectations of some function $f : \mathbb{X} \to \mathbb{R}$ with respect to π. Examples include the extraction of Bayesian statistics from posterior distributions derived from observations as well as the computation of observables of a physical system $x \in \mathbb{X}$ under the Boltzmann distribution with non-normalized density $\pi(x) = \mathrm{e}^{-\beta U(x)}$ for the energy function U at the inverse temperature β.

A common strategy to tackle this estimation is to resort to Markov chain Monte Carlo algorithms (MCMCs). The MCMC approach aims to simulate a realization of a time-homogeneous Markov chain $\{Y_n,\ n \in \mathbb{N}\}$, such that the distribution of the n-th iterate Y_n with $n \to \infty$ is arbitrarily close to π, regardless of the initial distribution of Y_0. In particular, the Metropolis-Hastings kernel (MH) is the cornerstone of MCMC simulations, with a number of successful variants following the process of a *proposal* step followed by an *accept/reject* step (see e.g. [62]). In large dimensions, proposal distributions are typically chosen to generate local moves that depend on the last state of the chain in order to guarantee an admissible acceptance rate. However, local samplers suffer from long mixing times as exploration is inherently slow, and mode switching, when there is more than one, can be extremely infrequent.

On the other hand, independent proposals are able to generate more global updates, but they are difficult to design. Developments in deep generative modelling, in particular versatile autoregressive and normalising flows [39, 37, 20, 55], spurred efforts to use learned probabilistic models to improve

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

the exploration ability of MCMC kernels. Among a rapidly growing body of work, references include [36, 2, 53, 25, 33]. While these works show that global moves in a number of practical problems can be successfully informed by machine learning models, it remains the case that the acceptance rate of independent proposals decreases dramatically with dimensions – except in the unrealistic case that they perfectly reproduce the target. This is a well-known problem in the MCMC literature [12, 71, 1], and it was recently noted that deep learning-based suggestions are no exception in works focusing on physical systems [19, 46].

In this paper we focus on the benefits of combining local and global samplers. Intuitively, local steps interleaved between global updates from an independent proposal (learned or not) increase accuracy by allowing accurate sampling in tails that are not usually well handled by the independent proposal. Also, mixing time is usually improved by the local-global combination, which prevents long chains of consecutive rejections. Here we focus on a global kernel of type iterative-sampling importance resampling (i-SIR) [73, 4, 5]. This kernel uses multiple proposals in each iteration to take full advantage of modern parallel computing architectures. For local samplers, we consider common techniques such as Metropolis Adjusted Langevin (MALA) and Hamiltonian Monte Carlo (HMC). We call this combination strategy Explore-Exploit MCMC ($\mathrm{Ex^2MCMC}$) in the following.

Contributions The main contributions of the paper are as follows:

- We provide theoretical bounds on the accuracy and convergence speed of $\mathrm{Ex^2MCMC}$ strategies. In particular, we prove V-uniform geometric convergence of $\mathrm{Ex^2MCMC}$ under assumptions much milder than those required to prove uniform geometric ergodicity of the global sampler i-SIR alone.
- We provide convergence guarantees for an adaptive version of the strategy, called $\mathrm{FlEx^2MCMC}$, which involves learning an efficient proposal while sampling, as in adaptive MCMC.
- We perform a numerical evaluation of $\mathrm{Ex^2MCMC}$ and $\mathrm{FlEx^2MCMC}$ for various sampling problems, including sampling GANs as energy-based models. The results clearly show the advantages of the combined approaches compared to purely local or purely global MCMC methods.

Notations Denote $\mathbb{N}^* = \mathbb{N} \setminus \{0\}$. For a measurable function $f : \mathbb{X} \mapsto \mathbb{R}$, we define $|f|_\infty = \sup_{x \in \mathbb{X}} |f(x)|$ and $\pi(f) := \int_{\mathbb{X}} f(x)\pi(\mathrm{d}x)$. For a function $V : \mathbb{X} \mapsto [1, \infty)$ we introduce the V*-norm* of two probability measures ξ and ξ' on $(\mathbb{X}, \mathcal{X})$, $\|\xi - \xi'\|_V := \sup_{|f(x)| \le V(x)} |\xi(f) - \xi'(f)|$. If $V \equiv 1$, $\|\cdot\|_1$ is equal to the total variation distance (denoted $\|\cdot\|_{\mathrm{TV}}$).

2 Explore-Exploit Samplers

Suppose we are given a target distribution π on a measurable space $(\mathbb{X}, \mathcal{X})$ that is known only up to a normalizing constant. We will often assume that $\mathbb{X} = \mathbb{R}^d$ or a subset thereof. Two related problems are sampling from π and estimating integrals of a function $f : \mathbb{X} \mapsto \mathbb{R}$ w.r.t. π, i.e., $\pi(f)$. Among the many methods devoted to solving these problems, there is a popular family of techniques based on *Importance Sampling* (IS) and relying on independent proposals, see e.g. [1, 74]. We first give a brief overview of IS, to describe the global sampler i-SIR. We recall ergodicity results for the latter before investigating the Explore-Exploit sampling strategy which couples the global sampler with a local kernel. Then we present the main theoretical result of the paper on the ergodicity of the coupled strategy.

2.1 From Importance Sampling to i-SIR

The primary purpose of IS is to approximate integrals of the form $\pi(f)$. Its main instrument is a (known) *proposal distribution*, which we denote by $\lambda(\mathrm{d}x)$. To describe the algorithm, we assume that $\pi(\mathrm{d}x) = w(x)\lambda(\mathrm{d}x)/\lambda(w)$. In this formula, $w(x)$ is the *importance weight* function assumed to be known and positive, i.e., $w(x) > 0$ for all $x \in \mathbb{X}$, and $\lambda(w)$ is the *normalizing constant* of the distribution π. Typically $\lambda(w)$ is unknown. If we assume that π and λ have positive densities w.r.t. a common dominant measure, denoted also by π and λ respectively, then the *self-normalized importance sampling* (SNIS, see [61]) estimator of $\pi(f)$ is given by

$$\widehat{\pi}_N(f) = \textstyle\sum_{i=1}^N \omega_N^i f(X^i) \; , \tag{1}$$

where $X^{1:N} \overset{\text{i.i.d.}}{\sim} \lambda$, and $\omega_N^i = w(X^i)/\sum_{j=1}^N w(X^j)$ are the self-normalized importance weights. Note that computing ω_N^i does not require the knowledge of $\lambda(w)$. The main problem in the practical applications of IS is the choice of the proposal distribution λ. The representation $\pi(\mathrm{d}x) = w(x)\lambda(\mathrm{d}x)/\lambda(w)$ implies that the support of λ covers the support of π. At the same

Algorithm 1: Single stage of i-SIR algorithm with independent proposals

```
Procedure i-SIR $(Y_k, \lambda)$:
    Input  : Previous state $Y_k$; proposal distribution $\lambda$;
    Output : New state $Y_{k+1}$; pool of proposals $X_{k+1}^{2:N} \sim \lambda$;
    Set $X_{k+1}^1 = Y_k$, draw $X_{k+1}^{2:N} \sim \lambda$; for $i \in [N]$ do
        compute the normalized weights $\omega_{i,k+1} = w(X_{k+1}^i) / \sum_{\ell=1}^N w(X_{k+1}^\ell)$;
    Draw the proposal index $I_{k+1} \sim \mathrm{Cat}(\omega_{1,k+1}, \ldots, \omega_{N,k+1})$;
    Set $Y_{k+1} := X_{k+1}^{I_{k+1}}$.
```

time, too large variance of λ is obviously detrimental to the quality of (1). This suggests *adaptive importance sampling* techniques (discussed in [16]), which involve learning the proposal λ to improve the quality of (1). We return to this idea in section 3.

IS -based techniques can also be used to draw an (approximate) sample from π. For instance, Sampling Importance Resampling (SIR, [68]) follows the steps:

1. Draw $X^{1:N} \overset{\text{i.i.d.}}{\sim} \lambda$;
2. Compute the self-normalized importance weights $\omega_N^i = w(X^i) / \sum_{\ell=1}^N w(X^\ell)$, $i \in \{1, \ldots, N\}$;
3. Select M samples $Y^{1:M}$ from the set $X^{1:N}$ choosing X^i with probability ω_N^i with replacement.

The drawback of the procedure is that it is only asymptotically valid with $N \to \infty$. Alternatively, SIR can be repeated to define a Markov Chain as in *iterated SIR* (i-SIR), proposed in [73] and also studied in [4, 43, 42, 5]. At each iteration of i-SIR described in Algorithm 1, a candidate pool $X_{k+1}^{2:N}$ is sampled from the proposal and the next state Y_{k+1} is choosen among the candidates and the previous state $X_{k+1}^1 = Y_k$ according to the importance weights. i-SIR shares similarities with the Multiple-try Metropolis (MTM) algorithm [44], but is computationally simpler and exhibits more favorable mixing properties; see Appendix A.1. The Markov chain $\{Y_k,\ k \in \mathbb{N}\}$ generated by i-SIR has the following Markov kernel

$$\mathsf{P}_N(x, \mathsf{A}) = \int \delta_x(\mathrm{d}x^1) \sum_{i=1}^N \frac{w(x^i)}{\sum_{j=1}^N w(x^j)} \mathbb{1}_{\mathsf{A}}(x^i) \prod_{j=2}^N \lambda(\mathrm{d}x^j).$$

Interpreting i-SIR as a systematic-scan two-stage Gibbs sampler (see Appendix A.2 for more details), it follows easily that the Markov kernel P_N is reversible w.r.t. the target π, Harris recurrent and ergodic (see Theorem 5). Provided also that $|w|_\infty < \infty$, it was shown in [5] that the Markov kernel P_N is uniformly geometrically ergodic. Namely, for any initial distribution ξ on $(\mathbb{X}, \mathcal{X})$ and $k \in \mathbb{N}$,

$$\|\xi \mathsf{P}_N^k - \pi\|_{\mathrm{TV}} \leq \kappa_N^k \quad \text{with } \epsilon_N = \frac{N-1}{2\mathrm{L} + N - 2}, \mathrm{L} = |w|_\infty / \lambda(w)\ , \text{ and } \kappa_N = 1 - \epsilon_N. \tag{2}$$

We provide a simple direct proof of (2) in Appendix B.1. Yet, note that the bound (2) relies significantly on the restrictive condition that weights are uniformly bounded $|w|_\infty < \infty$. Moreover, even when this condition is satisfied, the rate κ_N can be close to 1 when the dimension d is large.[1] We illustrate this phenomenon on a Gaussian target in Appendix E.2 Figure 7 with an experiment that also contrasts the degradation as dimension grows of the purely global sampler with the robustness of the local-global kernels analyzed in the next section.

2.2 Coupling with local kernels: $\mathrm{Ex}^2\mathrm{MCMC}$

After each i-SIR step, we apply a local MCMC kernel R (rejuvenation kernel), with an invariant distribution π. We call this startegy $\mathrm{Ex}^2\mathrm{MCMC}$ because it combines steps of exploration by i-SIR and steps of exploitation by the local MCMC moves. The resulting algorithm, formulated in Algorithm 2, defines a Markov chain $\{Y_j,\ j \in \mathbb{N}\}$ with Markov kernel $\mathsf{K}_N(x, \cdot) = \mathsf{P}_N \mathsf{R}(x, \cdot) = \int \mathsf{P}_N(x, \mathrm{d}y) \mathsf{R}(y, \cdot)$.

We now present the main theoretical result of this paper on the properties of $\mathrm{Ex}^2\mathrm{MCMC}$. Under rather weak conditions, provided that R is geometrically regular (see [21, Chapter 14]), it is possible

[1] Indeed, consider a simple scenario $\pi(x) = \prod_{i=1}^d p(x_i)$ and $\lambda(x) = \prod_{i=1}^d q(x_i)$ for some densities $p(\cdot)$ and $q(\cdot)$ on $\mathbb{R}$. Then it is easy to see that $\mathrm{L} = (\sup_{y \in \mathbb{R}} p(y)/q(y))^d$ grows exponentially with d.

Algorithm 2: Single stage of $\mathrm{Ex}^2\mathrm{MCMC}$ algorithm with independent proposals

```
Procedure Ex$^2$MCMC $(Y_k, \lambda, \mathsf{R})$:
    Input  : Previous state $Y_k$; proposal distribution $\lambda$; rejuvenation kernel $\mathsf{R}$;
    Output : New sample $Y_{k+1}$; pool of proposals $X_{k+1}^{2:N} \sim \lambda$;
    $Z_{k+1}$ , $X_{k+1}^{2:N} = \text{i-SIR}(Y_k, \lambda)$;
    Draw $Y_{k+1} \sim \mathsf{R}(Z_{k+1}, \cdot)$.
```

to establish that $\mathrm{Ex}^2\mathrm{MCMC}$ remains V-uniformly geometrically ergodic even if the weight function $w(x)$ is unbounded.

Definition 1 (V-Geometric Ergodicity)**.** *A Markov kernel* Q *with invariant probability measure* π *is* V*-geometrically ergodic if there exist constants* $\rho \in (0,1)$ *and* $M < \infty$ *such that, for all* $x \in \mathbb{X}$ *and* $k \in \mathbb{N}$, $\|\mathsf{Q}^k(x,\cdot) - \pi\|_V \leq M\,\{V(x) + \pi(V)\}\rho^k$.

In particular, V-geometric ergodicity ensures that the distribution of the k-th iterate 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 $V(x)$. Denote by $\mathrm{Var}_\lambda[w] = \int \{w(x) - \lambda(w)\}^2 \lambda(\mathrm{d}x)$ the variance of the importance weight functions under the proposal distribution and consider the following assumptions:

A1. *(i)* R *has* π *as its unique invariant distribution; (ii) There exists a function* $V \colon \mathbb{X} \to [1,\infty)$*, such that for all* $r \geq r_{\mathsf{R}} > 1$ *there exist* $\lambda_{\mathsf{R},r} \in [0,1)$, $\mathsf{b}_{\mathsf{R},r} < \infty$*, such that* $\mathsf{R}V(x) \leq \lambda_{\mathsf{R},r} V(x) + \mathsf{b}_{\mathsf{R},r} 1_{\mathsf{V}_r}$*, where* $\mathsf{V}_r = \{x \colon V(x) \leq r\}$*;*

A2. *(i) For all* $r \geq r_{\mathsf{R}}$, $w_{\infty,r} := \sup_{x \in \mathsf{V}_r} \{w(x)/\lambda(w)\} < \infty$ *and (ii)* $\mathrm{Var}_\lambda[w]/\{\lambda(w)\}^2 < \infty$.

A1-(ii) states that R satisfies a Foster-Lyapunov drift condition for V. This condition is fulfilled by most classical MCMC kernels - like Metropolis-Adjusted Langevin (MALA) algorithm or Hamiltonian Monte Carlo (HMC), typically under tail conditions for the target distribution; see [63, 22], and [21, Chapter 2] with the references therein. **A**2-(i) states that the (normalized) importance weights $w(\cdot)/\lambda(w)$ are upper bounded on level sets of V_r. This is a mild condition: if $\mathbb{X} = \mathbb{R}^d$, and V is norm-like, then the level sets V_r are compact and $w(\cdot)$ is bounded on V_r as soon as π and λ are positive and continuous. **A**2-(ii) states that the variance of the importance weights is bounded; note that this variance is also equal to the χ^2-distance between the proposal and the target distributions which plays a key role in the non-asymptotic analysis of the performance of IS methods [1, 70].

Theorem 2. *Assume* ***A****1 and* ***A****2. Then, for all* $x \in \mathbb{X}$ *and* $k \in \mathbb{N}$,

$$\|\mathsf{K}_N^k(x,\cdot) - \pi\|_V \leq c_{\mathsf{K}_N} \{\pi(V) + V(x)\} \tilde{\kappa}_{\mathsf{K}_N}^k \ , \tag{3}$$

where the constant c_{K_N}, $\tilde{\kappa}_{\mathsf{K}_N} \in [0,1)$ *are given in the proof. In addition,* $c_{\mathsf{K}_N} = c_{\mathsf{K}_\infty} + O(N^{-1})$ *and* $\tilde{\kappa}_{\mathsf{K}_\infty} = \tilde{\kappa}_{\mathsf{K}_N} + O(N^{-1})$ *with explicit expressions provided in* (13)*.*

The proof of Theorem 2 is provided in Appendix B.2. We stress that in many situations, the mixing rate $\tilde{\kappa}_{\mathsf{K}_N}$ of the $\mathrm{Ex}^2\mathrm{MCMC}$ Markov Kernel K_N is significantly better than the corresponding mixing rate of the local kernel R, provided N is large enough. This is due to the fact that assumptions **A**1 and **A**2 do not require to identify the small sets of the rejuvenation kernel R (see [21, Definition 9.3.5]). At the same time, the quantitative bounds on the mixing rates relies on the constants appearing in the small set condition, see [21, Theorem 19.4.1]. Focusing on MALA (see, e.g. [66]) as the rejuvenation kernel R we detail bounds in Appendix C and prove in Theorem 20 that the ratio of mixing times of K_N is typically very favorable compared to MALA provided that N is large enough.

3 Adaptive version: $\mathrm{FlEx}^2\mathrm{MCMC}$

The performance of proposal-based samplers depends on the distribution of importance weights which is related to the similarity of the proposal and target distributions[2]. Therefore, yet another strategy to improve sampling performance is to select the proposal distribution λ from a family of parameterized distributions $\{\lambda_\theta\}$ and fit the parameter $\theta \in \Theta = \mathbb{R}^q$ to the target π, for example, by minimizing a Kullback-Leibler divergence (KL) [57, 2, 50] or matching moments [59]. In *adaptive*

[2] more specifically, it depends on the the quantities appearing in **A**2, namely, the maximum of the importance weight on a level set of the drift function for the local kernel R and the variance of the importance weights under the proposal

MCMCs, parameter adaptation is performed along the MCMC run [6, 9, 64]. In this section we propose an adaptive version of Ex^{2}MCMC, which we call FlEx^{2}MCMC.

Normalizing flow proposal. A flexible way to parameterize proposal distributions is to combine a tractable distribution φ with an invertible parameterized transformation. Let $T : \mathbb{X} \mapsto \mathbb{X}$ be a C^1 diffeomorphism. We denote by $T\#\varphi$ the push-forward of φ under T, that is, the distribution of $X = T(Z)$ with $Z \sim \varphi$. Assuming that φ has a p.d.f. (also denoted φ), the corresponding push-forward density (w.r.t. the Lebesgue measure) is given by $\lambda_T(y) = \varphi\big(T^{-1}(y)\big)\, \mathrm{J}_{T^{-1}}(y)$, where J_T denotes the Jacobian determinant of T. The parameterized family of diffeomorphisms $\{T_\theta\}_{\theta\in\Theta}$ defines a family of distributions $\{\lambda_{T_\theta}\}_{\theta\in\Theta}$, denoted for simplicity as $\{\lambda_\theta\}_{\theta\in\Theta}$. This construction is called a *normalizing flow* (NF) and a great deal of work has been devoted to ways of parameterizing invertible flows T_θ with neural networks; see [40, 55] for reviews.

Simultaneous learning and sampling. As with adaptive MCMC methods, the parameters of a NF proposal are learned for the global proposal during sampling, see also [25]. We work with M copies of the Markov chains $\{(Y_k[j], X_k^{1:N}[j])\}_{k\in\mathbb{N}^*}$ indexed by $j \in \{1, \ldots, M\}$. At each step $k \in \mathbb{N}^*$, each copy is sampled as in Ex^{2}MCMC using the NF proposal, independently from the other copies, but conditionally to the the current value of the parameters θ_{k-1}. We then adapt the parameters by taking steps of gradient descent on a convex combination of the *forward* KL, $\mathrm{KL}(\pi||\lambda_\theta) = \int \pi(x)\log(\pi(x)/\lambda_\theta(x))\mathrm{d}x$ and the backward KL $\mathrm{KL}(\lambda_\theta||\pi) = \int \lambda_\theta(x)\log(\pi(x)/\lambda_\theta(x))\mathrm{d}x = \int \varphi(z)\log w_\theta \circ T_\theta(z)\mathrm{d}z$. Let $\{\gamma_k,\ k \in \mathbb{N}\}$ be a sequence of nonnegative stepsizes and $\{\alpha_k,\ k \in \mathbb{N}\}$ be a nondecreasing sequence in $[0, 1]$ with $\alpha_\infty = \lim_{k\to\infty} \alpha_k$. The update rule is $\theta_k = \theta_{k-1} + \gamma_k M^{-1} \sum_{j=1}^M H(\theta_{k-1}, X_k^{1:N}[j], Z_k^{2:N}[j])$ where $H(\theta, x^{1:N}, z^{2:N}) = \alpha_k H^f(\theta, x^{1:N}) + (1-\alpha_k)H^b(\theta, z^{2:N})$ with

$$H^f(\theta, x^{1:N}) = \sum\nolimits_{\ell=1}^N \frac{w_\theta(x^\ell)}{\sum_{i=1}^N w_\theta(x^i)} \nabla_\theta \log \lambda_\theta(x^\ell)\ , \quad w_\theta(x) = \pi(x)/\lambda_\theta(x)\ , \tag{4}$$

$$H^b(\theta, z^{2:N}) = -\frac{1}{N-1}\sum\nolimits_{\ell=2}^N \{\nabla_\theta \log \pi \circ T_\theta(z^\ell) + \nabla_\theta \log \mathrm{J}_{T_\theta}(z^\ell)\}\ . \tag{5}$$

Note that we use a Rao-Blackwellized estimator of the gradient of the forward KL (4) where we fully recycle all the N candidates sampled at each iteration of i-SIR. The quality of this estimator is expected to improve along the iterations k of the algorithm as the variance of importance weights decreases as the proposal improves. Note also that using only gradients from the backward KL (5) is prone to mode-collapse [57, 54, 50, 25], hence the need for also using gradients from the forward KL $H^f(\theta, x^{1:N})$, which requires the simultaneous sampling from π. See also Appendix E.5 for further discussions. The FlEx^{2}MCMC algorithm is summarized in Algorithm 3.

Since the parameters of the Markov kernel θ_k are updated using samples $X_k^{1:N}$ from the chain, $((Y_k, X_k^{1:N}))_{k\in\mathbb{N}}$ is no longer Markovian. This type of problems has been considered in [48, 13, 30, 7] and to prove convergence of the strategy we need to strengthen assumptions compared to the previous section.

A3. *There exists a function* $W : \mathbb{X} \to \mathbb{R}_+$ *such that* $\varphi(W^2) = \int W^2(z)\varphi(\mathrm{d}z) < \infty$*, and a constant* $L < \infty$ *such that, for all* $\theta, \theta' \in \Theta$ *and* $z \in \mathbb{X}$*,* $\|\nabla_\theta \log \pi \circ T_\theta(z) - \nabla_\theta \log \pi \circ T_{\theta'}(z)\| \leq L\|\theta - \theta'\|W(z)$ *and* $\|\nabla_\theta \log \mathrm{J}_{T_\theta}(z) - \nabla_\theta \log \mathrm{J}_{T_{\theta'}}(z)\| \leq L\|\theta - \theta'\|W(z)$.

A4. *(i) For all* $d \geq d_{\mathsf{R}}$*,* $w_{\infty,d} = \sup_{\theta\in\Theta}\sup_{x\in\mathsf{V}_d} w_\theta(x)/\lambda_\theta(w_\theta) < \infty$ *and (ii)* $\sup_{\theta\in\Theta}\mathrm{Var}_\varphi(w_\theta \circ T_\theta)/\{\lambda_\theta(w_\theta)\}^2 < \infty$.

A3 is a continuity condition on the NF push-forward density w.r.t. its parameters θ. **A**4 implies that the Markov kernel $\mathsf{K}_{N,\theta} = \mathsf{P}_{N,\theta}\mathsf{R}$ satisfies a drift and minorization condition uniform in θ.

Theorem 3 (simplified). *Assume* ***A*** *1-****A*** *3-****A*** *4 and that* $\sum_{k=0}^\infty \gamma_k = \infty$, $\sum_{k=0}^\infty \gamma_k^2 < \infty$ *and* $\lim_{k\to\infty}\alpha_k = \alpha_\infty$*. Then, w.p. 1, the sequence* $\{\theta_k,\ k \in \mathbb{N}\}$ *converges to the set* $\{\theta \in \Theta, 0 = \alpha_\infty \nabla\mathrm{KL}(\pi||\lambda_\theta) + (1-\alpha_\infty)\nabla\mathrm{KL}(\lambda_\theta||\pi)\}$.

Theorem 3 proves the convergence of the learning of parameters θ to a stationary point of the loss. The proof is postponed to Appendix D. Note that once the proposal learning has converged, FlEx^{2}MCMC boils back to Ex^{2}MCMC with a fixed learned proposal. Our experiments show that adaptivity can significantly speed up mixing for i-SIR, especially for distributions with complex geometries and that the addition of a rejuvenation kernel further improves samples quality.

Algorithm 3: Single stage of FlEx^{2}MCMC. Steps of Ex^{2}MCMC use the NF proposal with parameters θ_k. Step 4 updates the parameters using the gradient estimate obtained from all the chains.

```
Input  : weights $\theta_k$, batch $Y_k[1:M]$
Output : new weights $\theta_{k+1}$, batch $Y_{k+1}[1:M]$
1 for $j \in [M]$ do
2     $Y_{k+1}[j] = \mathrm{Ex}^2\mathrm{MCMC}\ (Y_k, T_{\theta_k}\#\varphi, \mathsf{R})$
3 Draw $Z[1:M] \sim \varphi$.
4 Update $\theta_k = \theta_{k-1} + \gamma_k M^{-1} \sum_{j=1}^{M} H(\theta_{k-1}, X_k^{1:N}[j], Z_k^{2:N}[j])$
```

4 Related Work

The possibility to parametrize very flexible probabilistic models with neural networks thanks to deep learning has rekindled interest in adapting MCMC kernels; see e.g. [72, 36, 2, 53, 33]. While significant performance gain were found in problems of moderate dimensions, these learning-based methods were found to suffer from increasing dimensions as fitting models accurately becomes more difficult [19, 46]. Similarly to FlEx^{2}MCMC, a few work proposed adaptive algorithms that alternates between global and local MCMC moves to ensure ergodicity without requiring a perfect learning of the proposal[59, 25]. More precisely, [59] focused on multimodal distributions and analysed a mode jumping algorithm using proposals parametrized as mixture of simple distributions. While [25], closer to this work, introduced a combination of a local and a global sampler leveraging normalizing flows with a more classical choice for the global sampler: independent Metropolis-Hasting (IMH) instead of i-SIR. The present work builds on these previous propositions of combinations of local and global sampler by clarifying the reasons of their effectiveness through entirely novel detailed mathematical and empirical analyses. We chose to focus on i-SIR with an adaptive proposal as the global sampler since (i) the learning component allows to tackle high-dimensional targets, (ii) theoretical guarantees can be obtained for i-SIR whereas IMH is more difficult to analyze, (iii) IMH and i-SIR (as a multiple-try MCMC) are expected to have similar performances for comparable computational budget [11] but IMH is sequential where i-SIR can be parallelized by increasing the number N of proposals per iteration.

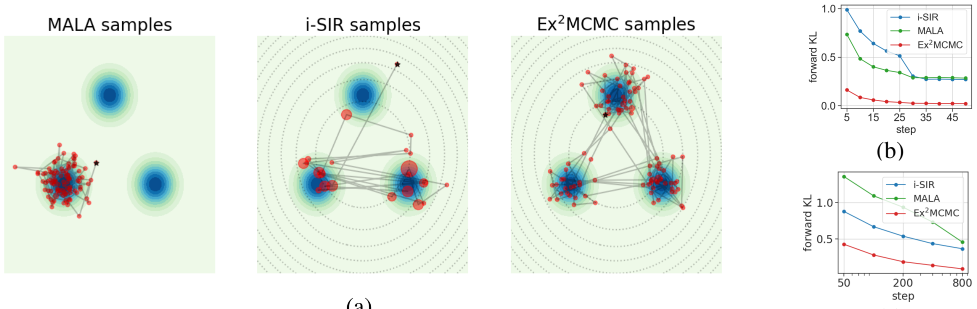
Another line of work exploits both normalizing flows and common local MCMC kernels for sampling [57, 36, 54, 77], yet following the different paradigm of using the flow as a reparametrization map, a method sometimes referred to as neural transport: the flow T is trained to transport a simple distribution φ near π, which is equivalent to bringing $T^{-1}\#\pi$ (the pushforward of the original target distribution π by the inverse flow T^{-1}) close to φ. If φ is simple enough to be efficiently sampled by local samplers, the hope is that local samplers can also obtain high-quality samples of $T^{-1}\#\pi$ – samples which can be transported back through T to obtain samples of π. This method attempts to reparametrize the space to disentangle problematic geometries for local kernels. Yet, it is unclear what will happen in the tails of the distribution for which the flow is likely poorly learned. Furthermore, in order to derive an ergodicity theory for these transported samplers, [57] necessitated substantial constraints on maps (see section 2.2.2.).

5 Numerical experiments

We provide the code to reproduce the experiments below at `https://github.com/svsamsonov/ex2mcmc_new`.

5.1 Synthetic examples

Multimodal distributions. Let us start with a toy example highlighting differences between purely global i-SIR, purely local MALA and Ex^{2}MCMC combining both. We consider sampling from a mixture of 3 equally weighted Gaussians in dimension $d = 2$. In Figure 1a, we compare single chains produced by each algorithms. The global proposal is a wide Gaussian, with pools of $N = 3$ candidate. The MALA stepsize is chosen to reach a target acceptance rate of ~ 0.67. This simple experiment illustrates the drawbacks of both approaches: i-SIR samples reach all the modes of the target, but the chains often get stuck for several steps hindering variability. MALA allows for better local exploration of each particular mode, yet it fails to cover all the target support. Meanwhile, Ex^{2}MCMC retains the benefits of both methods, combining the i-SIR-based global exploration with MALA-based local exploration.



(a)

(b)

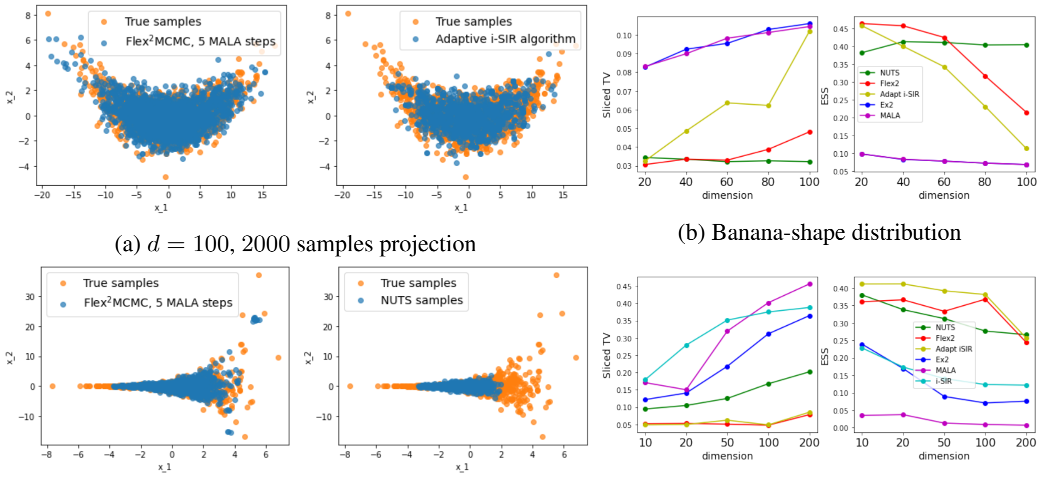
(c)

Figure 1: (a) – Single chain mixing visualization. – Blue color levels represent the target 2d density. Random chain initialization is noted in black, 100 steps are plotted per sampler: the size of each red dot corresponds to the number of consecutive steps the walkers remains at a given location. Note that the variance of the global proposal (dotted countour lines) should be relatively large to cover well all the modes. (b - c) – Inhomogeneous 2d Gaussian mixture. – Quantitative analysis during burn-in of parallel chains (b, $M = 500$ chains KDE) and for after burn-in for single chains statistics (c, $M = 100$ average).

In larger dimensions, an adaptive proposal is necessary. In Appendix E.5 we show that $\mathrm{FlEx^2MCMC}$ can mix between modes of a $50d$ Gaussian mixture, provided that the rough location of all the modes is known and used to initialize walkers. We also stress the robustness of the on-the-fly training exploiting running MCMC chains to evaluate the forward KL term of the loss.

To illustrate further the performance of the combined kernel, we keep the $2d$ target mixture model yet assigning the uneven weights $(2/3, 1/6, 1/6)$ to the 3 modes. We start M chains drawing from the initial distribution $\xi \sim \mathcal{N}(0, 4\,\mathrm{I}_d)$ and use the same hyper-parameters as above. In Figure 1b we provide a simple illustration to the statement (2) and Theorem 2, namely we compare the target density to the instantaneous distributions for each sampler propagating ξ during burn-in steps. As MALA does not mix easily between modes, the different statistical weights of the different modes can hardly be rendered in few iterations and KL and TV distances stalls after a few iterations. i-SIR can visit the different modes, yet it does not necessarily move at each step which slows down its covering of the modes full support, which again shows in the speed of decrease of the TV and KL. Overcoming both of these shortcomings, $\mathrm{Ex^2MCMC}$ instantaneous density comes much closer to the target. Finally, Figure 1c evaluates the same metrics yet for the density estimate obtained with single chain samples after burn-in. Results demonstrate once again the superiority of $\mathrm{Ex^2MCMC}$. Further details on these experiments can be found in Appendix E.3.

Distributions with complex geometry. Next, we turn to highly anisotropic distributions in high dimensions. Following [52] and [32], we consider the *funnel* and the *banana-shape* distributions. We remind densities in Appendix E.6 along with providing experiments details. For $d \in [10; 200]$, we run i-SIR, MALA, $\mathrm{Ex^2MCMC}$, $\mathrm{FlEx^2MCMC}$, adaptive i-SIR (using the same proposal as $\mathrm{FlEx^2MCMC}$, but without interleaved local steps) and the versatile sampler NUTS [35] as a baseline. Here the parameter adaptation for $\mathrm{FlEx^2MCMC}$ is performed in a pre-run and parameters are frozen before sampling. For the adaptive samplers, a simple RealNVP-based normalizing flow [20] is used such that total running times, including training, are comparable with NUTS. For $\mathrm{Ex^2MCMC}$ and i-SIR the global proposal is a wide Gaussian with a pool of $N = 2000$ candidates drawn at each iteration. For MALA we tune the step size in order to keep acceptance rate approximately at 0.5. We report the average sliced TV distance and ESS in Figure 2 (see Appendix E.1 for metrics definition). In most cases, $\mathrm{FlEx^2MCMC}$ is the most reliable algorithm. The only exception is at very high dimension for the banana where NUTS performs the best: in this case, tuning the flow to learn tails in high-dimension faithfully was costly such that we proceeded to an early stopping to maintain comparability with the baseline. Remarkably, $\mathrm{FlEx^2MCMC}$ compensates significantly for the imperfect flow training, improving over adaptive-i-SIR, but NUTS eventually performs better. Conversely, for the funnel, most of the improvement comes from well-trained proposal flow, leading to similar behaviors of adaptive i-SIR and $\mathrm{FlEx^2MCMC}$, while both algorithms clearly outperforms NUTS in terms of metrics.



(a) $d = 100$, 2000 samples projection

(b) Banana-shape distribution

(c) $d = 100$, 1000 samples projection

(d) Neal's funnel

Figure 2: Anisotropic Funnel and Banana-shape distributions – (a) and (b) visualize samples projected onto the first 2 coordinates of tested algorithms (blue) versus true samples obtained by reparametrization (orange). (c) and (d) compare Sliced Total Variation and Effective Sample Size as a function of dimension. i-SIR is removed from (b) as corresponding metrics for $d > 20$ are significantly worse.

5.2 Sampling from GANs as Energy-based models (EBMs)

Generative adversarial networks (GANs [27]) are 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 generates an observation $G(z) \in \mathbb{R}^{\mathsf{D}}$ in the observation space. The discriminator takes a sample in the observation space and aims to discriminate between true examples and false examples produced by the generator. Recently, it has been advocated to consider GANs as Energy-Based Models (EBMs) [75, 17]. Following [17], we consider the EBM model induced by the GAN in latent space. Recall that an EBM is defined by a Boltzmann distribution $p(z) = \mathrm{e}^{-E(z)} / \mathrm{Z}$, $z \in \mathbb{R}^d$, where $E(z)$ is the energy function and Z is the normalizing constant. Note that Wasserstein GANs also allow for an energy-based interpretation (see [17]), although the interpretation of the discriminator in this case is different. The energy function is given by

$$E_{JS}(z) = -\log p_0(z) - \operatorname{logit}\big(D(G(z))\big) \,, \quad E_W(z) = -\log p_0(z) - D(G(z)), \quad z \in \mathbb{R}^d \,, \quad (6)$$

for the vanilla Jensen-Shannon and Wasserstein GANs, respectively. Here $\operatorname{logit}(y)$, $y \in (0, 1)$ is the inverse of the sigmoid function and $p_0(z) = \mathcal{N}(0, \mathrm{I}_d)$.

MNIST results. We consider a simple Jensen-Shannon GAN model trained on the MNIST dataset with latent space dimension $d = 2$. We compare samples obtained by i-SIR, MALA, and $\mathrm{Ex}^2\mathrm{MCMC}$ from the energy-based model associated with $E_{JS}(z)$, see (6). We use a wide normal distribution as the global proposal for i-SIR and $\mathrm{Ex}^2\mathrm{MCMC}$, and pools of candidates at each iteration $N = 10$. The step-size of MALA is tuned to keep an acceptance rate ~ 0.5. We visualize chains of 100 steps in the latent space obtained with each method in Figure 3. Note that the poor agreement between the proposal and the landscape makes it difficult for i-SIR to accept from the proposal and for MALA to explore many modes of the latent distribution, as shown in Figure 3. $\mathrm{Ex}^2\mathrm{MCMC}$ combines effectively global and local moves, encouraging better diversity associated with a better mixing time. The images corresponding to the sampled latent space locations are displayed in Figure 4 and reflect the diversity issue of MALA and i-SIR. Further details and experiments are provided in Appendix E.7.1, including similar results for WGAN-GP [31] and the associated EBM $E_W(z)$.

Cifar-10 results. We consider two popular architectures trained on Cifar-10, DC-GAN [60] and SN-GAN [49]. In both cases the dimension of the latent space equals $d = 128$. Together with the non-trivial geometry of the corresponding energy landscapes, the large dimension makes sampling with NUTS unfeasible in terms of computational time. We perform sampling from mentioned GANs as energy-based models using i-SIR, MALA, $\mathrm{Ex}^2\mathrm{MCMC}$, and $\mathrm{FlEx}^2\mathrm{MCMC}$. In i-SIR and $\mathrm{Ex}^2\mathrm{MCMC}$ we use the prior $p_0(z)$ as a global proposal with a pool of $N = 10$ candidates. For $\mathrm{FlEx}^2\mathrm{MCMC}$ we perform training and sampling simultaneously. Implementation details are

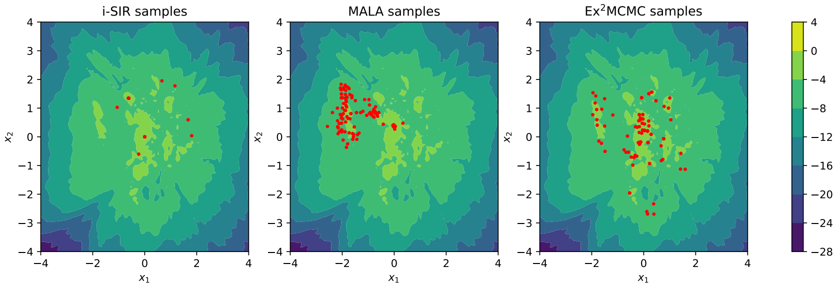
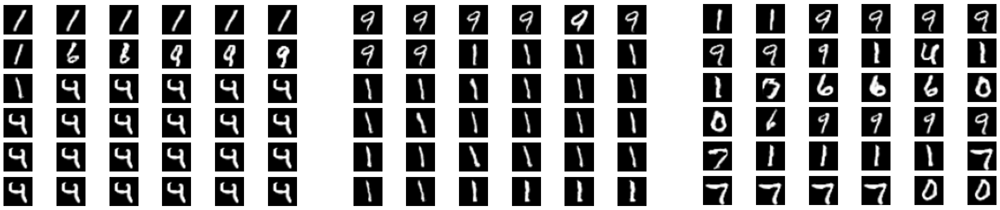


Figure 3: MNIST energy landscape and single chain latent samples visualizations.



(a) i-SIR samples

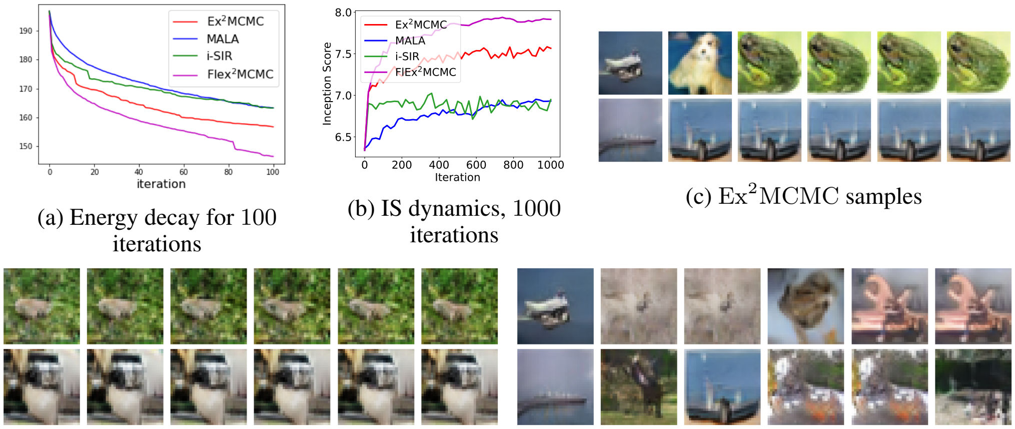
(b) MALA samples

(c) Ex^{2}MCMC samples

Figure 4: MNIST samples visualization. – Single chains run, sequential steps.

provided in Appendix E.7.2. To evaluate sampling quality, we report the values of the energy function $E(z)$, averaged over 500 independent runs of each sampler. We also visualize the inception score (IS) dynamics calculated over 10000 independent trajectories. We present the results in Figure 5 together with the images produced by each sampler. Note that $\mathrm{Ex^2MCMC}$ and $\mathrm{FlEx^2MCMC}$ reach low level of energies faster than other methods, and reach high IS samples in a limited number of iterations. Visualizations indicate that MALA is unlikely to escape the mode of the distribution $p(z)$ it started from, while i-SIR and $\mathrm{Ex^2MCMC/FlEx^2MCMC}$ better explores the target support. However, global move appear to become more rare after some number of iterations for $\mathrm{Ex^2MCMC/FlEx^2MCMC}$, which then exploit a particular mode with MALA steps. We here hit the following limitation: i-SIR remains at relatively high-energies, failing to explore well modes basins but still accepting global moves, while $\mathrm{Ex^2MCMC/FlEx^2MCMC}$ explores well modes basins but eventually remains trapped. We predict that improving further the quality of the $\mathrm{FlEx^2MCMC}$ proposal by scaling the normalizing flow architecture would allow for more global moves.See Appendix E.7.2 for additional experiments (including ones with SN-GAN), FID dynamics, and visualizations.

6 Conclusions and further research directions



(a) Energy decay for 100 iterations

(b) IS dynamics, 1000 iterations

(c) Ex^{2}MCMC samples

(d) MALA samples

(e) i-SIR samples

Figure 5: Cifar-10 energy and sampling results with DC-GAN architecture. Along the horizonthal lines we visualize each 10th sample from a single trajectory.

The present paper examines the benefits of combining local and global samplers. From a theoretical point of view, we show that global samplers are more robust when coupled with local samplers. Namely, a V-geometric ergodicity is obtained for the $\mathrm{Ex}^2\mathrm{MCMC}$ kernel under minimal assumptions. Meanwhile, the global samplers drives exploration when properly adjusted. Therefore, we also describe the adaptive version $\mathrm{FlEx}^2\mathrm{MCMC}$ of the strategy involving the learning of a global proposal parametrized by a normalizing flow. We also check for the learning convergence along the adaptive MCMC run. Finally, a series of numerical experiments confirms the superiority of the strategy, including the high-dimensional examples. While the startegy was described and analyzed for the $\mathrm{i\text{-}SIR}$ global kernel, we note that it would be possible to extend the theory to other independent global samplers. We expect that the benefit of the combination would remain. Further studies of $\mathrm{FlEx}^2\mathrm{MCMC}$, in particular the derivation of its mixing rate, is an interesting direction for future work.

Acknowledgement

The article was partly prepared within the framework of the HSE University Basic Research Program. This work was partly supported by SCAI Project-ANR-19-CHIA-0002. M.G. acknowledges funding from Hi!Paris. A.D. and E.M. acknowledges support from the Lagrange Mathematics and Computing Research Center.

References

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

[2] M. Albergo, G. Kanwar, and P. Shanahan. Flow-based generative models for Markov chain Monte Carlo in lattice field theory. *Physical Review D*, 100(3):034515, 2019.

[3] C. Andrieu. On random-and systematic-scan samplers. *Biometrika*, 103(3):719–726, 2016.

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

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

[6] C. Andrieu and É. Moulines. On the ergodicity properties of some adaptive mcmc algorithms. *The Annals of Applied Probability*, 16(3):1462–1505, 2006.

[7] C. Andrieu, É. Moulines, and P. Priouret. Stability of stochastic approximation under verifiable conditions. *SIAM Journal on control and optimization*, 44(1):283–312, 2005.

[8] C. Andrieu, V. B. Tadić, and M. Vihola. On the stability of some controlled markov chains and its applications to stochastic approximation with markovian dynamic. *The Annals of Applied Probability*, 25(1):1–45, 2015.

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

[10] C. Andrieu and M. Vihola. Markovian stochastic approximation with expanding projections. *Bernoulli*, 20(2):545–585, 2014.

[11] M. Bédard, R. Douc, and E. Moulines. Scaling analysis of multiple-try MCMC methods. *Stochastic Processes and their Applications*, 122(3):758–786, 2012.

[12] T. Bengtsson, P. J. Bickel, and B. Li. Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. *arXiv: Statistics Theory*, pages 316–334, 2008.

[13] A. Benveniste, M. Métivier, and P. Priouret. *Adaptive algorithms and stochastic approximations*, volume 22 of *Applications of Mathematics (New York)*. Springer-Verlag, Berlin, 1990. Translated from the French by Stephen S. Wilson.

[14] N. Bonneel, M. Van De Panne, S. Paris, and W. Heidrich. Displacement interpolation using lagrangian mass transport. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, pages 1–12, 2011.

[15] V. S. Borkar. *Stochastic approximation: a dynamical systems viewpoint*, volume 48. Springer, 2009.

[16] M. F. Bugallo, V. Elvira, L. Martino, D. Luengo, J. Miguez, and P. M. Djuric. Adaptive importance sampling: The past, the present, and the future. *IEEE Signal Processing Magazine*, 34(4):60–79, 2017.

[17] T. Che, R. Zhang, J. Sohl-Dickstein, H. Larochelle, L. Paull, Y. Cao, and Y. Bengio. Your GAN is Secretly an Energy-based Model and You Should Use Discriminator Driven Latent Sampling. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12275–12287. Curran Associates, Inc., 2020.

[18] M.-F. Chen and F.-Y. Wang. Estimation of spectral gap for elliptic operators. *Trans. Amer. Math. Soc.*, 349(3):1239–1267, 1997.

[19] L. Del Debbio, J. Marsh Rossney, and M. Wilson. Efficient modeling of trivializing maps for lattice $\phi 4$ theory using normalizing flows: A first look at scalability. *Physical Review D*, 104(9), 2021.

[20] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[21] R. Douc, E. Moulines, P. Priouret, and P. Soulier. *Markov chains*. Springer Series in Operations Research and Financial Engineering. Springer, Cham, 2018.

[22] A. Durmus and E. Moulines. On the geometric convergence for MALA under verifiable conditions. 2022.

[23] A. Eberle. Reflection couplings and contraction rates for diffusions. *Probab. Theory Related Fields*, pages 1–36, 2015.

[24] D. L. Ermak. A computer simulation of charged particles in solution. i. technique and equilibrium properties. *The Journal of Chemical Physics*, 62(10):4189–4196, 1975.

[25] M. Gabrié, G. M. Rotskoff, and E. Vanden-Eijnden. Adaptive Monte Carlo augmented with normalizing flows. *Proceedings of the National Academy of Sciences*, 119(10), mar 2022.

[26] M. Girolami and B. Calderhead. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

[27] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, Cambridge, MA, USA, 2014. MIT Press.

[28] U. Grenander. Tutorial in pattern theory. Division of Applied Mathematics, Brown University, Providence, 1983.

[29] U. Grenander and M. I. Miller. Representations of knowledge in complex systems. *J. Roy. Statist. Soc. Ser. B*, 56(4):549–603, 1994. With discussion and a reply by the authors.

[30] M. G. Gu and F. H. Kong. A stochastic approximation algorithm with markov chain monte-carlo method for incomplete data estimation problems. *Proceedings of the National Academy of Sciences*, 95(13):7270–7274, 1998.

[31] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[32] H. Haario, E. Saksman, and J. Tamminen. Adaptive proposal distribution for random walk metropolis algorithm. *Computational Statistics*, 14(3):375–395, 1999.

[33] D. C. Hackett, C.-C. Hsieh, M. S. Albergo, D. Boyda, J.-W. Chen, K.-F. Chen, K. Cranmer, G. Kanwar, and P. E. Shanahan. Flow-based sampling for multimodal distributions in lattice field theory. *arXiv preprint*, 2107.00734, 2021.

[34] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

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

[36] M. D. Hoffman, P. Sountsov, J. V. Dillon, I. Langmore, D. Tran, and S. Vasudevan. NeuTralizing Bad Geometry in Hamiltonian Monte Carlo Using Neural Transport. In *1st Symposium on Advances in Approximate Bayesian Inference, 2018 1–5*, 2019.

[37] C.-W. Huang, D. Krueger, A. Lacoste, and A. Courville. Neural autoregressive flows. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2078–2087. PMLR, 10–15 Jul 2018.

[38] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR 2015*, 2015.

[39] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improving variational inference with inverse autoregressive flow, 2016.

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

[41] H. J. Kushner and G. G. Yin. *Stochastic approximation and recursive algorithms and applications*, volume 35 of *Applications of Mathematics (New York)*. Springer-Verlag, New York, second edition, 2003. Stochastic Modelling and Applied Probability.

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

[43] A. Lee, C. Yau, M. B. Giles, A. Doucet, and C. C. Holmes. 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, 2010.

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

[45] J. S. Liu, W. H. Wong, and A. Kong. Covariance structure of the gibbs sampler with applications to the comparisons of estimators and augmentation schemes. *Biometrika*, 81(1):27–40, 1994.

[46] A. H. Mahmoud, M. Masters, S. J. Lee, and M. A. Lill. Accurate Sampling of Macromolecular Conformations Using Adaptive Deep Learning and Coarse-Grained Representation. *Journal of Chemical Information and Modeling*, 62(7):1602–1617, apr 2022.

[47] J. Mattingly, A. Stuart, and D. Higham. Ergodicity for {SDEs} and approximations: locally lipschitz vector fields and degenerate noise. *Stochastic Processes and their Applications*, 101(2):185 – 232, 2002.

[48] M. Métivier and P. Priouret. Théorèmes de convergence presque sure pour une classe d'algorithmes stochastiques à pas décroissant. *Probability Theory and related fields*, 74(3):403–428, 1987.

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

[50] C. A. Naesseth, F. Lindsten, and D. Blei. Markovian score climbing: Variational inference with KL(p||q). *Advances in Neural Information Processing Systems*, 2020-Decem(MCMC), 2020.

[51] R. M. Neal. Bayesian learning via stochastic dynamics. In *Advances in Neural Information Processing Systems 5, [NIPS Conference]*, pages 475–482, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[52] R. M. Neal. Slice sampling. *The Annals of Statistics*, 31(3):705 – 767, 2003.

[53] K. A. Nicoli, S. Nakajima, N. Strodthoff, W. Samek, K. R. Müller, and P. Kessel. Asymptotically unbiased estimation of physical observables with neural samplers. *Physical Review E*, 101(2), 2020.

[54] F. Noé, S. Olsson, J. Köhler, and H. Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457), 2019.

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

[56] G. Parisi. Correlation functions and computer simulations. *Nuclear Physics B*, 180:378–384, 1981.

[57] M. D. Parno and Y. M. Marzouk. Transport map accelerated markov chain monte carlo. *SIAM-ASA Journal on Uncertainty Quantification*, 6(2):645–682, 2018.

[58] D. Paulin. Concentration inequalities for Markov chains by Marton couplings and spectral methods. *Electronic Journal of Probability*, 20(none):1 – 32, 2015.

[59] E. Pompe, C. Holmes, and K. Łatuszyński. A framework for adaptive mcmc targeting multimodal distributions. *Annals of Statistics*, 48(5):2930–2952, 2020.

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

[61] C. Robert and G. Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[62] C. P. Robert. *The Metropolis–Hastings Algorithm*, pages 1–15. John Wiley & Sons, Ltd, 2015.

[63] G. O. Roberts and J. S. Rosenthal. General state space markov chains and mcmc algorithms. *Probability surveys*, 1:20–71, 2004.

[64] G. O. Roberts and J. S. Rosenthal. Examples of adaptive mcmc. *Journal of computational and graphical statistics*, 18(2):349–367, 2009.

[65] G. O. Roberts and R. L. Tweedie. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363, 1996.

[66] G. O. Roberts and R. L. Tweedie. Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms. *Biometrika*, 83(1):95–110, 03 1996.

[67] P. J. Rossky, J. D. Doll, and H. L. Friedman. Brownian dynamics as smart Monte Carlo simulation. *The Journal of Chemical Physics*, 69(10):4628–4633, 1978.

[68] D. B. Rubin. 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, 1987.

[69] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen. Improved techniques for training gans. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[70] D. Sanz-Alonso. Importance sampling and necessary sample size: an information theory approach. *SIAM/ASA Journal on Uncertainty Quantification*, 6(2):867–879, 2018.

[71] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson. Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12):4629 – 4640, 2008.

[72] J. Song, S. Zhao, and S. Ermon. A-NICE-MC: Adversarial training for MCMC. In *Advances in Neural Information Processing Systems*, pages 5140–5150, 2017.

[73] H. Tjelmeland. Using all Metropolis–Hastings proposals to estimate mean values. Technical report, 2004.

[74] S. T. Tokdar and R. E. Kass. Importance sampling: a review. *WIREs Computational Statistics*, 2(1):54–60, 2010.

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

[76] D. Wu, L. Wang, and P. Zhang. Solving Statistical Mechanics Using Variational Autoregressive Networks. *Physical Review Letters*, 122(8):1–11, 2019.

[77] L. Zhang, C. A. Naesseth, and D. M. Blei. Transport Score Climbing: Variational Inference Using Forward KL and Adaptive Neural Transport. *arXiv preprint*, 2202.01841, 2022.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] The paper suggests novel MCMC technique and is validated on artificial and standard datasets.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 2 and Section 3 in the main text.
   (b) Did you include complete proofs of all theoretical results? [Yes] Yes, the proofs of Section 2 and Section 3 are provided in Appendix B and Appendix D.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Code to reproduce experiments is attached to the supplement. Due to size constraints, we are not available to present all the pre-trained GANs models for the section Section 5, but we intend to do so when possible.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] The hyperparameters are provided in the supplement paper.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Partially yes, but not for all experiments.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We provide this information in the supplement paper.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A] We use only the common knowledge datasets.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A i-SIR Algorithm

A.1 i-SIR and Multiple-try Metropolis (MTM) algorithm

In the MTM algorithm, N i.i.d.sample proposals $\{X^i_{k+1}\}_{i=1}^N$ are drawn from a kernel $\mathsf{T}(y, \cdot)$ in each iteration. In a second step, a sample Y^*_{k+1} is selected with probability proportional to the weights (the exact expression of the weighting weights differs from ours, but this does not change the complexity of the algorithm). In a third step, $N-1$ i.i.d. proposals are drawn from the kernel $\mathsf{T}(Y^*_{k+1}, \cdot)$ and it is assumed that the move is $Y_{k+1} = Y^*_{k+1}$ with an *generalised M-H* ratio, see [44, eq. 3]. This step is bypassed in i-SIR, reducing the computational complexity by a factor of 2.

A.2 i-SIR as a systematic scan two-stage Gibbs sampler

We analyze a slightly modified version of the i-SIR algorithm, with an extra randomization of the state position. The k-th iteration is defined as follows. Given a state $Y_k \in \mathbb{X}$,

(i) draw $I_{k+1} \in \{1, \dots, N\}$ uniformly at random and set $X^{I_{k+1}}_{k+1} = Y_k$;

(ii) draw $X^{1:N\setminus\{I_{k+1}\}}_{k+1}$ independently from the proposal distribution λ;

(iii) compute, for $i \in \{1, \dots, N\}$, the normalized importance weights

$$\omega^i_{N,k+1} = w(X^i_{k+1}) / \sum_{\ell=1}^N w(X^\ell_{k+1});$$

(iv) select Y_{k+1} from the set $X^{1:N}_{k+1}$ by choosing X^i_{k+1} with probability $\omega^i_{N,k+1}$.

Thus, compared to the simplified i-SIR algorithm given in the introduction, the state is inserted uniformly at random into the list of candidates instead of being inserted at the first position. Of course, this change has no impact as long as we are interested in integrating functions that are permutation invariant with respect to candidates, which is the case throughout our work. Still, this randomization makes the analysis much more transparent.

In what follows, we show that i-SIR can be interpreted as a systematic-scan two-stage Gibbs sampler sampling, which alternately samples from the full conditionals of the extended target $\boldsymbol{\varphi}_N$, which is carefully defined below in terms of the state and candidate pool. Here we essentially follow the work of [73, 4, 5]. This is formalized by a dual representation of $\boldsymbol{\varphi}_N$, presented below in Theorem 4, which provides the two complete conditionals in question. We introduce the Markov kernel

$$\boldsymbol{\Lambda}_N(y, \mathrm{d}x^{1:N}) = \frac{1}{N} \sum_{i=1}^N \delta_y(\mathrm{d}x^i) \prod_{j \neq i} \lambda(\mathrm{d}x^j)$$

on $\mathbb{X} \times \mathcal{X}^{\otimes N}$, which probabilistically describes the candidate selection operation in i-SIR. Note that by construction, for each $y \in \mathbb{X}$, $\ell \in \{1, \dots, N\}$ and nonnegative measurable function $h : \mathbb{X} \to \mathbb{R}^+$,

$$\boldsymbol{\Lambda}_N h(y) = \int \boldsymbol{\Lambda}_N(y, \mathrm{d}x^{1:N}) h(x^\ell) = \left(1 - \frac{1}{N}\right) \lambda(h) + \frac{1}{N} h(y).$$

Using the kernel $\boldsymbol{\Lambda}_N$ we may now define properly the extended target $\boldsymbol{\varphi}_N$ as the probability law

$$\boldsymbol{\varphi}_N(\mathrm{d}(y, x^{1:N})) = \pi(\mathrm{d}y) \boldsymbol{\Lambda}_N(y, \mathrm{d}x^{1:N}) = \frac{1}{N} \sum_{i=1}^N \pi(\mathrm{d}y) \delta_y(\mathrm{d}x^i) \prod_{j \neq i} \lambda(\mathrm{d}x^j)$$

on $(\mathbb{X}^{N+1}, \mathcal{X}^{\otimes(N+1)})$. Note that since for every $A \in \mathcal{X}$, $\boldsymbol{\varphi}_N(1_{A \times \mathbb{X}}) = \pi(A)$, the target π coincides with the marginal of $\boldsymbol{\varphi}_N$ with respect to the state. Moreover, it is easily seen that $\boldsymbol{\Lambda}_N$ provides the conditional distribution, under $\boldsymbol{\varphi}_N$, of the candidate pool given the state.

On the other hand, using that $\pi(\mathrm{d}y)\delta_y(\mathrm{d}x^i) = w(x^i)\lambda(\mathrm{d}x^i)\delta_{x^i}(\mathrm{d}y)/\lambda(w)$, the marginal distribution $\boldsymbol{\pi}_N$ of $\boldsymbol{\varphi}_N$ with respect to $x^{1:N}$ is given by

$$\boldsymbol{\pi}_N(\mathrm{d}x^{1:N}) = \frac{1}{\lambda(w)} \Gamma_N 1_{\mathbb{X}}(x^{1:N}) \prod_{j=1}^N \lambda(\mathrm{d}x^j) , \tag{7}$$

where we have set

$$\Gamma_N(x^{1:N}, \mathrm{d}y) = \sum_{i=1}^{N} w(x^i)\delta_{x^i}(\mathrm{d}y)/N, \quad \Pi_N(x^{1:N}, \mathrm{d}y) = \Gamma_N(x^{1:N}, \mathrm{d}y)/\Gamma_N 1_{\mathbb{X}}(x^{1:N})$$

It is interesting to note that the marginal $\boldsymbol{\pi}_N$ has a probability density function, proportional to $\Gamma_N 1_{\mathbb{X}}(x^{1:N}) = \sum_{i=1}^{N} w(x^i)/N$, with respect to the product measure $\lambda^{\otimes N}$. Using (7), we immediately obtain the following result.

Theorem 4 (duality of extended target)**.** *For every $N \in \mathbb{N}^*$,*

$$\boldsymbol{\varphi}_N(\mathrm{d}(y, x^{1:N})) = \pi(\mathrm{d}y)\boldsymbol{\Lambda}_N(y, \mathrm{d}x^{1:N}) = \boldsymbol{\pi}_N(\mathrm{d}x^{1:N})\Pi_N(x^{1:N}, \mathrm{d}y).$$

Using this dual representation of $\boldsymbol{\varphi}_N$, i-SIR can be interpreted as a two-stage Gibbs sampler. Given the state Y_k, N candidates $X_{k+1}^{1:N}$ are sampled from $\boldsymbol{\Lambda}_N(Y_k, \cdot)$. In a second step, the next state Y_{k+1} is sampled given the current candidates from $\Pi_N(X_{k+1}^{1:N}, \cdot)$. The two-stages Gibbs sampler generates a Markov chain $((Y_k, X_k^{1:N}))_{k\in\mathbb{N}}$ with Markov kernel

$$\mathbf{P}_N((y, x^{1:N}), C) = \int \boldsymbol{\Lambda}_N(y, \mathrm{d}\tilde{x}^{1:N})\Pi_N(\tilde{x}^{1:N}, \mathrm{d}\tilde{y})1_C(\mathrm{d}(y, \tilde{x}^{1:N})) \ , \quad C \in \mathcal{X}^{\otimes(N+1)} \ .$$

Note that the Markov kernel $\mathbf{P}_N(y, x^{1:N}, \cdot)$ does not depend on $x^{1:N}$, which means that only the state Y_k needs to be stored from one iteration to another. Given a distribution $\boldsymbol{\xi}$ on $(\mathbb{X}^{n+1}, \mathcal{X}^{\otimes(n+1)})$, we denote by $\mathbb{P}_{\boldsymbol{\xi}}$ the distribution of the canonical Markov chain $((Y_k, X_k^{1:N}))_{k\in\mathbb{N}}$ with kernel $\mathbf{P}_N$. With these notations, for any nonnegative measurable function $f : \mathbb{X}^{n+1} \to \mathbb{R}$, we get, for $k \in \mathbb{N}^*$,

$$\mathbb{E}_{\boldsymbol{\xi}}\left[f(Y_k, X_k^{1:N}) \,\middle|\, \mathcal{F}_{k-1} \right] = \int \mathbf{P}_N((Y_{k-1}, X_{k-1}^{1:N}), \mathrm{d}(y, x^{1:N}))f(x^{1:N}) = \mathbf{P}_N f(Y_{k-1}, X_{k-1}^{1:N}) \ .$$

The systematic scan two-stages Gibbs sampler is one of the MCMC algorithm structures that has given rise to many works. We summarize in the theorem below the important properties of this sampler; see [45], [61, Chapter 9], [3] and the references therein.

Theorem 5. *Assume that for any $y \in \mathbb{X}$, $w(y) > 0$. Then,*

- *The Markov kernel $\mathbf{P}_N$ is Harris recurrent and ergodic with unique invariant distribution $\boldsymbol{\varphi}_N$.*
- *The Markov kernel P_N is reversible w.r.t. π, Harris recurrent and ergodic.*

The proof follows from [61, Theorem 9.6, Lemma 9.11]. The following theorem establishes the unbiasedness of the estimator $\Pi_N f(X^{1:N})$ under $\boldsymbol{\varphi}_N$.

Theorem 6. *For every $N \in \mathbb{N}^*$ and π-integrable function f,*

$$\pi(f) := \int \Pi_N f(x^{1:N})\boldsymbol{\pi}_N(\mathrm{d}x^{1:N}) = \int \Pi_N f(x^{1:N})\pi(\mathrm{d}x^1)\prod_{j=2}^{N}\lambda(\mathrm{d}x^j) \ .$$

Proof. Using (7) we get

$$\begin{aligned}\int \boldsymbol{\pi}_N(\mathrm{d}x^{1:N})\Pi_N f(x^{1:N}) &= \int \frac{1}{N\lambda(w)}\sum_{\ell=1}^{N} w(x^\ell)\Pi_N f(x^{1:N})\prod_{j=1}^{N}\lambda(\mathrm{d}x^j) \\ &= \frac{1}{N\lambda(w)}\int \sum_{i=1}^{N} w(x^i)f(x^i)\prod_{j=1}^{N}\lambda(\mathrm{d}x^j) = \pi(f),\end{aligned}$$

and the first identity follows. The second identity stems from the fact that the function $\Pi_N f(x^{1:N})$ is invariant under permutation. □

B Proofs of Section 2

B.1 Uniform geometric ergodicity of the $\mathrm{i\text{-}SIR}$ Markov kernel

Here we provide a simple direct proof of the bound (2). We preface the proof by a technical lemma.

Lemma 7. *Let $Y^{1:M}$ be M independent random variables, satisfying $\mathbb{E}[Y_i] = 1$, and $\operatorname{Var}[Y_i] < \infty$ for $i \in \{1, \ldots, M\}$. Then, for $S_M = \sum_{i=1}^{M} Y_i$ and $a, b > 0$*

$$\mathbb{E}\left[(a + bS_M)^{-1}\right] \leq (a + bM/2)^{-1} + (4/a)\operatorname{Var}[S_M]/M^2 \ .$$

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

$$\frac{1}{a + bS_M} = \frac{1}{a + bS_M} 1\{S_M < K\} + \frac{1}{a + bS_M} 1\{S_M \geq K\} \leq \frac{1}{a + bK} + \frac{1}{a} 1\{S_M < K\}$$

and in particular, $\mathbb{E}[(a + bS_M)^{-1}] \leq (a + bK)^{-1} + a^{-1}\mathbb{P}(S_M < K)$. By Markov's inequality,

$$\mathbb{P}(S_M < K) = \mathbb{P}(S_M - M < -(M - K)) \leq \frac{\operatorname{Var}[S_M]}{(M - K)^2}$$

In particular, for $K = M/2$, we have $\mathbb{P}(S_M < K) \leq 4\operatorname{Var}[S_M]/M^2$. □

Proof of (2). For $(x, \mathsf{A}) \in \mathbb{X} \times \mathcal{X}$, we get

$$\begin{aligned}
\mathsf{P}_N(x, \mathsf{A}) &= \int \delta_x(\mathrm{d}x^1) \sum_{i=1}^{N} \frac{w(x^i)}{\sum_{j=1}^{N} w(x^j)} 1_{\mathsf{A}}(x^i) \prod_{j=2}^{N} \lambda(\mathrm{d}x^j) \\
&= \int \frac{w(x)}{w(x) + \sum_{j=2}^{N} w(x^j)} 1_{\mathsf{A}}(x) \prod_{j=2}^{N} \lambda(\mathrm{d}x^j) + \int \sum_{i=2}^{N} \frac{w(x^i)}{w(x) + \sum_{j=2}^{N} w(x^j)} 1_{\mathsf{A}}(x^i) \prod_{j=2}^{N} \lambda(\mathrm{d}x^j) \\
&\geq \sum_{i=2}^{N} \int \frac{w(x^i)}{w(x) + w(x^i) + \sum_{j=2, j\neq i}^{N} w(x^j)} 1_{\mathsf{A}}(x^i) \prod_{j=2}^{N} \lambda(\mathrm{d}x^j) \\
&\overset{(a)}{\geq} \sum_{i=2}^{N} \int \pi(\mathrm{d}x^i) 1_{\mathsf{A}}(x^i) \int \frac{\lambda(w)}{w(x) + w(x^i) + \sum_{j=2, j\neq i}^{N} w(x^j)} \prod_{j=2, j\neq i}^{N} \lambda(\mathrm{d}x^j) \ . \qquad (8)
\end{aligned}$$

Here in (a) we used Fubini's theorem together with $w(x)\lambda(\mathrm{d}x) = \pi(\mathrm{d}x)\lambda(w)$. Finally, since the function $f \colon z \mapsto (z + a)^{-1}$ is convex on $\mathbb{R}_+$ and $a > 0$, we get for $i \in \{2, \ldots, N\}$,

$$\begin{aligned}
&\int \frac{\lambda(w)}{w(x) + w(x^i) + \sum_{j=2, j\neq i}^{N} w(x^j)} \prod_{j=2, j\neq i}^{N} \lambda(\mathrm{d}x^j) \\
&\geq \frac{\lambda(w)}{\int w(x) + w(x^i) + \sum_{j=2, j\neq i}^{N} w(x^j) \prod_{j=2, j\neq i}^{N} \lambda(\mathrm{d}x^j)} \\
&\geq \frac{\lambda(w)}{w(x) + w(x^i) + (N-2)\lambda(w)} \geq \frac{1}{2\mathrm{L} + N - 2} \ .
\end{aligned}$$

With the bound above we obtain the inequality

$$\mathsf{P}_N(x, \mathsf{A}) \geq \pi(\mathsf{A}) \times \frac{N - 1}{2\mathrm{L} + N - 2} = \epsilon_N \pi(\mathsf{A}) \ . \qquad (9)$$

This means that the whole space $\mathbb{X}$ is $(1, \epsilon_N \pi)$-small (see [21, Definition 9.3.5]). Since $\mathsf{P}_N(x, \cdot)$ and π are probability measures, (9) implies

$$\|\mathsf{P}_N(x, \cdot) - \pi\|_{\mathrm{TV}} = \sup_{\mathsf{A} \in \mathcal{X}} |\mathsf{P}_N(x, \mathsf{A}) - \pi(\mathsf{A})| \leq 1 - \epsilon_N = \kappa_N \ .$$

The statement follows from [21, Theorem 18.2.4] applied with $m = 1$. □

B.2 Proof of Theorem 2

We preface the proof with some preparatory lemmas.

Lemma 8. *Let* $\mathsf{K} \subset \mathbb{X}$*, such that* $w_{\infty,\mathsf{K}} := \sup_{x\in\mathsf{K}}\{w(x)/\lambda(w)\} < \infty$ *and* $\pi(\mathsf{K}) > 0$*. Then, for all* $(x, \mathsf{A}) \in \mathsf{K} \times \mathcal{X}$*, we get that*

$$\mathsf{P}_N(x, \mathsf{A}) \geq \epsilon_{N,K}\pi_K(\mathsf{A}) ,$$

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

Note that if the weight function w is upper semi-continuous, then for any compact K, $w_{\infty,\mathsf{K}} = \sup_{x\in\mathsf{K}} w(x) < \infty$. Moreover, $\lim_{N\to\infty} \epsilon_{N,K} = \pi(\mathsf{K})$.

Proof. Let $(x, \mathsf{A}) \in \mathbb{X} \times \mathcal{X}$. Then, using the lower bound (8), we obtain

$$\begin{aligned}\mathsf{P}_N(x, \mathsf{A}) &\geq \sum_{i=2}^{N} \int \pi(\mathrm{d}x^i)1_{\mathsf{A}}(x^i) \int \frac{\lambda(w)}{w(x) + w(x^i) + \sum_{j=2,j\neq i}^{N} w(x^j)} \prod_{j=2,j\neq i}^{N} \lambda(\mathrm{d}x^j) \\ &\geq (N-1) \int \pi(\mathrm{d}y)1_{\mathsf{A}}(y) \frac{1}{w(x)/\lambda(w) + w(y)/\lambda(w) + N - 2} ,\end{aligned}$$

where the last inequality follows from Jensen's inequality and the convexity of the function $z \mapsto (z + a)^{-1}$ on $\mathbb{R}_+$. We conclude by noting that

$$\begin{aligned}P_N(x, \mathsf{A}) &\geq (N-1) \int \pi(\mathrm{d}y)1_{\mathsf{A}\cap\mathsf{K}}(y) \frac{1}{w(x)/\lambda(w) + w(y)/\lambda(w) + N - 2} \\ &\geq \frac{N-1}{2w_{\infty,\mathsf{K}} + N - 2} \int \pi(\mathrm{d}y)1_{\mathsf{A}\cap\mathsf{K}}(y) = \frac{(N-1)\pi(\mathsf{K})}{2w_{\infty,\mathsf{K}} + N - 2}\pi_{\mathsf{K}}(\mathsf{A}) .\end{aligned}$$

□

Lemma 9. *Assume* ***A****1. Then for all* $x \in \mathbb{X}$*, any function* $V : \mathbb{X} \to [1, \infty)$ *with* $\pi(V) < \infty$*,* $\lambda(V) < \infty$*, and* $N \geq 3$*, it holds that*

$$\mathsf{P}_N V(x) \leq V(x) + \mathsf{b}_{\mathsf{P}_N} , \tag{10}$$

where $\mathsf{b}_{\mathsf{P}_N}$ *is given in* (12)*.*

Note that

$$\mathsf{b}_{\mathsf{P}_\infty} := \lim_{N\to\infty} \mathsf{b}_{\mathsf{P}_N} = 2\pi(V) + 4\operatorname{Var}_{\lambda}[w]/\lambda(V) . \tag{11}$$

Proof. Note first that

$$\begin{aligned}\mathsf{P}_N V(x) &= V(x) \int \frac{w(x)}{w(x) + \sum_{j=2}^{N} w(x^j)} \prod_{j=2}^{N} \lambda(\mathrm{d}x^j) + \int \sum_{i=2}^{N} \frac{w(x^i)}{w(x) + \sum_{j=2}^{N} w(x^j)} V(x^i) \prod_{j=2}^{N} \lambda(\mathrm{d}x^j) \\ &\leq V(x) + (N-1)U_N\end{aligned}$$

where we have set

$$U_N = \int \frac{w(x^2)V(x^2)\lambda(\mathrm{d}x^2)}{w(x^2) + \sum_{j=3}^{N} w(x^j)} \prod_{j=3}^{N} \lambda(\mathrm{d}x^j) .$$

Since the function $z \mapsto z/(z + a)$ is concave on $\mathbb{R}_+$ for $a > 0$, we have

$$\begin{aligned}&\int \frac{w(x^2)}{w(x^2) + \sum_{j=3}^{N} w(x^j)} V(x^2)\lambda(\mathrm{d}x^2) = \lambda(V) \int \frac{w(x^2)}{w(x^2) + \sum_{j=3}^{N} w(x^j)} \frac{V(x^2)\lambda(\mathrm{d}x^2)}{\lambda(V)} \\ &\leq \lambda(V) \frac{\int w(x^2)V(x^2)\lambda(\mathrm{d}x^2)/\lambda(V)}{\int w(x^2)V(x^2)\lambda(\mathrm{d}x^2)/\lambda(V) + \sum_{j=3}^{N} w(x^j)} \leq \frac{\pi(V)\lambda(w)}{\pi(V)\lambda(w)/\lambda(V) + \sum_{j=3}^{N} w(x^j)} .\end{aligned}$$

The bound above implies that, with renormalization,

$$U_N \leq \int \frac{\pi(V)}{\pi(V)/\lambda(V) + \sum_{j=3}^{N} w(x^j)/\lambda(w)} \prod_{j=3}^{N} \lambda(\mathrm{d}x^j)$$

Applying now Lemma 7 with $a = \pi(V)/\lambda(V)$, $b = 1$, $M = N-2$, and $Y_j = w(x^j)/\lambda(w)$, we obtain that

$$U_N \leq \frac{\pi(V)}{\pi(V)/\lambda(V) + (N-2)/2} + \frac{4\operatorname{Var}_\lambda[w]}{(N-2)\lambda(V)} \,.$$

Combining the bounds above yields (10) with

$$\mathsf{b}_{\mathsf{P}_N} = \frac{(N-1)\pi(V)}{\pi(V)/\lambda(V) + (N-2)/2} + \frac{4(N-1)\operatorname{Var}_\lambda[w]}{(N-2)\lambda(V)} \,. \tag{12}$$

□

Lemma 10. *Let* P *be a Markov kernel on* $(\mathbb{X}, \mathcal{X})$, γ *be a probability measure on* $(\mathbb{X}, \mathcal{X})$, *and* $\epsilon > 0$. *Let* $\mathsf{C} \in \mathcal{X}$ *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* $(1, \epsilon\gamma_{\mathsf{Q}})$*-small set for* $\mathrm{P}\mathsf{Q}$, *where* $\gamma_{\mathsf{Q}}(\mathsf{A}) = \int \gamma(\mathrm{d}y)\mathsf{Q}(y, \mathsf{A})$ *for* $\mathsf{A} \in \mathcal{X}$.

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

$$\mathrm{P}\mathsf{Q}(x, \mathsf{A}) = \int \mathrm{P}(x, \mathrm{d}y)\mathsf{Q}(y, \mathsf{A}) \geq \epsilon \int \gamma(\mathrm{d}y)\mathsf{Q}(y, \mathsf{A}) = \epsilon\gamma_{\mathsf{Q}}(\mathsf{A})\,.$$

□

Lemma 11. *Let* P *and* Q *be two irreducible Markov kernels with* π *as their unique invariant distribution. Let* $V : \mathbb{X} \to [1, \infty)$ *be a measurable function. Suppose that there exist* $\lambda_Q \in [0, 1)$ *and* $\mathsf{b}_{\mathrm{P}}, \mathsf{b}_{\mathsf{Q}} \in \mathbb{R}_+$ *such, that* $\mathrm{P}V(x) \leq V(x) + \mathsf{b}_{\mathrm{P}}$ *and* $\mathsf{Q}V(x) \leq \lambda_{\mathsf{Q}}V(x) + \mathsf{b}_{\mathsf{Q}}$. *Let* $r_0 \geq 1$. *Also assume that for all* $r \geq r_0$, *there exist* $\epsilon_r > 0$ *and a probability measure* γ_r *such that for all* $(x, \mathsf{A}) \in \mathsf{V}_r \times \mathcal{X}$, $\mathrm{P}(x, \mathsf{A}) \geq \epsilon_r\gamma_r(\mathsf{A})$, *where* $\mathsf{V}_r = \{x \in \mathbb{X} : V(x) \leq r\}$. *Define* $\mathrm{K} = \mathrm{P}\mathsf{Q}$ *and* $\lambda_{\mathrm{K}} = \lambda_{\mathsf{Q}}$, $\mathsf{b}_{\mathrm{K}} = \mathsf{b}_{\mathrm{P}} + \mathsf{b}_{\mathsf{Q}}$. *Then,*

$$\mathrm{K}V(x) \leq \lambda_{\mathrm{K}}V(x) + \mathsf{b}_{\mathrm{K}} \textit{ and, for all } x \in \mathsf{V}_r,\ \mathrm{K}(x, \mathsf{A}) \geq \epsilon_r\gamma_{\mathsf{Q},r}(\mathsf{A}),$$

where $\gamma_{\mathsf{Q},r}(\mathsf{A}) = \int \gamma_r(\mathrm{d}y)\mathsf{Q}(y, \mathsf{A})$. *Moreover, let* $r \geq r_0$ *be such that* $\lambda_{\mathrm{K}} + 2\mathsf{b}_{\mathrm{K}}/(1+r) < 1$. *Then, for any* $x \in \mathbb{X}$ *and* $k \in \mathbb{N}$,

$$\|\mathrm{K}^k(x, \cdot) - \pi\|_V \leq c_{\mathrm{K}}\{V(x) + \pi(V)\}\rho_{\mathrm{K}}^k \,,$$

where

$$\rho_{\mathrm{K}} = \frac{\log(1-\epsilon_r)\log\bar{\lambda}_{\mathrm{K}}}{\log(1-\epsilon_r) + \log\bar{\lambda}_{\mathrm{K}} - \log\bar{b}_{\mathrm{K}}} \,, \quad c_{\mathrm{K}} = (\lambda_{\mathrm{K}} + \mathsf{b}_{\mathrm{K}})(1 + \bar{b}_{\mathrm{K}}/[(1-\epsilon_r)(1-\bar{\lambda}_{\mathrm{K}})]),$$
$$\bar{\lambda}_{\mathrm{K}} = \lambda_{\mathrm{K}} + 2\mathsf{b}_{\mathrm{K}}/(1+r) \,, \quad \bar{b}_{\mathrm{K}} = \lambda_{\mathrm{K}}r + \mathsf{b}_{\mathrm{K}} \,.$$

Proof. By Lemma 10, it holds that for any $(x, \mathsf{A}) \in \mathsf{V}_r \times \mathcal{X}$, $\mathrm{K}(x, \mathsf{A}) \geq \epsilon_r\gamma_{\mathsf{Q},r}(\mathsf{A})$. Moreover, for any $x \in \mathbb{X}$, $\mathrm{K}V(x) = \mathrm{P}\mathsf{Q}V(x) \leq \lambda_{\mathsf{Q}}\mathrm{P}V(x) + \mathsf{b}_{\mathsf{Q}} \leq \lambda_{\mathsf{Q}}V(x) + \mathsf{b}_{\mathsf{Q}} + \mathsf{b}_{\mathrm{P}}$. The proof is completed with [21, Theorem 19.4.1]. □

Proof of Theorem 2. The proof consists of the 3 main steps:

1. Lemma 8 implies that for all $r \geq r_{\mathsf{R}}$, the level sets V_r for the Markov kernel P_N are $(1, \epsilon_{r,N}\gamma_r)$-small for the Markov kernel P_N, where
$$\epsilon_{r,N} = (N-1)\pi(\mathsf{V}_r)/[2w_{\infty,r} + N - 2],$$
and $\gamma_r(\mathsf{A}) = \int \pi_{\mathsf{V}_r}(\mathrm{d}y)\mathsf{R}(y, \mathsf{A})$ with $\pi_{\mathsf{V}_r}(B) = \pi(B \cap \mathsf{V}_r)/\pi(\mathsf{V}_r)$, for any $B \in \mathcal{X}$.
2. Lemma 9 implies that for all $x \in \mathbb{X}$, $\mathsf{P}_N V(x) \leq V(x) + \mathsf{b}_{\mathsf{P}_N}$, where $\mathsf{b}_{\mathsf{P}_N}$ is given in (12).
3. We finally show (see Lemma 11) that the Markov kernel K_N also satisfies a Foster-Lyapunov condition with the same drift function V as R, that is, $\mathsf{K}_N V \leq \lambda_{\mathsf{R}}V + \mathsf{b}_{\mathsf{K}_N}$ with $\mathsf{b}_{\mathsf{K}_N} = \mathsf{b}_{\mathsf{R}} + \mathsf{b}_{\mathsf{P}_N}$.

We conclude by using Lemma 11. We choose $r_N = r_{\mathsf{R}} \vee \{4\mathsf{b}_{\mathsf{K}_N}/(1-\lambda_{\mathsf{R}}) - 1\}$. Then $\lambda_{\mathsf{R}} + 2\mathsf{b}_{\mathsf{K}_N}/(1+r_N) \leq (1+\lambda_{\mathsf{R}})/2 < 1$, and Lemma 11 implies (3) with

$$\log\tilde{\kappa}_{\mathsf{K}_N} = \frac{\log(1-\epsilon_{r,N})\log\bar{\lambda}_{\mathsf{K}_N}}{\log(1-\epsilon_{r,N}) + \log\bar{\lambda}_{\mathsf{K}_N} - \log\bar{b}_{\mathsf{K}_N}} \,,$$
$$c_{\mathsf{K}_N} = (\lambda_{\mathsf{R}} + \bar{b}_{\mathsf{K}_N})(1 + \bar{b}_{\mathsf{K}_N}/[2(1-\epsilon_{r_N,N})(1-\bar{\lambda}_{\mathsf{K}_N})]) \,,$$
$$\bar{\lambda}_{\mathsf{K}_N} = (1+\lambda_{\mathsf{R}})/2 \,, \quad \bar{b}_{\mathsf{K}_N} = \lambda_{\mathsf{R}}r_N + \mathsf{b}_{\mathsf{K}_N} \,.$$

Set $\mathsf{b}_{\mathsf{K}_\infty} = \lim_{N\to\infty} \mathsf{b}_{\mathsf{K}_N} = \mathsf{b}_\mathsf{R}+\mathsf{b}_{\mathsf{P}_\infty}$, where $\mathsf{b}_{\mathsf{P}_\infty}$ is defined in (11), $r_\infty = r_\mathsf{R} \vee [4\mathsf{b}_{\mathsf{K}_\infty}/(1-\lambda_\mathsf{R})-1]$ and $\epsilon_\infty = \pi(\mathsf{V}_{r_\infty})$. With these notations, we have

$$\begin{aligned} \log \tilde{\kappa}_{\mathsf{K}_\infty} &= \frac{\log(1-\epsilon_\infty)\log\bar{\lambda}_{\mathsf{K}_\infty}}{\log(1-\epsilon_\infty)+\log\bar{\lambda}_{\mathsf{K}_\infty} - \log\bar{b}_{\mathsf{K}_\infty}} \;, \\ c_{\mathsf{K}_\infty} &= (\lambda_\mathsf{R} + \bar{b}_{\mathsf{K}_\infty})(1+\bar{b}_{\mathsf{K}_\infty}/[(1-\epsilon_\infty)(1-\bar{\lambda}_{\mathsf{K}_\infty})]) \\ \bar{\lambda}_{\mathsf{K}_\infty} &= (1+\lambda_\mathsf{R})/2 \;,\; \bar{b}_{\mathsf{K}_\infty} = \lambda_\mathsf{R} r_\infty + \mathsf{b}_{\mathsf{K}_\infty} \;. \end{aligned} \tag{13}$$

□

C Metropolis-Adjusted Langevin rejunevation kernel

This section addresses the convergence of the Metropolis Adjusted Langevin algorithm (MALA) for sampling from a positive target probability density π on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$, where $\mathcal{B}(\mathbb{R}^d)$ is the Borel σ field of $\mathbb{R}^d$ endowed with the Euclidean topology. For simplicity, let $U = -\log\pi$ be the associated potential function. MALA is a Markov chain Monte Carlo (MCMC) method based on Langevin diffusion associated with π:

$$\mathrm{d}\mathbf{X}_t = -\nabla U(\mathbf{X}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t \;, \tag{14}$$

where $(\mathbf{B}_t)_{t\geq 0}$ is a d-dimensional Brownian motion. It is known that under mild conditions this diffusion admits a strong solution $(\mathbf{X}_t^{(x)})_{t\geq 0}$ for any starting point $x \in \mathbb{R}^d$ and defines a Markov semigroup $(\mathbf{P}_t)_{t\geq 0}$ for any $t \geq 0$, $x \in \mathbb{R}^d$ and $\mathsf{A} \in \mathcal{B}(\mathbb{R}^d)$ by $\mathbf{P}_t(x,\mathsf{A}) = \; PP(\mathbf{X}_t^{(x)} \in \mathsf{A})$. Moreover, this Markov semigroup admits π as its unique stationary measure, is ergodic and even V-uniformly geometrically ergodic with additional assumptions on U (see [65, 47]). However, sampling a path solution of (14) is a real challenge in most cases, and discretizations are used instead to obtain a Markov chain with similar long-term behaviour. Here we consider the Euler-Maruyama discretization, which is given by (14), defined for all $k \geq 0$ by

$$Y_{k+1} = Y_k - \gamma\nabla U(Y_k) + \sqrt{2\gamma}Z_{k+1} \;, \tag{15}$$

where γ is the step size of the discretization and. $\{Z_k,\ k \in \mathbb{N}^*\}$ is a i.i.d. sequence of d-dimensional standard Gaussian random variables. This algorithm was proposed by [24, 56] and later studied by [28, 29, 51, 65]. According to [65], this algorithm is called the Unadjusted Langevin algorithm (ULA). A drawback of this method is that even if the Markov chain $\{Y_k,\ k \in \mathbb{N}\}$ has a unique stationary distribution π_γ and is ergodic (which is guaranteed under mild assumptions about U), π_γ is different from π most of the time. To solve this problem, in [67, 65] it is proposed to use the Markov kernel associated with the recursion defined by the Euler-Maruyama discretization (15) as a proposal kernel in a Metropolis-Hastings algorithm that defines a new Markov chain $\{X_k,\ k \in \mathbb{N}\}$ by:

$$X_{k+1} = X_k + 1_{\mathbb{R}_+}(U_{k+1} - \alpha_\gamma(X_k, \tilde{Y}_{k+1}))\{\tilde{Y}_{k+1} - X_k\} \;, \tag{16}$$

where $\tilde{Y}_{k+1} = X_k - \gamma\nabla U(X_k) + \sqrt{2\gamma}Z_{k+1}$, $\{U_k,\ k \in \mathbb{N}^*\}$ is a sequence of i.i.d. uniform random variables on $[0,1]$ and $\alpha_\gamma : \mathbb{R}^{2d} \to [0,1]$ is the usual Metropolis acceptance ratio. This algorithm is called Metropolis Adjusted Langevin Algorithm (MALA) and has since been used in many applications.

Denote by r_γ the proposal transition density associated to the Euler-Maruyama discretization (15) with stepsize $\gamma > 0$, *i.e.*, for any $x, y \in \mathbb{R}^d$,

$$r_\gamma(x,y) = (4\pi\gamma)^{-d/2}\exp\left(-(4\gamma)^{-1}\|y - x + \gamma\nabla U(x)\|^2\right) \;.$$

Then, the Markov kernel R_γ of the MALA algorithm (16) is given for $\gamma > 0$, $x \in \mathbb{R}^d$, and $\mathsf{A} \in \mathcal{B}(\mathbb{R}^d)$ by

$$\begin{aligned} R_\gamma(x,\mathsf{A}) &= \int_{\mathbb{R}^d} 1_\mathsf{A}(y)\alpha_\gamma(x,y)r_\gamma(x,y)\mathrm{d}y + \delta_x(\mathsf{A})\int_{\mathbb{R}^d}\{1-\alpha_\gamma(x,y)\}r_\gamma(x,y)\mathrm{d}y \;, \qquad (17) \\ \alpha_\gamma(x,y) &= 1 \wedge \frac{\pi(y)r_\gamma(y,x)}{\pi(x)r_\gamma(x,y)} \;. \end{aligned}$$

It is well-known, see e.g. [65], that for any $\gamma > 0$, R_γ is reversible with respect to π and π-irreducible.

H1. *The function $U : \mathbb{R}^d \to \mathbb{R}$ is three times continuously differentiable. In addition, $\nabla U(0) = 0$ and there exists $\mathtt{L} \geq 0$ and $\mathtt{M} \geq 0$ such that $\sup_{x \in \mathbb{R}^d} \|\mathrm{D}^2 U(x)\| \leq \mathtt{L}$ such that $\sup_{x \in \mathbb{R}^d} \|\mathrm{D}^3 U(x)\| \leq \mathtt{M}$.*

The condition $\nabla U(0) = 0$ is satisfied (up to a translation) as soon as U has a local minimum, which is the case when $\lim_{\|x\| \to +\infty} U(x) = +\infty$, since U is continuous.

H2. *There exist $\mathtt{m} > 0$ and $\mathtt{K} \geq 0$ such that for any $x, y \in \mathbb{R}^d$, $\|x\| \geq \mathtt{K}$ and $\|y\| = 1$,*

$$\mathrm{D}^2 U(x)\{y\}^{\otimes 2} \geq \mathtt{m} \; .$$

Note that under **H** 1 and **H** 2, for any $x, y \in \mathbb{R}^d$, $\|y\| = 1$, it holds that

$$\mathrm{D}^2 U(x)\{y\}^{\otimes 2} \geq \mathtt{m} - (\mathtt{m} + \mathtt{L})\mathbb{1}_{\mathrm{B}(0,\mathtt{K})}(x) \; .$$

In the case $\mathtt{K} = 0$, **H** 2 amounts to U being strongly convex and the convexity constant being equal to $\mathtt{m}$. However, if $\mathtt{K} > 0$, **H** 2 is a slight strengthening of the condition of strong convexity at infinity considered in [18, 23]: there is $\mathtt{m}' > 0$ and $\mathtt{K}' \geq 0$ such that for each $x, y \in \mathbb{R}^d$, $\|x - y\| \geq \mathtt{K}'$

$$\langle \nabla U(x) - \nabla U(y), x - y \rangle \geq \mathtt{m}' \|x - y\|^2 \; . \tag{18}$$

Indeed, if (18) holds for any $x, y \in \mathbb{R}^d$ that $\|x\| \vee \|y\| \geq \mathtt{K}'$ instead of $\|x - y\| \geq \mathtt{K}'$, then a simple calculation implies that **H** 2 holds with $\mathtt{m} \leftarrow \mathtt{m}'$ and $\mathtt{K} \leftarrow \mathtt{K}' + 1$. Finally, while the condition (18) holds for $x, y \in \mathbb{R}^d$, $\|x - y\| \geq \mathtt{K}'$, is weaker than **H** 2, it may be more convenient in many situations to check whether the latter holds.

Lemma 12. *Assume **H** 1 and **H** 2 hold. The function U satisfies for any $x \in \mathbb{R}^d$,*

$$\langle \nabla U(x), x \rangle \geq (\mathtt{m}/2)\|x\|^2 - \tilde{\mathtt{C}} \mathbb{1}_{\mathrm{B}(0,\tilde{\mathtt{K}})}(x) \; ,$$

with $\tilde{\mathtt{K}} = 2\mathtt{K}(1 + \mathtt{L}/\mathtt{m})$ and $\tilde{\mathtt{C}} = \mathtt{L}\tilde{\mathtt{K}}^2$.

Note that under **H** 1 and **H** 2, $\mathtt{m} \leq \mathtt{L}$. Define for any $\eta > 0$, $V_\eta : \mathbb{R}^d \to [1, +\infty)$ for any $x \in \mathbb{R}^d$ by

$$V_\eta(x) = \exp(\eta \|x\|^2) \; . \tag{19}$$

The analysis of MALA is naturally related to the study of the ULA algorithm. More precisely, since for any $x \in \mathbb{R}^d$ and $\mathsf{A} \in \mathcal{B}(\mathbb{R}^d)$, the Markov kernel corresponding to ULA (15) is given by

$$Q_\gamma(x, \mathsf{A}) = \int_{\mathbb{R}^d} \mathbb{1}_{\mathsf{A}}(x - \gamma \nabla U(x) + \sqrt{2\gamma} z) \, \mathrm{g}(z) \mathrm{d}z.$$

To show that MALA satisfies a Lyapunov condition, we first state a drift condition for the ULA algorithm.

Proposition 13. *Assume **H** 1 and **H** 2 and let $\bar\gamma \in \big(0, \mathtt{m}/(4\mathtt{L}^2)\big]$. Then, for any $\gamma \in (0, \bar\gamma]$, $x \in \mathbb{R}^d$,*

$$Q_\gamma V_{\bar\eta}(x) \leq \exp(-\bar\eta \mathtt{m} \gamma \|x\|^2/4) V_{\bar\eta}(x) + b^{\mathrm{U}}_{\bar\gamma} \gamma \mathbb{1}_{\mathrm{B}(0, K^{\mathrm{U}})}(x) \; ,$$

where $V_{\bar\eta}$ is defined in (19), $\bar\eta = \mathtt{m}/16$, $K^{\mathrm{U}} = \max(\tilde{\mathtt{K}}, 4\sqrt{d/\mathtt{m}})$, $\tilde{\mathtt{K}}$ is defined in Lemma 12 and

$$\begin{aligned} b^{\mathrm{U}}_{\bar\gamma} = & \big[\bar\eta\{\mathtt{m}/4 + (1 + 16\bar\eta\bar\gamma)(4\bar\eta + 2\mathtt{L} + \bar\gamma\mathtt{L}^2)\}(K^{\mathrm{U}})^2 + 4\bar\eta d\big] \\ & \times \exp(\bar\gamma\bar\eta\{\mathtt{m}/4 + (1 + 16\bar\eta\bar\gamma)(4\bar\eta + 2\mathtt{L} + \bar\gamma\mathtt{L}^2)\}(K^{\mathrm{U}})^2 + 4\bar\eta\bar\gamma d) \; . \end{aligned}$$

Proof. The proof follows from [22, Proposition 6]. □

We now introduce for $\bar\gamma > 0$ the auxiliary constant

$$C_{1,\bar\gamma} = 2(2^{1/2}\mathtt{M} \vee \bar\gamma^{1/2}\mathtt{M}\mathtt{L} \vee 2\mathtt{L}^2[1 \vee \bar\gamma^{1/2} \vee \bar\gamma\mathtt{L} \vee (\bar\gamma\mathtt{L}^{4/3})^{3/2}]) \; . \tag{20}$$

For $\bar\gamma \in \big(0, \mathtt{m}^3/(4\mathtt{L}^4)\big]$, we also define $C_{2,\bar\gamma}$ as

$$C_{2,\bar\gamma} = 2\mathtt{L} + (\bar\gamma/2)\mathtt{L}^2 + 2^{-3/2}\bar\gamma^{3/2}\mathtt{L}^3 + \{2^{1/2}\mathtt{L}^2 + (2^{1/2}\mathtt{L}^2 + 2^{-3/2}\bar\gamma^{1/2})\mathtt{L}^3\}^2(2^4/\mathtt{m}^3) \; .$$

Using Proposition 13, we state a drift condition for the MALA kernel R_γ.

Proposition 14. *Assume **H** 1 and **H** 2. Then, there exist* $\Gamma > 0$ *(given in* (21)*) such that for any* $\bar{\gamma} \in (0, \Gamma]$, $\gamma \in (0, \bar{\gamma}]$ *and* $x \in \mathbb{R}^d$,

$$R_\gamma V_{\bar{\eta}}(x) \leq (1 - \varpi\gamma) V_{\bar{\eta}}(x) + b^{\mathrm{M}}_{\bar{\gamma}} \gamma \mathbb{1}_{\mathrm{B}(0, K^{\mathrm{M}})}(x) \ ,$$

where $V_{\bar{\eta}}$ *is defined by* (19)*,* R_γ *is the Markov kernel of MALA defined by* (17)*,* $\bar{\eta} = \mathtt{m}/16$*,* $\varpi = \bar{\eta}\mathtt{m}(K^{\mathrm{M}})^2/16$*, and*

$$\begin{aligned} \Gamma_{1/2} &= \min\left(1, \mathtt{m}^3/(4\mathtt{L}^4), d^{-1}\right) \ , \quad \Gamma = \min\left(\Gamma_{1/2}, 4/\{\mathtt{m}\bar{\eta}(K^{\mathrm{M}})^2\}\right) \ , \\ K^{\mathrm{M}} &= \max(2^4, 2\mathtt{K}, K^{\mathrm{U}}, \tilde{\mathtt{K}}, 4b_{1/2}^{1/2}/(\mathtt{m}\bar{\eta})^{1/2}) \ , \quad b_{1/2} = C_{2,\Gamma_{1/2}} d + \sup_{u \geq 1}\{u\mathrm{e}^{-u/2^7}\} \ , \\ b^{\mathrm{M}}_{\bar{\gamma}} &= b^{\mathrm{U}}_{\bar{\gamma}} + \bar{\eta}\mathtt{m}(K^{\mathrm{M}})^2 \mathrm{e}^{\bar{\eta}(K^{\mathrm{M}})^2}/16 + C_{1,\bar{\gamma}}\bar{\gamma}^{1/2}\left\{d + \sqrt{3}d^2 + (K^{\mathrm{M}})^2\right\} \ , \end{aligned} \tag{21}$$

where $K^{\mathrm{U}}, b^{\mathrm{U}}_{\bar{\gamma}}$ *are defined in Proposition 13, and* $\tilde{\mathtt{K}}$ *is defined in Lemma 12.*

Proof. The proof follows from [22, Proposition 7]. □

Quantitative bound on the mixing rate of the MALA sampler requires also the *minorization condition* for the MALA kernel. The result below is due to [22, Proposition 12].

Proposition 15. *Assume **H** 1 and **H** 2. Then for any* $K \geq 0$ *there exists* $\tilde{\Gamma}_K > 0$ *(given in* (22) *below), such that for any* $x, y \in \mathbb{R}^d$, $\|x\| \vee \|y\| \leq K$*, and* $\gamma \in (0, \tilde{\Gamma}_K]$ *we have*

$$\|\delta_x R_\gamma^{\lceil 1/\gamma \rceil} - \delta_y R_\gamma^{\lceil 1/\gamma \rceil}\|_{\mathrm{TV}} \leq 2(1 - \varepsilon(K)/2) \ ,$$

where

$$\begin{aligned} \varepsilon(K) &= 2\mathbf{\Phi}\left(-\sqrt{3}(\mathtt{L}+1)^{1/2}K\right) , \quad \tilde{\Gamma}_{1/2} = \mathtt{m}/(4\mathtt{L}^2) \ , \\ \tilde{\Gamma}_K &= \tilde{\Gamma}_{1/2} \wedge \left[\frac{\varepsilon(K)}{2C_{1,\tilde{\Gamma}_{1/2}}(d + \sqrt{3}d^2 + K^2 + 2\tilde{b}^{\mathrm{U}}_{\tilde{\Gamma}_{1/2}}/\mathtt{m})}\right]^2 \ , \\ \tilde{b}^{\mathrm{U}}_{\tilde{\Gamma}_{1/2}} &= 2d + [\max(\tilde{\mathtt{K}}, 2\sqrt{(2d)/\mathtt{m}})]^2\left(\tilde{\Gamma}_{1/2}\mathtt{L}^2 + 2\mathtt{L} + \mathtt{m}/2\right) \ , \end{aligned} \tag{22}$$

where $C_{1,\tilde{\Gamma}_{1/2}}$ *is defined in* (20)*,* $\tilde{\mathtt{K}}$ *is defined in Lemma 12, and* $\mathbf{\Phi}(\cdot)$ *is the cumulative distribution function of the Gaussian distribution with zero mean an unit variance on* $\mathbb{R}$*.*

It is interesting to note that γ is the discretization step of the underlying Langevin diffusion. We have to iterate the kernel $1/\gamma$ times for the diffusion to progress by one time unit. Combining Proposition 14 and Proposition 15 yields the following ergodicity result in $V_{\bar{\eta}}$-norm.

Theorem 16. *Assume **H** 1 and **H** 2. Then, there exist* $\bar{\Gamma} > 0$ *(defined in* (23) *below), such that for any* $\gamma \in \left(0, \bar{\Gamma}\right]$*, there exist* $C_{\bar{\Gamma}} \geq 0$ *and* $\rho_{\bar{\Gamma}} \in [0, 1)$ *(given in* (23)*) satisfying for any* $x \in \mathbb{R}^d$*,*

$$\|\delta_x R_\gamma^k - \pi\|_{V_{\bar{\eta}}} \leq C_{\bar{\Gamma}} \rho_{\bar{\Gamma}}^{\gamma k}\{V_{\bar{\eta}}(x) + \pi(V_{\bar{\eta}})\} \ ,$$

where $\bar{\eta} = \mathtt{m}/16$*,*

$$\begin{aligned} \log \rho_{\bar{\Gamma}} &= \frac{\log(1 - 2^{-1}\varepsilon(K_{\bar{\Gamma}}))\log\bar{\lambda}}{\log(1 - 2^{-1}\varepsilon(K_{\bar{\Gamma}})) + \log\bar{\lambda} - \log\bar{b}^{\mathrm{M}}_{\bar{\Gamma}}} \ , \\ \bar{\lambda} &= (1+\lambda)/2 \ , \quad \lambda = \mathrm{e}^{-\varpi} \ , \quad \bar{b}^{\mathrm{M}}_{\bar{\Gamma}} = \lambda b^{\mathrm{M}}_{\bar{\Gamma}} + M_{\bar{\Gamma}} \ , \quad \bar{\Gamma} = \Gamma \wedge \tilde{\Gamma}_{K_\Gamma} \ , \\ M_{\bar{\gamma}} &= \left(\frac{4b^{\mathrm{M}}_{\bar{\gamma}}(1+\bar{\gamma})}{1-\lambda}\right) \vee 1 \ , \quad K_{\bar{\gamma}} = (\log(M_{\bar{\gamma}})/\bar{\eta})^{1/2} \ , \quad \bar{\gamma} \in \{\bar{\Gamma}, \Gamma\} \ , \\ C_{\bar{\Gamma}} &= \rho_{\bar{\Gamma}}^{-1}\{\lambda+1\}\{1 + \bar{b}^{\mathrm{M}}_{\bar{\Gamma}}/[1 - 2^{-1}\varepsilon(K_{\bar{\Gamma}})(1-\bar{\lambda})]\} \ , \end{aligned} \tag{23}$$

and ϖ *is given in Proposition 14.*

Proof. The proof follows from [22, Theorem 2]. For completeness we repeat here the main steps of the proof. Proposition 14 shows that there exist $\Gamma > 0$ (given in (21)) such that for any $\bar{\gamma} \in (0, \Gamma]$, $\gamma \in (0, \bar{\gamma}]$ and $x \in \mathbb{R}^d$,

$$R_\gamma V_{\bar{\eta}}(x) \leq (1 - \varpi\gamma) V_{\bar{\eta}}(x) + b_{\bar{\gamma}}^{\mathrm{M}} \gamma \ ,$$

where the constants ϖ and $b_{\bar{\gamma}}^{\mathrm{M}}$ are given in Proposition 14. Hence, setting $\lambda = \mathrm{e}^{-\varpi} < 1$, we obtain by induction that

$$R_\gamma^{\lceil 1/\gamma \rceil} V_{\bar{\eta}}(x) \leq \lambda V_{\bar{\eta}}(x) + b_{\bar{\gamma}}^{\mathrm{M}} \ .$$

Now we set $M_{\bar{\gamma}}$ and $K_{\bar{\gamma}}$ as in (23). Then Proposition 15 implies that for any $\bar{\gamma} \in \left(0, \tilde{\Gamma}_{K_\Gamma}\right]$, any $x, y \in \{V_{\bar{\eta}}(\cdot) \leq M_{\bar{\gamma}}\}$, and $\gamma \in (0, \bar{\gamma}]$,

$$\| \updelta_x R_\gamma^{\lceil 1/\gamma \rceil} - \updelta_y R_\gamma^{\lceil 1/\gamma \rceil} \|_{\mathrm{TV}} \leq 2(1 - \varepsilon(K_{\bar{\gamma}})) \ .$$

Now it remains to combine both statements with $\bar{\gamma} = \Gamma \wedge \tilde{\Gamma}_{K_\Gamma}$ and apply [21, Theorem 19.4.1] to the Markov kernel $R_\gamma^{\lceil 1/\gamma \rceil}$. □

Comparison with $\mathrm{Ex}^2\mathrm{MCMC}$ kernel. Based on the results above, we first state the quantitative mixing rate bounds for $\mathrm{Ex}^2\mathrm{MCMC}$ algorithm with the MALA kernel $R_\gamma^{\lceil 1/\gamma \rceil}$ (iterated $\lceil 1/\gamma \rceil$ times) applied as rejuvenation kernel. The corresponding Markov kernel writes for $x \in \mathbb{R}^d$ and $\mathsf{A} \in \mathcal{B}(\mathbb{R}^d)$ as

$$\mathsf{K}_{N,\gamma}(x, \mathsf{A}) = \mathsf{P}_N R_\gamma^{\lceil 1/\gamma \rceil}(x, \mathsf{A}) = \int \mathsf{P}_N(x, \mathrm{d}y) R_\gamma^{\lceil 1/\gamma \rceil}(y, \mathsf{A}) \ ,$$

where $R_\gamma(x, \mathsf{A})$ is defined in (17). Note also that, for $r \geq 1$, and $V_{\bar{\eta}}$ defined in (19), the level sets

$$\mathsf{V}_{\bar{\eta}, r} = \{x \colon V_{\bar{\eta}}(x) \leq r\} = \{x \colon \|x\| \leq \sqrt{\log r / \bar{\eta}}\} \ .$$

The result above allows to state the following ergodicity result for $\mathsf{K}_{N,\gamma}$ kernel.

Theorem 17. *Assume **H** 1, **H** 2, and **A**1,**A**2 with $V_{\bar{\eta}}$ defined in* (19). *Then there exist $\bar{\Gamma}$ (defined in* (23)*), such that for any $\gamma \in (0, \bar{\Gamma}]$, $x \in \mathbb{R}^d$, and $k \in \mathbb{N}$,*

$$\| \mathsf{K}_{N,\gamma}^k(x, \cdot) - \pi \|_V \leq c_N \{\pi(V_{\bar{\eta}}) + V_{\bar{\eta}}(x)\} \tilde{\kappa}_N^k \ ,$$

where $V_{\bar{\eta}}$ is defined in (19)*, and the constants c_N, $\tilde{\kappa}_N \in [0, 1)$ are given by*

$$\begin{aligned}
&\log \tilde{\kappa}_N = \frac{\log(1 - \epsilon_{r_N, N}) \log \bar{\lambda}}{\log(1 - \epsilon_{r_N, N}) + \log \bar{\lambda} - \log \bar{b}_N} \ , \quad r_N = 1 \vee \{4 b_N / (1 - \lambda) - 1\} \ , \\
&\epsilon_{r_N, N} = (N - 1) \pi(\mathsf{V}_{\bar{\eta}, r_N}) / [2 w_{\infty, r_N} + N - 2], \quad b_N = \mathsf{b}_{\mathsf{P}_N} + \bar{b}_{\bar{\Gamma}}^{\mathrm{M}} \ , \\
&c_N = (\lambda + \bar{b}_N)(1 + \bar{b}_N / [2(1 - \epsilon_{r_N, N})(1 - \bar{\lambda})]) \\
&\bar{\lambda} = (1 + \lambda)/2 \ , \quad \bar{b}_N = \lambda r_N + b_N \ ,
\end{aligned} \tag{24}$$

and λ is defined in (23).

Proof. The proof follows from the combination of Theorem 2 and Proposition 14. □

To derive the geometric ergodicity rates in Theorem 17, it is not required to identify the small sets of the MALA rejuvenation kernel R_γ. The only quantity of interest is the Foster-Lyapunov drift condition satisfied by $R_\gamma^{\lceil 1/\gamma \rceil}$. Theorem 16 implies that the rate of convergence of MALA is $\gamma \log \rho_{\bar{\Gamma}}$. The following statement allows to quantify the improvement in the convergence rate of $\mathsf{K}_{N,\gamma}$ compared to $R_\gamma^{\lceil 1/\gamma \rceil}$. Following [58], we consider the relative improvement of the *mixing time* of the considered Markov kernels. To introduce formally the mixing time, we need an auxiliary definition of the V-Dobrushin coefficient. We refer the reader to [21, Section 18.3] for more detailed exposition. Recall that $M_{1,V}(\mathbb{X})$ is a set of probability measures on $(\mathbb{X}, \mathcal{X})$, such that $\xi(V) < \infty$.

Definition 18 (V-Dobrushin coefficient)**.** *Let $V : \mathbb{X} \mapsto [1; +\infty)$ be a measurable function, and Q be a Markov kernel on $(\mathbb{X}, \mathcal{X})$, such that $\xi(V) < \infty$ implies $\xi\mathsf{Q}(V) < \infty$ for any measure $\xi \in M_{1,V}(\mathbb{X})$. Then the V-Dobrushin coefficient of the Markov kernel Q, is defined by*

$$\Delta_V(\mathsf{Q}) = \sup_{\xi \neq \xi' \in M_{1,V}(\mathbb{X})} \frac{\|\xi\mathsf{Q} - \xi'\mathsf{Q}\|_V}{\|\xi - \xi'\|_V} .$$

It is known (see e.g. [21, Theorem 18.4.1]), that V-geometric ergodicity of the Markov kernel Q (see Definition 1) is equivalent to the fact, that

$$\Delta_V(\mathsf{Q}^m) \leq 1 - \varepsilon$$

for some $m \in \mathbb{N}^*$ and $0 < \varepsilon < 1$.

Definition 19. *Let Q be V-geometrically ergodic Markov kernel. Then the corresponding mixing time $t_{\mathrm{mix}} \in \mathbb{N}^*$ is defined as*

$$t_{\mathrm{mix}} = \inf_{m \in \mathbb{N}^*} \{m : \Delta_V(\mathsf{Q}^m) \leq 1/4\} .$$

Note that if Q is V-geometrically ergodic with factor $0 < \rho < 1$ given in $Definition$ 1, its mixing time t_{mix} is bounded as $t_{\mathrm{mix}} \leq (\log(1/\rho))^{-1} \log(4M)$.

Now we compare the mixing time of $\mathsf{K}_{N,\gamma}$, which is inversely proportional to $\log(1/\tilde{\kappa}_N)$, to the mixing time of $R_\gamma^{\lceil 1/\gamma \rceil}$, which is inversely proportional to $\log(1/\rho_{\bar{\Gamma}})$.

Theorem 20. *Assume **H** 1-**H** 2 and **A1**-**A2** with $V_{\bar{\eta}}$. Then there exist $\bar{\Gamma}$ (defined in* (23)*), such that for any $\gamma \in \left(0, \bar{\Gamma}\right]$, it holds that*

$$\lim_{N \to \infty} \frac{\log(\rho_{\bar{\Gamma}})}{\log(\tilde{\kappa}_N)} = \frac{\log(1 - 2^{-1}\varepsilon(K_{\bar{\Gamma}}))}{\log(1 - \epsilon_\infty)} \times \frac{\log(1 - 2^{-1}\varepsilon(K_{\bar{\Gamma}})) + \log \bar{\lambda} - \log \bar{b}_{\bar{\Gamma}}^{\mathrm{M}}}{\log(1 - \epsilon_\infty) + \log \bar{\lambda} - \log \bar{b}_\infty} , \qquad (25)$$

where $\lambda, \bar{\lambda}$, and $\bar{b}_{\bar{\Gamma}}^{\mathrm{M}}$ are defined in (23)*, $\varepsilon(\cdot)$ is defined in* (22)*, and*

$$r_\infty = 1 \vee \{4b_\infty/(1 - \lambda) - 1\} , \quad \epsilon_\infty = \pi(\mathsf{V}_{\bar{\eta}, r_\infty}) , \quad b_\infty = \mathsf{b}_{\mathsf{K}_\infty} + \bar{b}_{\bar{\Gamma}}^{\mathrm{M}} , \quad \bar{b}_\infty = \lambda r_\infty + b_\infty .$$

Proof. The proof follows by combining the expressions (23) and (24). □

The ratio $\log(1 - 2^{-1}\varepsilon(K_{\bar{\Gamma}}))/\log(1 - \epsilon_\infty)$ is extremely small in most settings. This explains the observed behavior: the mixing time of the Ex2MCMC kernel is much smaller than the mixing time of the MALA algorithm, which we observe in practice in all the examples we discuss. The difference is even more spectacular when the dimension increases. To illustrate this phenomenon, we consider the following numerical scenario for (25). We assume that **H** 1-**H** 2 holds with $\mathtt{m} = 0.1, \mathtt{M} = 2.0, \mathtt{L} = 1.0$, and $\mathtt{K} = 5.0$. One can evaluate that even for $d = 2$ the respective value $K_{\bar{\Gamma}} \approx 10^3$. We now show, how the bound for $K_{\bar{\Gamma}}$ scales with the dimension d. The respective plot for $d \in [2; 100]$ is given in Figure 6. It implies that $K_{\bar{\Gamma}}$ grows as $\sqrt{d}$. At the same time, the standard bound $\mathbf{\Phi}(-x) \leq \exp\{-x^2/2\}$, valid for $x \geq 0$, yields that $\varepsilon(K_{\bar{\Gamma}})/2 \leq \exp\{-(3/2)(L + 1)K_{\bar{\Gamma}}^2\}$. At the same time, ϵ_∞ typically does not decrease with the growth of d due to the construction of r_∞. Hence, the ratio (25) decreases exponentially with the growth of d in our model scenario.

D Proof of Theorem 3

The proof relies on results of stochastic approximation with Markovian dynamics, see e.g. [7, 8]. For reader's convenience, before going into the details, we give an outline of the proof. The motivation of such algorithms is to find the roots of the function $h : \Theta \to \mathbb{R}^q$, $\Theta \subset \mathbb{R}^q$

$$h(\theta) = \int_{\mathbb{U} \times \mathbb{E}} H(\theta, u, e)\mu(\mathrm{d}e)\rho_\theta(\mathrm{d}e) ,$$

for families of functions $\{H(\theta, u, e) : \Theta \times \mathbb{U} \times \mathbb{E} \to \Theta\}$, a family of probability distributions $\{\rho_\theta, \theta \in \Theta$ of $(\mathbb{E}, \mathcal{E})$ and a probability distribution μ on a space $(\mathbb{U}, \mathcal{U})$. These roots are not available analytically and a way of finding them numerically consists of considering the controlled Markov

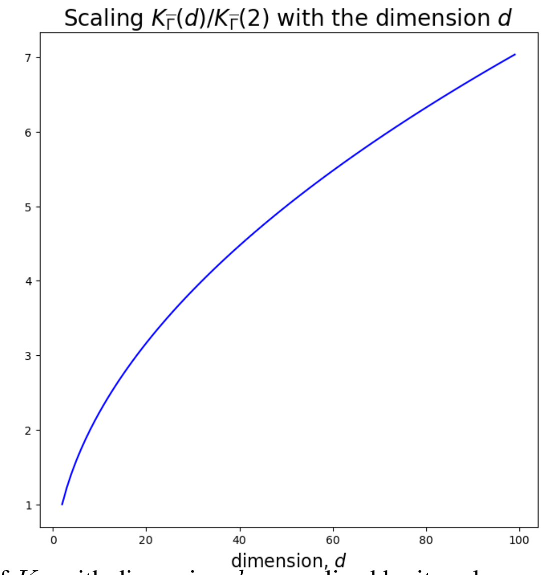


Figure 6: Scaling of $K_{\bar{\Gamma}}$ with dimension d, normalized by its value corresponding to $d = 2$.

chain on $\left\{(\Theta \times \mathbb{U})^{\mathbb{N}}, (\mathcal{B}(\Theta) \otimes \mathcal{U})^{\otimes \mathbb{N}}\right\}$ initialized at some $(\theta_0, U_0) = (\vartheta, u) \in \Theta \times \mathbb{U}$ and defined recursively for a sequence of stepsize $\{\gamma_i,\ i \in \mathbb{N}\}$ by

$$\begin{aligned} &U_{i+1} \sim P_{\theta_i}(U_i, \cdot)\ , \quad E_{i+1} \sim \rho_{\theta_i} \\ &\theta_{i+1} = \theta_i + \gamma_{i+1} H(\theta_i, U_{i+1}, E_{i+1})\ . \end{aligned}$$

Here $\{P_\theta, \theta \in \Theta\}$ is a family of Markov kernels such that for each $\theta \in \Theta$, $\mu P_\theta = \mu$. The rationale for this recursion goes as follows. Let us first rewrite the Robbins-Monro recursion

$$\theta_{i+1} = \theta_i + \gamma_{i+1}\{h(\theta_i) + \xi_{i+1}\},$$

where $\xi_{i+1} = H(\theta_i, U_{i+1}, E_{i+1})$ is referred to as the "noise". Therefore, $\{\theta_i\}$ is a noisy version of the sequence $\{\bar{\theta}_i\}$ defined as $\bar{\theta}_{i+1} = \bar{\theta}_i + \gamma_{i+1} h(\bar{\theta}_i)$. The convergence of such sequences has been studied by many authors, starting with [48] under various conditions. A crucial step of such convergence analysis consists of assuming that the sequence $\{\theta_i\}$ remains bounded with probability 1 in a compact set of Θ. This problem has traditionally can be circumvented by means of modifications of the recursion. Indeed, one of the major difficulties specific to the Markovian dynamic scenario is that $\{\theta_i\}$ governs the ergodicity of the controlled Markov chain $\{U_i\}$ and that stability properties of $\{\theta_i\}$ require "good" ergodicity properties which might vanish whenever $\{\theta_i\}$ approaches $\partial\Theta$ often away from the roots of $h(\theta)$, resulting in instability. Most existing results rely on modifications of the updates designed to ensure a form of ergodicity of $\{\xi_i\}$ which in turn ensures that $\{\theta_i\}$ inherits the stability properties of $\{\bar{\theta}_i\}$; see e.g. [7, 10] and the discussion in [8, Section 3]. We follow here [10]. Let $\{\mathcal{R}_i\}$ be a sequence of compact subsets of Θ and consider the recursion:

$$\begin{aligned} &U_{i+1} \sim P_{\theta_i}(U_i, \cdot) \quad E_{i+1} \sim \rho_{\theta_i} \\ &\theta^*_{i+1} = \theta_i + \gamma_{i+1} H(\theta_i, U_{i+1}, E_{i+1}) \\ &\theta_{i+1} = \theta^*_{i+1} 1_{\mathcal{R}_{i+1}}(\theta^*_{i+1}) + \theta^{\text{proj}}_{i+1} 1_{\mathcal{R}^c_{i+1}}(\theta^*_{i+1}) \end{aligned}$$

where, denoting $\mathcal{F}_i = \sigma(U_0, \theta_j, j \le i)$, $\theta^{\text{proj}}_{i+1}$ is a random variable measurable w.r.t $\mathcal{F}_i \vee \sigma(\theta^*_{i+1})$.

Using the results in [10], we aim to show that the SA-generated sequence $\{\theta_i\}$ remains in a feasible set Θ and do not approach $\partial\Theta$ with probability one for arbitrary initialization $(\theta_0, u) \in \mathcal{R}_0 \times \mathbb{U}$ under appropriate conditions on $\{H(\theta, u, e), (\theta, u, e) \in \Theta \times \mathbb{U} \times \mathbb{E}\}$, $\{P_\theta, \theta \in \Theta\}$ and $\{\mathcal{R}_i\}$. We denote throughout the probability distribution associated to the process $(\theta_i, U_i)_{i \ge 0}$ defined in Algorithm 1.1 and starting at $(\theta_0, U_0) \equiv (\theta, u) \in \Theta \times \mathbb{U}$ as $\mathbb{P}_{\theta,u}(\cdot)$ and the associated expectation as $\mathbb{E}_{\theta,u}[\cdot]$. The approach developed in [10] relies on the existence of a Lyapunov function $w : \Theta \to [0, \infty)$ for the recursion on θ and the subsequent proof that $\{w(\theta_i)\}$ is $\mathbb{P}_{\theta,u}$-a.s. under some adequate level. For any $M > 0$, we define the level sets $\mathcal{W}_M := \{\theta \in \Theta : w(\theta) \le M\}$. Following [10], we consider the following assumptions:

SA1. *There exists a continuously differentiable function $w : \Theta \to [0, \infty)$ such that*

(i) *For all $\theta, \theta' \in \Theta$,*
$$\|\nabla w(\theta) - \nabla w(\theta')\| \leq C_w \|\theta - \theta'\| \ .$$

(ii) *the projection sets are increasing subsets of Θ, that is, $\mathcal{R}_i \subset \mathcal{R}_{i+1}$ for all $i \geq 0$, and*
$$\hat{\Theta} := \bigcup_{i=0}^{\infty} \mathcal{R}_i \subset \Theta \ ,$$

(iii) *there exists a constant $M_0 > 0$ such that for any $\theta \in \mathcal{W}_{M_0}^{\mathrm{c}} \cap \hat{\Theta}$*
$$\langle \nabla w(\theta), h(\theta) \rangle \leq 0$$

(iv) *the family of random variables $\left\{\theta_i^{\mathrm{proj}}\right\}_{i \geq 1}$ satisfies for all $i \geq 1$ whenever $\theta_i^* \notin \mathcal{R}_i$*
$$\theta_i^{\mathrm{proj}} \in \mathcal{R}_i \quad \textit{and} \quad w\left(\theta_i^{\mathrm{proj}}\right) \leq w\left(\theta_i^*\right) \quad \mathbb{P}_{\theta,u} - \textit{a.s.}.$$

(v) *there exists constants $c \in [0, \infty)$ and a non-decreasing sequence of constants $\zeta_i \in [1, \infty)$ satisfying $\sup_{\theta \in \mathcal{R}_i} |\nabla w(\theta)| \leq c\zeta_i$ for all $i \geq 0$.*

Following [10], we introduce $\bar{H}(\theta, u, e) := H(\theta, u, e) - h(\theta)$. We need to impose some additional constraints on the noise sequence:

SA2. *For any $(\theta, u) \in \mathcal{R}_0 \times \mathbb{U}$ it holds that*

(i) $\mathbb{P}_{\theta,u}\left(\lim_{i\to\infty} \gamma_{i+1} \|\nabla w(\theta_i)\| \cdot \|H(\theta_i, U_{i+1}, E_{i+1})\| = 0\right) = 1,$
(ii) $\mathbb{E}_{\theta,u}\left[\sum_{i=0}^{\infty} \gamma_{i+1}^2 \|H(\theta_i, U_{i+1}, E_{i+1})\|^2\right] < \infty,$
(iii) $\mathbb{E}_{\theta,u}\left[\sup_{k\geq 0}\left|\sum_{i=0}^{k} \gamma_{i+1}\left\langle \nabla w(\theta_i), \bar{H}(\theta_i, U_{i+1}, E_{i+1})\right\rangle\right|\right] < \infty.$
(iv) $\lim_{\theta \to \partial\hat{\Theta}} w(\theta) = \infty$

Theorem 21. *Assume **SA1**-**SA2**. Then, for any $(\theta, u) \in \mathcal{R}_0 \times \mathrm{U}$*
$$\mathbb{P}_{\theta,u}\left(\limsup_{i\to\infty} w\left(\theta_i\right) < \infty\right) = 1.$$

Proof. The proof is a simple adaptation of [10, Theorem 2.5]. □

The condition $\lim_{\theta \to \partial\hat{\Theta}} w(\theta) = \infty$ is weakened in [10, Section 2.2]. Verifiable conditions implying **SA**2 are given in [10, Section 3, Condition 3.1]. They are summarized in the next assumption. In the assumptions below, it is implicitly assumed that **SA**1 holds with constants $(\zeta_i)_{i\geq 0}$.

We denote $\tilde{H}(\theta, u) = \int \bar{H}(\theta, u, e)\rho(\mathrm{d}e)$ and we consider the following assumptions:

SA3. *For all $\theta \in \hat{\Theta}$, the solution $g_\theta : \mathbb{U} \to \Theta$ to the Poisson equation $g_\theta(u) - P_\theta g_\theta(u) \equiv \tilde{H}(\theta, u)$ exists and for all $i \geq 0$ the step size Γ_{i+1} is independent of $\mathcal{F}_i$ and U_{i+1}. Moreover, there exist a measurable function $V : \mathbb{U} \to [1, \infty)$ and constants $c < \infty, \beta_H, \beta_g \in [0, 1/2]$ and $\alpha_g, \alpha_H, \alpha_V \in [0, \infty)$ such that for all $(\theta, u) \in \mathcal{R}_0 \times \mathbb{U}$*

(i) $\sup_{\theta \in \mathcal{R}_i} |\tilde{H}(\theta, u)| \leq c\zeta_i^{\alpha_H} V^{\beta_H}(u),$
(ii) $\mathbb{E}_{\theta,u}\left[V(U_i)\right] \leq c\zeta_i^{\alpha_V} V(u),$
(iii) $\sup_{\theta \in \mathcal{R}_i}\left[|g_\theta(u)| + |P_\theta g_\theta(u)|\right] \leq c\zeta_i^{\alpha_g} V^{\beta_g}(u),$
(iv) $\sum_{i=1}^{\infty} \gamma_{i+1}\ \zeta_i \mathbb{E}_{\theta,u}\left[\left|P_{\theta_i} g_{\theta_i}(U_i) - P_{\theta_{i-1}} g_{\theta_{i-1}}(U_i)\right|\right] < \infty,$
(v) $\sum_{i=1}^{\infty} \gamma_i^2 \zeta_i^{2+2((\alpha_H + \beta_H \alpha_V) \vee (\alpha_g + \beta_g \alpha_V))} < \infty,$
(vi) $\sum_{i=1}^{\infty} \gamma_{i+1} \gamma_i \zeta_i^{\alpha_H + \alpha_g + (\beta_H + \beta_g)\alpha_V} < \infty,$
(vii) $\sum^{\infty} |\gamma_{i+1} - \gamma_i|\, \zeta_i^{1 + \alpha_g + \beta_g \alpha_V} < \infty.$

For geometrically ergodic Markov chain, these conditions may be shown to boil down to "uniform-in-θ" geometric ergodicity conditions and "smoothness" of the mapping $\theta \mapsto P_\theta$.

MC1. *For any $r \in (0,1]$ and any $\theta \in \hat{\Theta}$, there exist constants $M_{\theta,r} \in [0,\infty)$ and $\rho_{\theta,r} \in (0,1)$, such that for any function $\|f\|_{V^r} < \infty$*

$$\left|P_\theta^k f(u) - \mu_\theta(f)\right| \le V^r(u)\|f\|_{V^r} M_{\theta,r}\rho_{\theta,r}^k$$

for all $k \geq 0$ and all $u \in \mathbb{U}$. Moreover, it holds that $\sup_{\theta\in\mathcal{R}_i} M_{\theta,r} \leq c_r\zeta_i^{\alpha_M}$ and $\sup_{\theta\in\mathcal{R}_i}\left(1-\rho_{\theta,r}\right)^{-1} \leq c_r\zeta_i^{\alpha_\rho}$.

MC2. *For any $\theta, \theta' \in \hat{\Theta}$, there exist a constant $D_{\theta,\theta',r} \in [0,\infty)$ and a constant $\beta_D \in (0,\infty)$ independent of θ, θ' and r such that for any function $\|f\|_{V^r} < \infty$*

$$\|P_\theta f - P_{\theta'} f\|_{V^r} \leq \|f\|_{V^r} D_{\theta,\theta',r}\left|\theta-\theta'\right|^{\beta_D} .$$

Moreover, $\sup_{(\theta,\theta')\in\mathcal{R}_i^2} D_{\theta,\theta',r} \leq c_r^D\zeta_i^{\alpha_D}$ for some constant $c_r^D \in [0,\infty)$ depending only on $r \in (0,1]$

MC3. ***SA**3-(i) and (ii) hold with constants α_H, β_H and α_V, and there exist constants $c < \infty, \alpha_\Delta \in [0,\infty)$ and $\beta_\Delta > 0$ such that*

$$\sup_{(\theta,\theta')\in\mathcal{R}_i^2}\left\|\tilde{H}(\theta,\cdot) - \tilde{H}\left(\theta',\cdot\right)\right\|_{V^{\beta_H}} \leq c\zeta_i^{\alpha_\Delta}\left|\theta-\theta'\right|^{\beta_\Delta} .$$

Up to this point, we have only considered the stability of the stochastic approximation process with expanding projections. Indeed, after showing the stability we know that the projections can occur only finitely often (almost surely), and the noise sequence can typically be controlled. Given this, the stochastic approximation literature provides several alternatives to show the convergence; see [41, 15]. We formulate below a convergence result following from [7].

SA4. *The set $\Theta \subset \mathbb{R}^d$ is open, the mean field $h : \Theta \to \mathbb{R}^d$ is continuous, and there exists a continuously differentiable function $\hat{w} : \Theta \to [0,\infty)$ such that*

(i) there exists a constant $M_0 > 0$ such that

$$\mathcal{L} := \{\theta \in \Theta : \langle\nabla\hat{w}(\theta), h(\theta)\rangle = 0\} \subset \{\theta \in \Theta : \hat{w}(\theta) < M_0\}$$

(ii) there exists $M_1 \in (M_0,\infty]$ such that $\{\theta \in \Theta : \hat{w}(\theta) \leq M_1\}$ is compact.

(iii) for all $\theta \in \Theta \setminus \mathcal{L}$, the inner product $\langle\nabla\hat{w}(\theta), h(\theta)\rangle < 0$ and the closure of $\hat{w}(\mathcal{L})$ has an empty interior.

Theorem 22. *Assume **SA**4 holds, and let $\mathcal{K} \subset \Theta$ be a compact set intersecting $\mathcal{L}$, that is, $\mathcal{K} \cap \mathcal{L} \neq \emptyset$. Suppose that $\left(\gamma_i\right)_{i\geq1}$ is a sequence of non-negative real numbers satisfying $\lim_{i\to\infty}\gamma_i = 0$ and $\sum_{i=1}^\infty\gamma_i = \infty$. Consider the sequence $\left(\theta_i\right)_{i\geq0}$ taking values in Θ and defined through the recursion $\theta_i = \theta_i - 1 + \gamma_i h\left(\theta_{i-1}\right) + \gamma_i\varepsilon_i$ for all $i \geq 1$, where $\left(\varepsilon_i\right)_{i\geq1}$ take values in $\mathbb{R}^d$. If there exists an integer i_0 such that $\{\theta_i\}_{i\geq i_0} \subset \mathcal{K}$ and $\lim_{m\to\infty}\sup_{n\geq m}\left|\sum_{i=m}^n\gamma_i\varepsilon_i\right| = 0$, then $\lim_{n\to\infty}\inf_{x\in\mathcal{L}\cap\mathcal{K}}\|\theta_n - x\| = 0$.*

We have now all the necessary elements to prove Theorem 3. For simplicity, we set $\alpha_k = \alpha_\infty$ for any $k \in \mathbb{N}$ and $\gamma_k = 1/(1+k)^\iota$ where $\iota \in (1/2, 1]$. In this case, the state space is $\mathbb{U} = \mathbb{X}^M$ and $\mathbb{E} = \mathbb{Z}^{(N-1)\cdot M}$, $U_k = (Y_k[j])_{j=1}^M$, $E_k = (Z_k^{2:N}[j])_{j=2}^N$. With $u = (y[j])_{j=1}^M$ and $e = (z^{2:N}[j])_{j=1}^M$, $H(\theta, u, e)$ is given by

$$H(\theta, u) = M^{-1}\sum_{mj=1}^N\left\{\alpha_\infty H^f(\theta, y[j], z^{2:N}[j]) + (1-\alpha_\infty)H^b(\theta, z^{2:N}[j])\right\} .$$

where H^f and H^b are defined respectively in (4) and (5). In this case, the Markov kernel P_θ is given for any nonnegative function f,

$$P_{\theta,N}f(y[1],\ldots,y[M]) = \int\prod_{j=1}^N \mathsf{K}_{\theta,N}(y[j], \mathrm{d}\tilde{y}[j])f(\tilde{y}[1],\ldots,\tilde{y}[M]) ,$$

and $\mathsf{K}_{\theta,N}$ is defined in (2.2) with $\lambda \leftarrow \lambda_\theta$ and $w \leftarrow w_\theta$. By construction, for any $\theta \in \Theta$, P_θ has a unique stationary distribution which is given by $\mu = \pi^{\otimes M}$. Using Theorem 6, and, for all $\theta \in \Theta$,

$$H^f(\theta, x^{1:N}) = \Pi_{\theta,N}[\nabla_\theta\log\lambda_\theta](x^{1:N})$$

we get that

$$h(\theta) = -\alpha_\infty \nabla_\theta \mathrm{KL}(\pi||\lambda_\theta) - (1-\alpha_\infty)\nabla_\theta \mathrm{KL}(\lambda_\theta||\pi)\ .$$

Recall that $\Theta = \mathbb{R}^q$. To check **SA**1, we set

$$w(\theta) = \alpha_\infty \mathrm{KL}(\pi||\lambda_\theta) - (1-\alpha_\infty)\mathrm{KL}(\lambda_\theta||\pi)\ \text{, for } \theta \in \Theta.$$

and for $i \in \mathbb{N}$, $\zeta_i = \log(i+1)$. The subset $\mathcal{R}_i$ is a ball centered at 0 and of radius r_i where r_i is chosen so that $\sup_{\|\theta\| \le r_i} \nabla w(\theta)\| \le c\zeta_i$ (such r_i exists using **A**3). It is easily checked that **SA**1 is satisfied thanks to **A**3 (note in particular that ∇w is globally Lipshitz under the stated conditions). Conditions **SA**3-(v)-(vi)-(vii) are automatically satisfied.

We choose the drift function for the Markov kernel $P_{\theta,N}$ as

$$V(y[1],\ldots,y[M]) = \sum_{i=1}^{M} V(y[i])\ ,$$

where V is the drift function in **A**1. **MC**1 follows from Theorem 2 under **A**4. It is important to note that it is essential to have explicit controls on the drift and reduction conditions here. Conditions **MC** 2 and **MC**2 follow from **A**3. The precise tuning of constants is done along the same lines as [10, Section 5.3].

E Numerical experiments

E.1 Metrics

ESTV To compute Empirical sliced total variation distance (ESTV), we perform 25 random one-dimensional projections and then perform Kernel Density Estimation there for reference and produced samples. We then take the 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.

EMD We compute the EMD as the transport cost between sample and reference points in L_2 using the algorithm proposed in [14]. Then we report the EMD rescaled by the target dimension d.

ESS ESS (effective sample size) measures how many independent samples from target yield (approximately) the same variance for estimating the mean of some function. The closer ESS is to 1, the better is the sampler. Following [26], we compute ESS component-wise for multivariate distributions. Namely, given a sample $\{Y_t\}_{t=1}^M, Y_t \in \mathbb{R}^d$ of size M, for $i = 1,\ldots,d$, we compute

$$\mathrm{ESS}_i = \frac{1}{1 + \sum_{k=1}^M \rho_k^{(i)}}\ .$$

Here $\rho_k^{(i)} = \frac{\mathrm{Cov}(Y_{t,i}, Y_{t+k,i})}{\mathrm{Var}(Y_{t,i})}$ is the autocorrelation at lag k for $i-$th component. We replace $\rho_k^{(i)}$ by its sample counterpart $\widehat{\rho}_k^{(i)}$, an report $\mathrm{ESS} = d^{-1}\sum_{i=1}^d \widehat{\mathrm{ESS}}_i$, where

$$\widehat{\mathrm{ESS}}_i = \frac{1}{1 + \sum_{k=1}^M \widehat{\rho}_k^{(i)}}\ .$$

E.2 Unimodal Gaussian target and impact of dimension

With the simple experiment presented on Figure 7, we illustrate the sensitivity of the purely global i-SIR to the match between the proposal and target, which typically worsens with dimension. Namely, the rate κ_N can be close to 1 when the dimension d is large, even when the restrictive condition that weights are uniformly bounded $|w|_\infty < \infty$ is satisfied.

To illustrate this phenomenon, we consider a simple problem of sampling from the standard normal distribution $\mathcal{N}(0, \mathrm{I}_d)$ with the proposal $\mathcal{N}(0, 2\,\mathrm{I}_d)$ in increasing dimensions d up to 300. Results visualized in Figure 7 show that the performance of vanilla i-SIR quickly deteriorates as most proposals get rejected. This problem can be tackled by using the Explore-Exploit strategy coupling i-SIR with local MCMC steps to define a new sampler. This simple experiment previously considers $\mathrm{Ex}^2\mathrm{MCMC}$ with MALA applied as R.

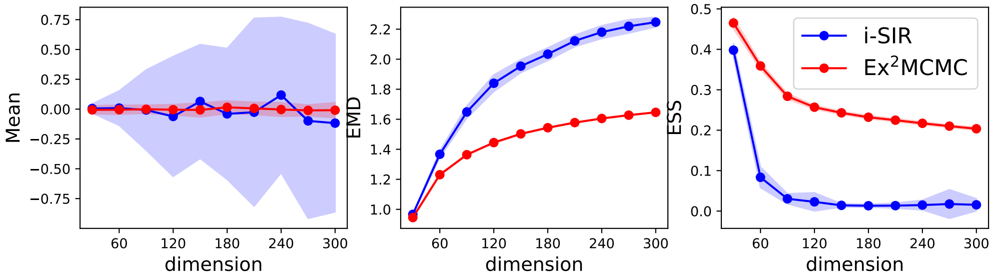
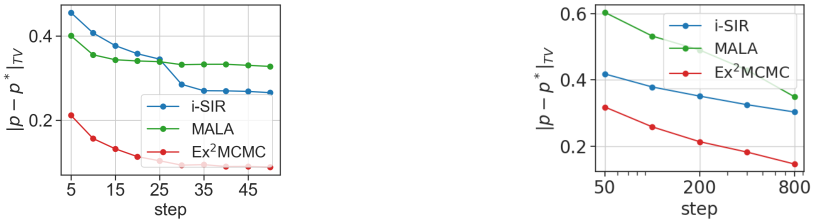


Figure 7: Sampling from $\mathcal{N}(0, \mathrm{I}_d)$ with the proposal $\mathcal{N}(0, 2\,\mathrm{I}_d)$. – See Appendix E.1 for the definitions of EMD and ESS metrics. We display confidence intervals for i-SIR and Ex^{2}MCMC obtained from 100 independent runs as blue and red regions, respectively. Ex^{2}MCMC helps to achieve efficient sampling even in high dimensions.



(a)

(b)

Figure 8: Inhomogeneous 2d Gaussian mixture. – Quantitative analysis during burn-in of parallel chains (a, $M = 500$ chains KDE) and for after burn-in for single chains statistics (b, $M = 100$ average).

E.3 Mixtures of Gaussians

Equally weighted Gaussians in two dimension The target density is

$$p_\beta(x) \propto \sum_{i=1}^{3} \beta_i \exp\{-\|x - \mu_i\|^2/(2\sigma^2)\} \; . \tag{26}$$

Here we choose $\sigma = 1$, $\beta_i = 1/3$, and μ_i, $i \in \{1, 2, 3, \}$ as vertices of an equilateral triangle with side length $4\sqrt{3}$ and center $(0, 0)$. The contour representation of (26) can be found in Figure 1a. We compare 3 sampling strategies:

- i-SIR algorithm with $N = 3$ particles and $\mathcal{N}(0, 4\,\mathrm{I})$ proposal distribution;
- MALA with step size $\gamma = 0.5$, tuned to obtain acceptance rate 0.67;
- Ex^{2}MCMC algorithm with the same parameters as i-SIR and 3 consecutive MALA steps with $\gamma = 0.5$ as rejuvenations.

We generate 100 observations within each sampler and represent them in Figure 1a. For the MALA sampler, we generate 300 samples and select every 3th to maintain compatibility with the Ex^{2}MCMC setup. Note that in this example, the variance of the global proposals in i-SIR should be relatively large to cover well all modes of the (26) mixture. However, since the modes are narrow, the step size of MALA cannot be very large to obtain a sensible acceptance rate. Therefore, Figure 1a shows the drawbacks of the two approaches: i-SIR covers all modes of the target, but the chain often gets stuck at a certain point, which affects the variability of the samples. MALA allows a better local exploration of each mode, but does not cover the whole support of the target. The Ex^{2}MCMC algorithm combines the advantages of both methods by combining i-SIR-based global exploration with MALA -based local exploration.

Now, the mixture model of (26) is modified with the weights parameters $\beta = (\beta_1, \beta_2, \beta_3) = (2/3, 1/6, 1/6)$ and same values of μ_i and σ. To compare the quality of the methods, we perform the following procedure

- starting with the initial distribution $\mathcal{N}(0, 4\,\mathrm{I})$, we generate the trajectory $(X_1, \ldots, X_n)$ for different values of $n \in [25, 800]$ for each of the compared methods (i-SIR MALA, $\mathrm{Ex}^2\mathrm{MCMC}$). Sampler hyperparameters are the same as above, and the burn-in period equals 50;
- We perform the kernel density estimate (KDE) $\widehat{p}_n$ based on the observations $(X_1, \ldots, X_n)$, and compute the total variation distance between $\widehat{p}_n$ and the target density p_β, and the forward $\mathrm{KL}(\widehat{p}_n || p_\beta)$. Then we average the results over 100 independent runs of each sampler.

Now we use the same values for the means and covariances but set the mixing weights to $\beta = (\beta_1, \beta_2, \beta_3) = (2/3, 1/6, 1/6)$. To compare the different sampling methods, we perform the following procedure.

- starting from the initial distribution $\mathcal{N}(0, 4\,\mathrm{I})$, we generate the trajectory $(X_1, \ldots, X_n)$ for different values of $n \in [25, 800]$ for each of the compared methods (i-SIR MALA, $\mathrm{Ex}^2\mathrm{MCMC}$). The hyperparameters of the sampler are the same as above, and the burn-in period is 50;
- We perform kernel density estimation (KDE) $\widehat{p}_n$ based on the observations $(X_1, \ldots, X_n)$ and calculate the total variation distance between $\widehat{p}_n$ and the target density p_β, as well as the forward value $\mathrm{KL}(\widehat{p}_n || p_\beta)$. We then average the results over 100 independent runs of each sampler.

The results for each sampler are given in Figure 1c, Figure 8b. We also provide a simple illustration to the statements of (2) and Theorem 2. Starting from the initial distribution $\xi \sim \mathcal{N}(0, 4\,\mathrm{I})$, we draw 500 independent chains of length 50 for each of the compared methods. Using these 500 observations, we create a KDE $\widehat{p}_n$ for the density corresponding to the distribution of $\xi \mathsf{Q}^n$ for different $n \in \{5, \ldots, 50\}$ and Q corresponding to i-SIR MALA or $\mathrm{Ex}^2\mathrm{MCMC}$Then we calculate the total variation distance between $\widehat{p}_n$ and the target density p_β. Corresponding plots can be found in Figure 1b, Figure 8a. Note that $\mathrm{Ex}^2\mathrm{MCMC}$ significantly outperforms the results of both MALA and i-SIRIndeed, the inhomogeneous mixture model is a complicated target for the Langevin-based methods. The trajectories generated by MALA tend to remain in a single mode of mixture (26), which reduces the reliability of the estimates and requires the generation of long trajectories even for $d = 2$. At the same time, it is difficult for i-SIR type methods without local exploration trajectories to quickly cover all the modes.

E.4 Normalizing flow RealNVP

We use the RealNVP architecture ([20]) for our experiments with adaptive MCMC. The key element of RealNVP is a coupling layer, defined as a transformation $f : \mathbb{R}^D \to \mathbb{R}^D$:

$$\begin{aligned} y_{1:d} &= x_{1:d} \\ y_{d+1:D} &= x_{d_1:D} \odot \exp(s(x_{1:d})) + t(x_{1:d}) \end{aligned}$$

where s and t are some functions from $\mathbb{R}^D$ to $\mathbb{R}^D$. Thus, it is clear that the Jacobian of such a transformation is a 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 the optimizer Adam ([38]) with $\beta_1 = 0.9,\ \beta_2 = 0.999$ and weight decay 0.01 to avoid overfitting.

E.5 High-dimensional multi-modal distribution

In an additional experiment we consider a high-dimensional toy target distribution: a Gaussian mixture similar as Appendix E.3 above in $50d$. Modes are equally weighted, isotropic and well-separated.

A purely local sampler would not mix between modes, as in the $2d$ case. A unimodal Gaussian proposal also fails in large dimension because of the concentration of the target measure in a small fraction of the proposal's bulk. Hence we only examine the performance of $\mathrm{FlEx}^2\mathrm{MCMC}$. We set the number of proposals per iterations to $N = 20$.

Using a RealNVP flow, we compare in Figure 9 the different outcomes depending on the choices of initialization of the MCMC walkers and training loss. Training the proposal offline through uniquely the backward KL (i.e. $\alpha = 0$ in the combinaison of KL losses) is typically unstable in this multimodal case and the network collapse on the first detected mode. Successful backward-KL training is probably possible, yet at the cost of designing a proper annealing schedule of the target distribution as in [76]. Resorting instead to a loss involving the forward KL ($\alpha = 0.9$ in this experiment), mixing

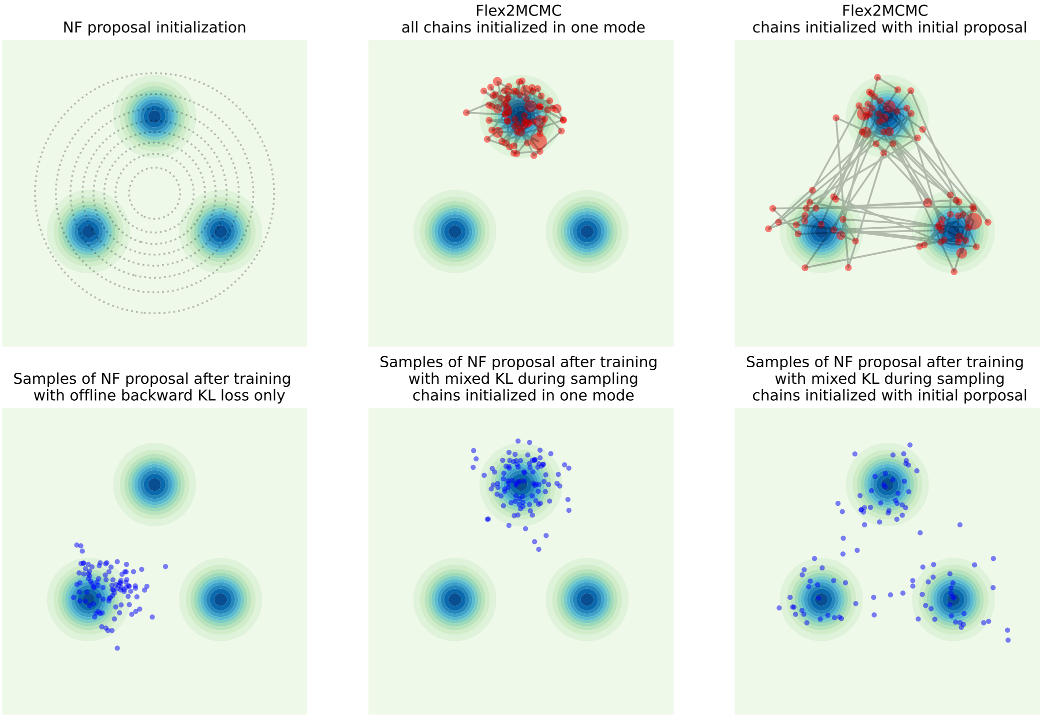


Figure 9: Importance of initialization and forward KL loss for multi-modal high-dimensional targets - All panels are $2d$ projections of a $50d$ Euclidian space, with a target mixture of 3 isotropic Gaussian. Using a normalizing flow proposal distribution initialized as an isotropic Gaussian covering the 3 modes (top left), training with backward KL loss only still typically leads to mode collapse on one of the modes (bottom left). Running instead the simulataneous training and sampling of $\mathrm{FlEx}^2\mathrm{MCMC}$with the mixture of backward and forward KL loss can lead to successful mixing between distant modes (top and bottom right), yet at the condition that chains are initialized such that all modes can be reached by the local-rejuvenation kernel- which is here enforced by an initialization as random draws of the initial proposal. Conversely, if all the chains are initialized in a single mode, the forward-KL estimated with states visited by the chains will not prevent a mode collapse (top and bottom center panels).

between the well separated modes in high-dimension is possible, provided that chain initialization ensures that all modes can be reached by the local kernel.

To summarize, the choice of loss composition depends on the information a priori available on the considered target distribution. If rough location of modes is available - as it might be the case in chemistry applications where isomers of interest are known but sampling is necessary for relative free energy calculations - relying on the forward KL to draw the proposal to the modes is a simple and efficient strategy. Conversely, if little is known, there is no free lunch with the local-global kernels and an annealing might be necessary to train the global proposal, possibly using only the backward KL loss.

E.6 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.

The corresponding densities are given for $x \in \mathbb{R}^d$ by

$$
\begin{aligned}
p_f(x) &= \mathrm{Z}^{-1} \exp\left(-x_1^2/2a^2 - (1/2)\mathrm{e}^{-2bx_1} \sum\nolimits_{i=2}^{d} \{x_i^2 + 2bx_1\}\right) , \quad d \geq 2, \\
p_b(x) &= \mathrm{Z}^{-1} \exp\left(- \sum\nolimits_{i=1}^{d/2} \{x_{2i}^2/2a^2 - (x_{2i-1} - bx_{2i}^2 + a^2 b)^2/2\}\right) , \quad d = 2k, k \in \mathbb{N} .
\end{aligned}
\tag{27}
$$

where Z is a normalizing constant. We set $a = 2$, $b = 0.5$ for funnel and $a = 5$, $b = 0.02$ for banana-shape distributions, respectively. For MALA we use an adaptive step size tuning strategy to maintain acceptance rate approximately 0.5. For i-SIR and Ex^{2}MCMC algorithms we use wide Gaussian global proposal $\mathcal{N}(0, \sigma_p^2 \mathrm{I})$ with $\sigma_p^2 = 4$ for Funnel and $\sigma_p^2 = 9$ for Banana-shape distribution.

For FlEx^{2}MCMC use a simple RealNVP-based normalizing flow [20] with 4 hidden layers. Note that for $p_f(x)$ the energy landscape in the region with $x_1 < 0$ is steep, so the distributions (27) are hard to capture, especially when the dimension d is large. Moreover, due to the complex geometry of the distribution support, we cannot hope that local samplers (MALA) or global samplers (i-SIR) alone will give good results. In this example, we want to compare FlEx^{2}MCMC with i-SIR MALA and the HMC-based NUTS sampler [35]. We also add a vanilla version of the Ex^{2}MCMC algorithm to the comparison. To generate the ground-truth samples, we use the explicit reparametrisation of (27). Indeed, given a random vector $(Z_1, \ldots, Z_d) \sim \mathcal{N}(0, \mathrm{I})$, we consider its transformation $(X_1, \ldots, X_d)$ under the formulas

$$
\begin{cases}
X_1 = aZ_1 \\
X_i = \mathrm{e}^{bX_1} Z_i \ , \quad i \in \{2, \ldots, d\} \ .
\end{cases}
$$

It is easy to check that $(X_1, \ldots, X_d)$ follows the density $p_f(x), x \in \mathbb{R}^d$. Similarly, for $d = 2k$ consider the transformation

$$
\begin{cases}
Y_{2i} = aZ_{2i} \\
Y_{2i-1} = Y_{2i} + bY_{2i}^2 - ba^2 \ , \quad i \in \{1, \ldots, k\} \ .
\end{cases}
$$

Then $(Y_1, \ldots, Y_d)$ follows the density $p_b(x), x \in \mathbb{R}^d$. We provide the average computation time for NUTS, adaptive i-SIR and FlEx^{2}MCMC algorithms in Table 1 and Table 2 for the Funnel and Banana-shape distributions, respectively, averaged over 50 runs. Note that different runs of NUTS algorithm yields high variance of the running time, especially for the Funnel distribution and dimensions $d \geq 50$.

We give the computation time for the above algorithms and additional implementation details in Appendix E.6. The implementation of FlEx^{2}MCMC is based on the use of 5 MALA steps as rejuvenation steps.

Method	$d = 10$	$d = 20$	$d = 50$	$d = 100$	$d = 200$
NUTS	33.4 ± 8.2	41.1 ± 12.3	61.6 ± 30.2	82.3 ± 73.2	88.4 ± 59.5
Adaptive i-SIR	38.1 ± 3.2	39.4 ± 2.8	45.3 ± 2.5	59.8 ± 0.7	80.4 ± 0.4
FlEx^{2}MCMC	46.8 ± 3.2	48.2 ± 2.8	54.2 ± 2.5	68.8 ± 0.8	89.5 ± 0.5

Table 1: Computational time for the Funnel distribution.

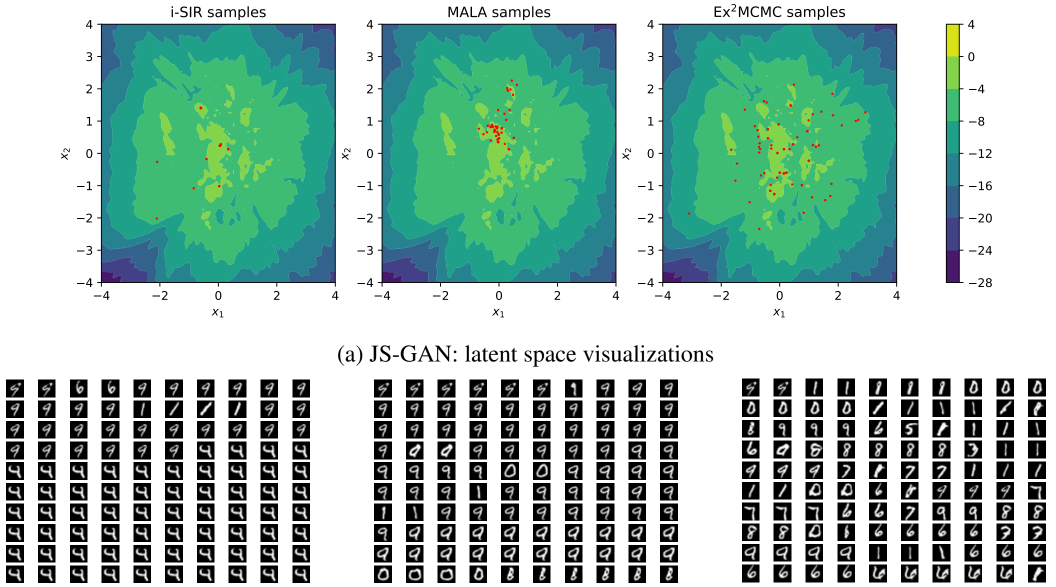
Method	$d = 20$	$d = 40$	$d = 60$	$d = 80$	$d = 100$
NUTS	27.6 ± 1.8	32.1 ± 1	34.2 ± 0.5	35.2 ± 0.5	35.9 ± 0.4
Adaptive i-SIR	24.5 ± 0.2	26.8 ± 0.3	28.5 ± 0.2	30.1 ± 0.2	32.8 ± 0.2
FlEx^{2}MCMC	39.3 ± 0.5	41.8 ± 0.3	43.5 ± 0.3	45.1 ± 0.3	47.8 ± 0.4

Table 2: Computational time for the Banana-shape distribution.

E.7 GANs as energy-based models

E.7.1 MNIST results

For this example, we consider both the Wasserstein GAN (WGAN) setup with energy function $E_W(z)$ and the classical Jensen-Shannon GAN with energy function $E_{JS}(z)$. In both cases, we use



(a) JS-GAN: latent space visualizations

(b) i-SIR samples

(c) MALA samples

(d) Ex^{2}MCMC samples

fully connected networks with 3 convolutional layers for discriminator and 3 linear + 3 convolutional layers for generator. For WGAN training, we use gradient penalty regularisation, following [31]. We provide additional visualisations of the latent space and samples along a given trajectory for Jensen-Shannon GAN in Appendix E.7.1 and for Wasserstein GAN in Appendix E.7.1. Sampling hyperparameters are summarized in Table 3. For fair comparison, we take each 3-rd sample produced by the MALA, when running this algorithm separately. Both for WGAN-GP and vanilla GAN experiments we apply i-SIR and Ex^{2}MCMC with wide Gaussian global proposal $\mathcal{N}(0, \sigma_p^2)$. The particular values of σ_p^2 are specified in Table 3.

E.7.2 Cifar-10 results

We consider two popular GAN architectures, DC-GAN [60] and SN-GAN [49]. Below we provide the details on experimental setup and evaluation for both of the models.

E.8 Training and sampling details.

For DC-GAN and SN-GAN experiments, we took the implementation and training script of the models from Mimicry repository `https://github.com/kwotsin/mimicry`. Both models were trained on a single GPU GeForce GTX 1060 for approximately 20 hours.

Both for DC-GAN and SN-GAN, the latent dimension is equal to $d = 128$. Following [17], for both models we consider sampling from the latent spatial distribution

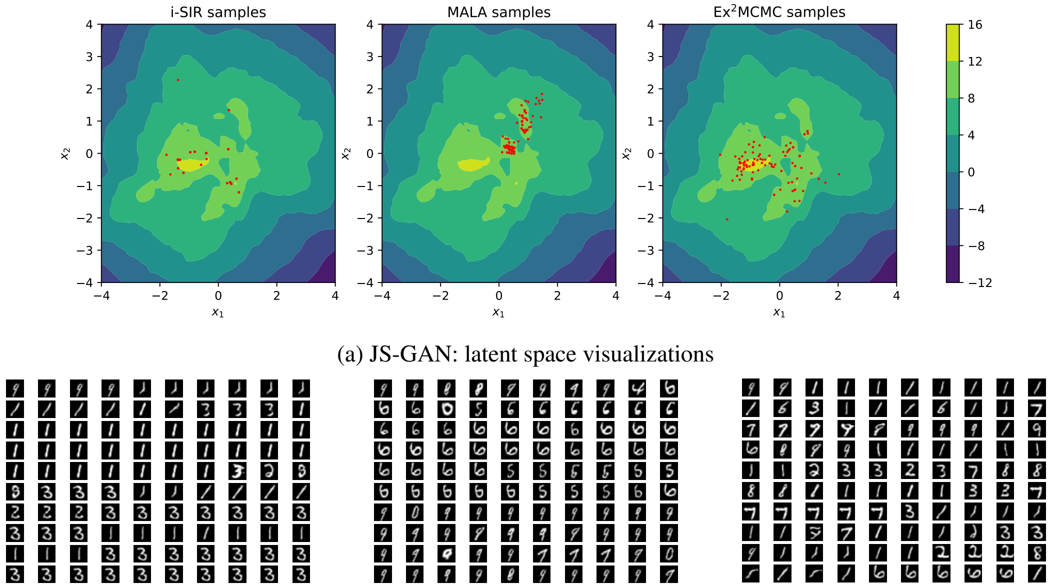
$$p(z) = \mathrm{e}^{-E_{JS}(z)}/Z \ , \quad z \in \mathbb{R}^d \ , \quad E_{JS}(z) = -\log p_0(z) - \operatorname{logit}\bigl(D(G(z))\bigr) \ ,$$

where $\operatorname{logit}(y) = \log\left(y/(1-y)\right)$ $y \in (0,1)$ is the inverse of the sigmoid function and $p_0(z) = \mathcal{N}(0, \mathrm{I})$.

Evaluation protocol We perform $n = 100$ iterations of the algorithms MALA, i-SIR Ex^{2}MCMC and FlEx^{2}MCMCḞor both the vanilla Ex^{2}MCMC algorithm (Algorithm 2) and FlEx^{2}MCMC we

Method	# iterations	MALA step size γ	# particles, N	σ_p^2	# MALA steps
JS-GAN	100	0.02	10	9	3
WGAN-GP	100	0.02	10	9	3

Table 3: MNIST hyperparameters.



(a) JS-GAN: latent space visualizations

(b) i-SIR samples

(c) MALA samples

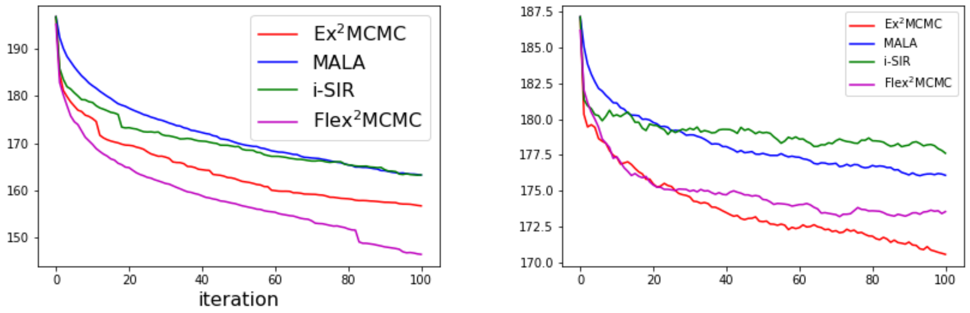
(d) Ex^{2}MCMC samples

use the Markov kernel (17), which corresponds to 3 MALA steps, as the rejuvenation kernel. The step size γ given for the algorithm Ex^{2}MCMC corresponds to its rejuvenation kernel MALA. For more experimental details, see Table 4. For i-SIR and Ex^{2}MCMC algorithms we use $\mathcal{N}(0, \sigma_p^2 \mathrm{I})$ with $\sigma_p^2 = 1$ as a global proposal distribution.

We run $M = 500$ independent chains for each of the above MCMC algorithms. Then, for the $j-$th iteration, we compute the average value of the energy function $E(z)$ averaged over M chains. Hyperparameters are specified in Table 4. Energy profiles for different algorithms for DC-GAN and SN-GAN are provided in Figure 17 and Figure 14, respectively. Note that in both cases Ex^{2}MCMC or FlEx^{2}MCMC algorithms yields lower energy samples. We visualize 10 randomly chosen trajectories obtained with each sampling methods in Figure 15-Figure 16 for SN-GAN and Figure 18-Figure 19 for DC-GAN, respectively. For each trajectory we visualize every 10-th sample. Both architectures indicate the same findings: MALA typically is not available to escape the mode of the corresponding target density $p(z)$ during one particular run. i-SIR travels well across the support of $p(z)$, yet the corresponding energy values are higher then the ones of Ex^{2}MCMC or FlEx^{2}MCMC . Some i-SIR trajectories can get trapped in one particular image due to the absence of local exploration moves. At the same time, Ex^{2}MCMC as illustrated in Figure 16-16a and Figure 19-19a, can both exploit the particular mode of the distribution and perform global moves over the support of $p(z)$. Of course, these global moves are more likely to occur during the first sampling iterations. For the DC-GAN architecture, we provide also the dynamics of FID (Frechet Inception Distance, [34]), and IS (Inception Score, [69]) values computed over 10000 independent trajectories. We plot the metrics in Figure 13a and Figure 13b. Metrics illustrate the image quality improvement achieved by FlEx^{2}MCMC and Ex^{2}MCMC algorithms.

GAN type	# iterations	MALA step size γ	# particles, N	σ_p^2	# MALA steps
SNGAN	100	5×10^{-3}	10	1	3
DCGAN	100	10^{-3}	10	1	3

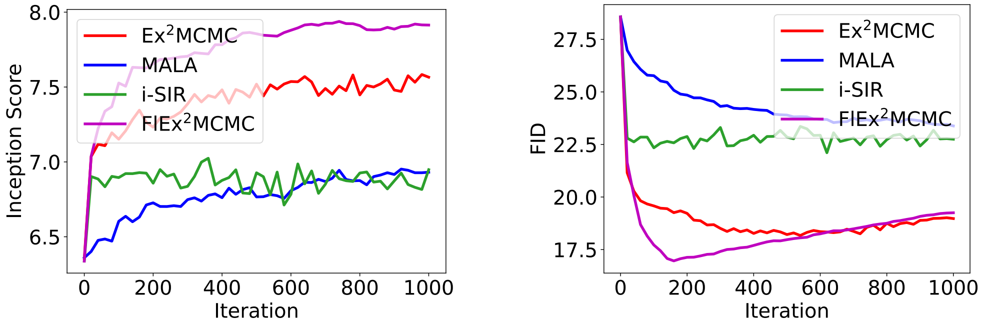
Table 4: CIFAR-10 hyperparameters.



(a) DC-GAN

(b) SN-GAN

Figure 12: Energy profile for DC-GAN and SN-GAN architectures on CIFAR-10 dataset.



(a) Inception Score dynamics for DC-GAN

(b) FID dynamics for DC-GAN

Figure 13: Dynamics of Inception Score (a) and FID (b) computed over 10000 independent trajectories for DC-GAN trained on CIFAR-10 dataset.

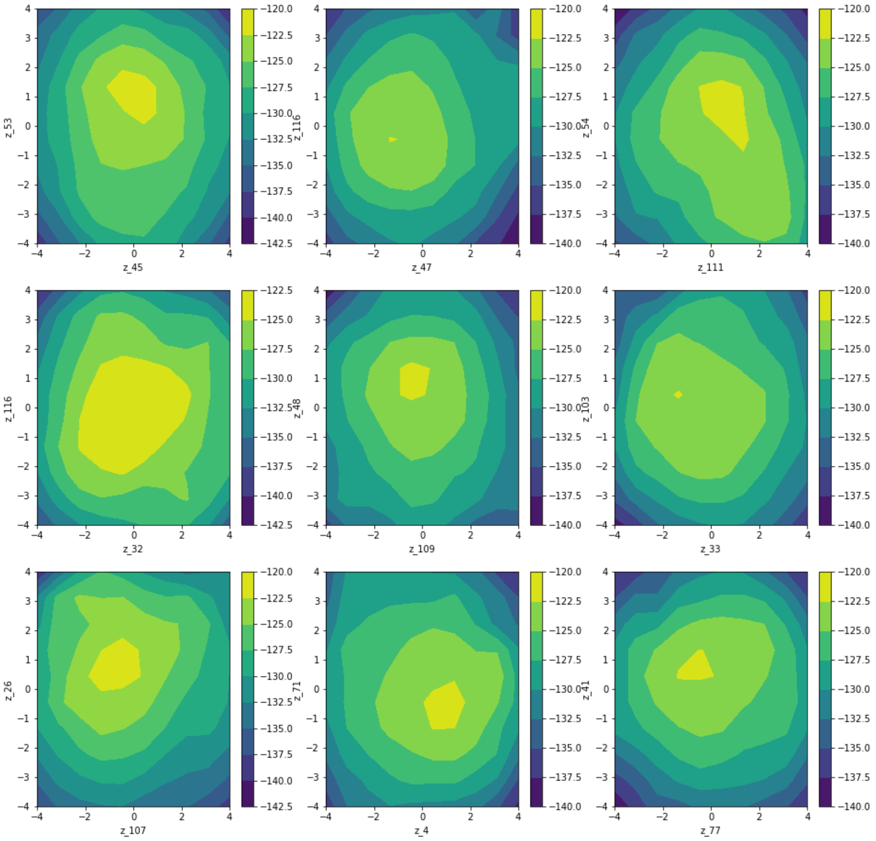
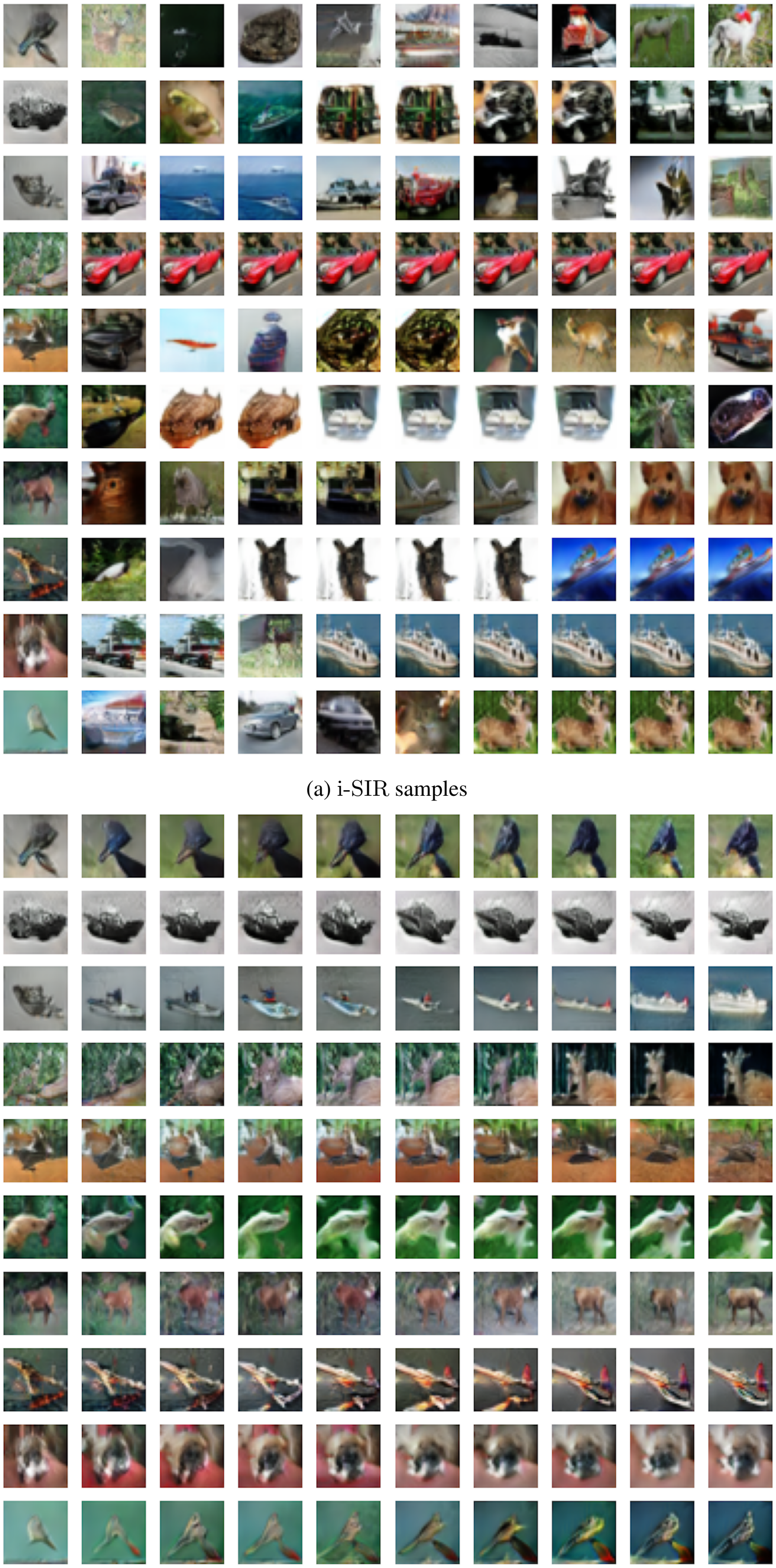


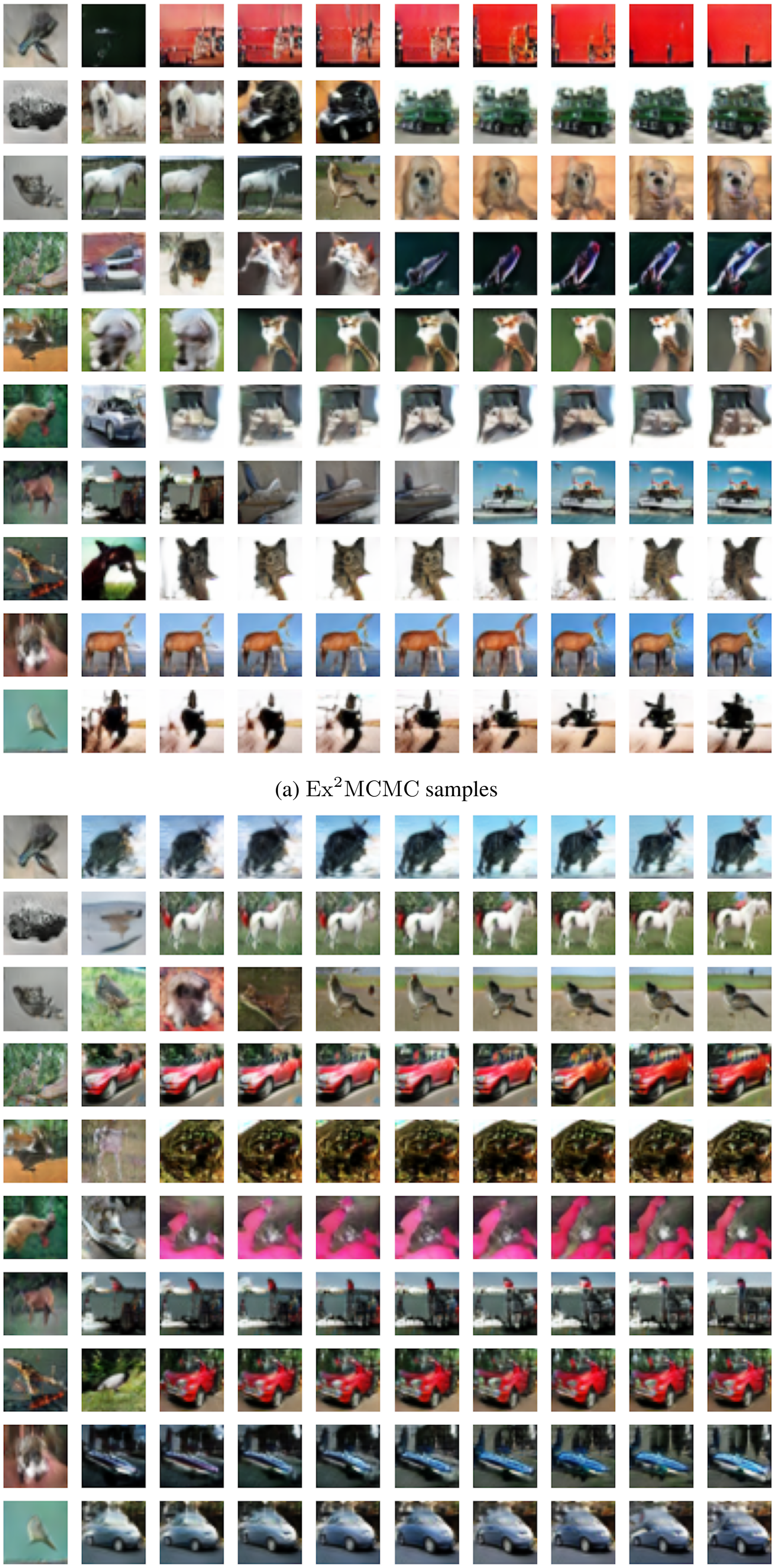
Figure 14: Energy profile for random axis pairs, SN-GAN



(a) i-SIR samples

(b) MALA samples

Figure 15: i-SIR and MALA samples, SN-GAN.



(a) Ex^{2}MCMC samples

(b) FlEx^{2}MCMC samples

Figure 16: Ex^{2}MCMC and FlEx^{2}MCMC samples, SN-GAN.

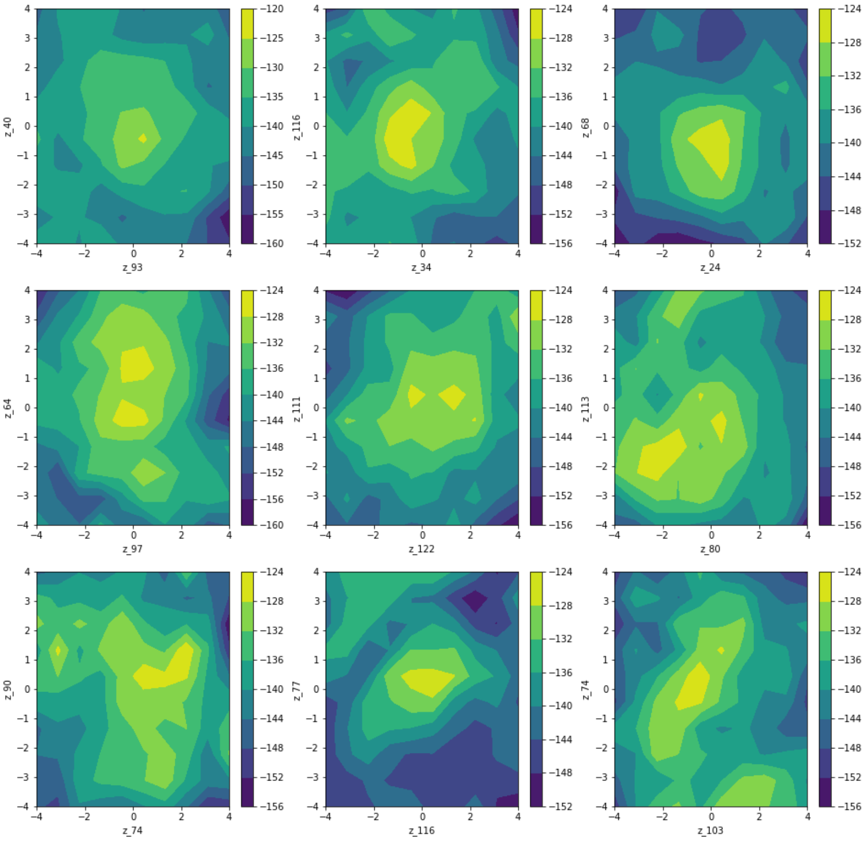
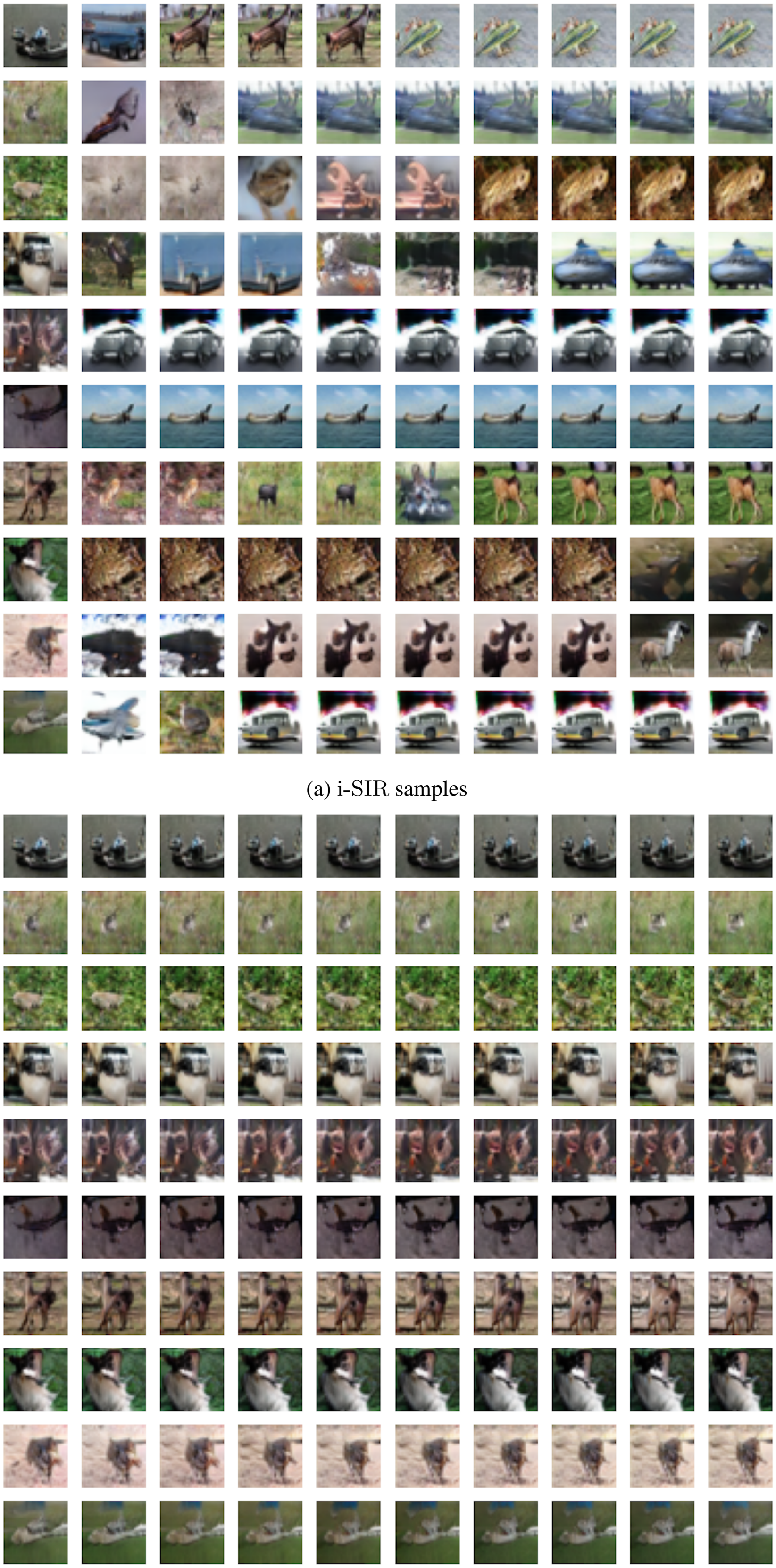


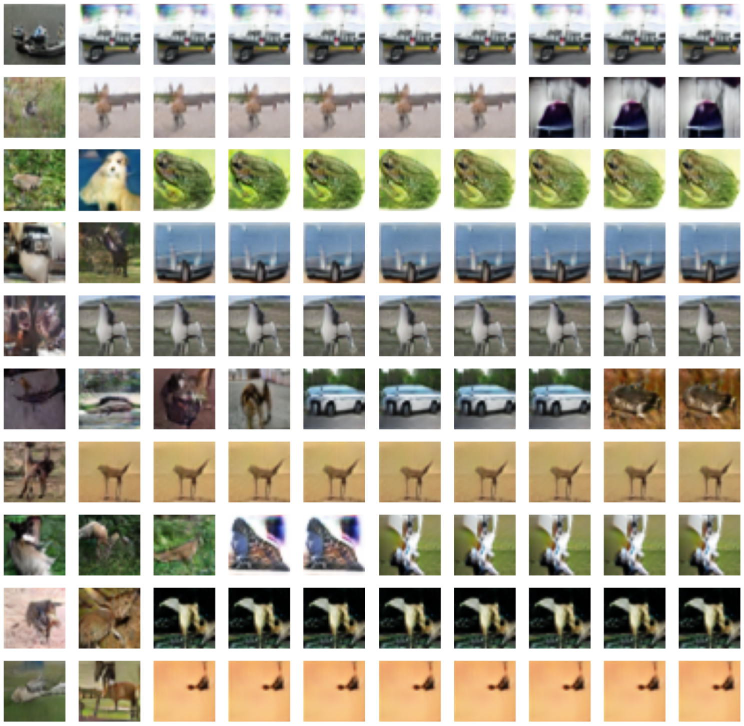
Figure 17: Energy profile for random axis pairs, DC-GAN



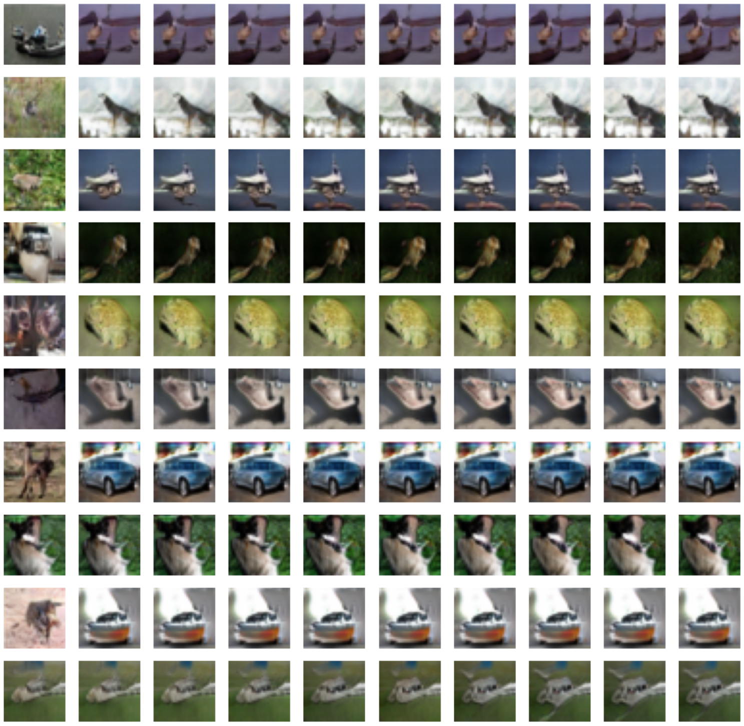
(a) i-SIR samples

(b) MALA samples

Figure 18: i-SIR and MALA samples, DC-GAN.



(a) Ex^{2}MCMC samples



(b) FlEx^{2}MCMC samples

Figure 19: Ex^{2}MCMC and FlEx^{2}MCMC samples, DC-GAN.