Local-Global MCMC kernels: the best of both worlds

Sergey Samsonov[1] **Evgeny Lagutin**[1] **Alain Durmus**[3]
Marylou Gabrié[2] **Alexey Naumov**[1] **Eric Moulines**[1,2]
[1]HSE University [2]Ecole Polytechnique [3]ENS Paris-Saclay

HDI Lab,
HSE University

NATIONAL RESEARCH UNIVERSITY

October 2022

Importance Sampling procedure

- Suppose that we are willing to estimate $\pi(f) = \int_{\mathbb{R}^D} f(x)\pi(\mathrm{d}x)$ for some distribution π;
- π is known up to a normalizing factor Z_Π, $\pi(\mathrm{d}x) = \tilde{\pi}(\mathrm{d}x)/Z_\Pi$;
- Importance Sampling (IS) consists of re-weighting samples from a proposal distribution λ.
- Assume that $\tilde{\pi}$ and λ have densities $\tilde{\pi}$ and λ, respectively.
- Define *importance weights* as $\tilde{w}(x) = \tilde{\pi}(x)/\lambda(x)$;
- The *self-normalized importance sampling* (SNIS) estimator of $\pi(f)$ is then given by

$$\Pi_N f(X^{1:N}) = \sum_{i=1}^{N} \omega_N^i f(X^i)\,,$$

where

$$X^{1:N} \sim \lambda\,, \omega_N^i = \frac{\tilde{w}(X^i)}{\sum_{j=1}^N \tilde{w}(X^j)}\,, i \in \{1, \ldots, N\}\,.$$

Self-normalized IS estimate

SNIS procedure

$$\Pi_N f(X^{1:N}) = \sum_{i=1}^{N} \omega_N^i f(X^i)\,,$$

where

$$X^{1:N} \sim \lambda\,, \omega_N^i = \frac{\tilde{w}(X^i)}{\sum_{j=1}^{N} \tilde{w}(X^j)}\,, i \in \{1, \ldots, N\}\,.$$

Pros and cons

- Advantage: Does not require have an access to the normalising constant of π, that is, $\lambda(\tilde{w}) = \int_{\mathsf{X}} \tilde{w}(x)\lambda(x)\,dx$ might be unknown;
- Disadvantage: The SNIS estimator is known to be biased

Bias of the SNIS estimate

The result below is due to [Agapiou et al., 2017, Theorem 2.1].

Theorem 1.

Assume that $\lambda(\tilde{w}^2) < \infty$, and set $\kappa[\pi, \lambda] = \lambda(\tilde{w}^2)/\lambda^2(\tilde{w})$. Then the bias and mean-squared error (MSE) of the SNIS estimator over bounded test functions f satisfying $|f|_\infty \leq 1$ are given respectively by

$$
\begin{aligned}
|\,\mathbb{E}[\Pi_N f(X^{1:N})] - \pi(f)| &\leq \frac{12\kappa[\pi, \lambda]}{N}, \\
\mathbb{E}[\{\Pi_N f(X^{1:N}) - \pi(f)\}^2] &\leq \frac{4\kappa[\pi, \lambda]}{N}\,.
\end{aligned}
\tag{1}
$$

From IS to SIR

- Sampling counterpart of the IS procedure is known as Sampling Importance Resampling (SIR; Rubin [1987]);
- Sample $X^1, \ldots, X^N$ - i.i.d. from λ and compute the importance weights $\omega_N^1, \ldots, \omega_N^N$;
- Sample $Y^1, \ldots, Y^M$ from $X^1, \ldots, X^N$ with replacement, and with probabilities proportional to the weights $\omega_N^1, \ldots, \omega_N^N$. That is, we sample from the empirical distribution

$$\hat{\pi}(\mathrm{d}x) = \sum_{i=1}^{N} \omega_N^i \delta_{X^i}(\mathrm{d}x)\,,$$

  where $\delta_y(\mathrm{d}x)$ denotes the Dirac mass at y.
- As $N \to \infty$, $Y^1, \ldots, Y^M \sim \hat{\Pi}$ will be distributed according to π.
- Main drawback: the described procedure is only asymptotically valid;
- Iterating samples from λ, we arrive at iterated SIR algorithm (i-SIR, Andrieu et al. [2010], and Andrieu et al. [2018]).

Iterated SIR (i-SIR) algorithm

Algorithm 1: Single stage of i-SIR algorithm

Input : Sample Y_j from previous iteration
Output: New sample Y_{j+1}

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

i-SIR properties

- Under appropriate conditions, the distribution of Y_k approaches π, regardless of the initial distribution;
- Disadvantage: Waste of computational resources: $N-1$ out of N generated particles in the chunk $X_{j+1}^{1:N}$ are not used

V-geometric ergodicity

Definition: V-norm

Let $V(x) : \mathbb{R}^d \mapsto [1; +\infty)$, then the V-*norm* of two probability measures ξ and ξ' on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$, is defied as

$$\|\xi - \xi'\|_V := \sup_{|f(x)| \leq V(x)} |\xi(f) - \xi'(f)| .$$

If $V(x) \equiv 1$, we get the total variation distance.

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 $x \in \mathsf{X}$ and $k \in \mathbb{N}$,

$$\|\mathrm{Q}^k(x, \cdot) - \pi\|_V \leq M \{V(x) + \pi(V)\} \rho^k.$$

i-SIR algorithm

Assumption B1

Assume that $|\tilde{w}|_\infty < \infty$.

The result below is due to Andrieu et al. [2018].

i-SIR ergodicity

Assume B1. Then the Markov kernel P_N is uniformly geometrically ergodic. Namely, for any initial distribution ξ on $(\mathrm{X}, \mathcal{X})$ and $k \in \mathbb{N}$,

$$\|\xi \mathrm{P}_N^k - \pi\|_{\mathrm{TV}} \leq \kappa_N^k, \tag{2}$$

with $\epsilon_N = \frac{N-1}{2\mathrm{L}+N-2}, \mathrm{L} = |\tilde{w}|_\infty / \lambda(\tilde{w})$ and $\kappa_N = 1 - \epsilon_N$. Hence, its mixing time is upper bounded by

$$\tau_{mix,N} = \lceil -\ln 4 / \ln \kappa_N \rceil,$$

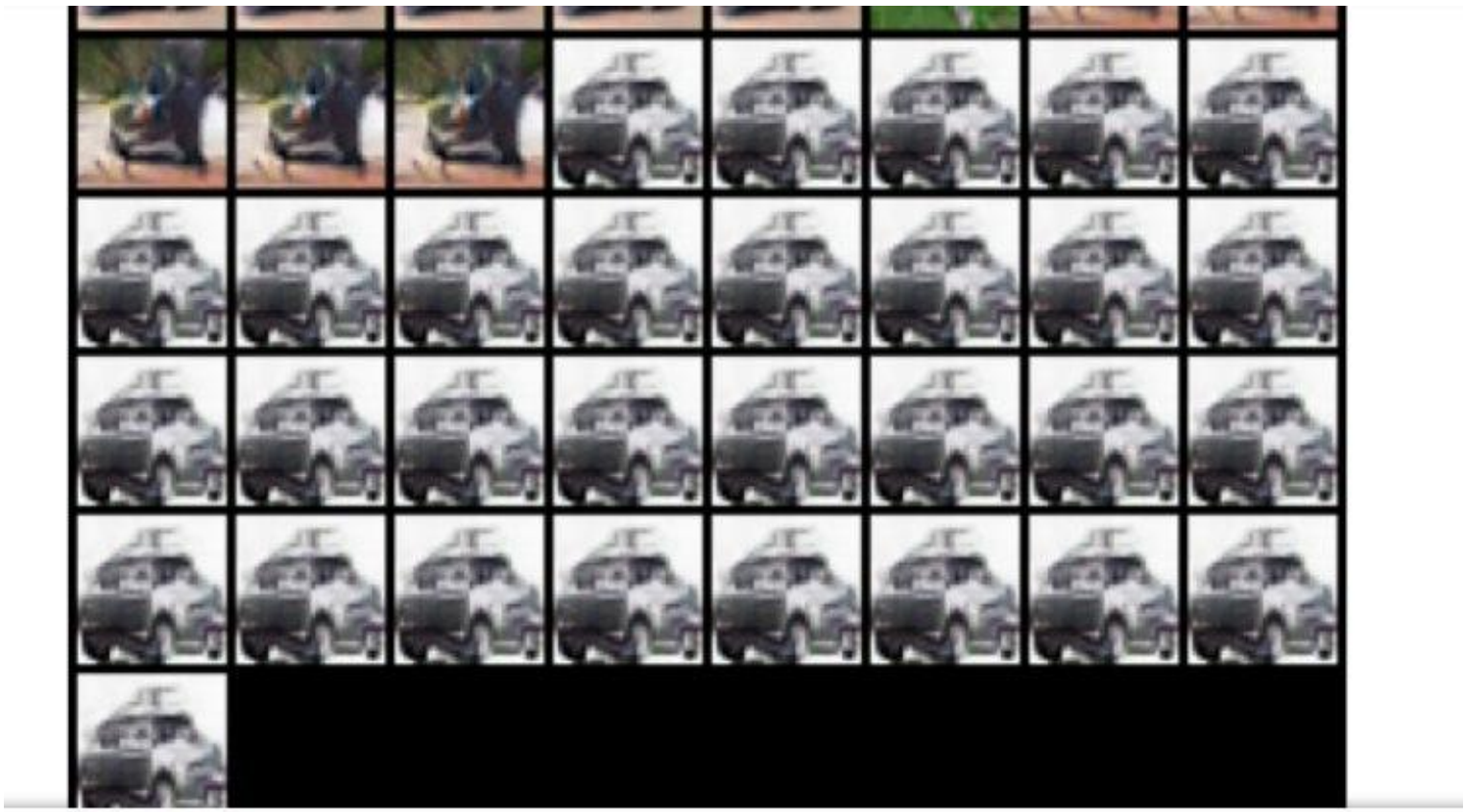
i-SIR algorithm

- Provided also that $|\tilde{w}|_\infty < \infty$, it was shown in Andrieu et al. [2018] that the Markov kernel P_N is uniformly geometrically ergodic. Namely, for any initial distribution ξ on $(\mathrm{X}, \mathcal{X})$ and $k \in \mathbb{N}$,

$$\|\xi \mathrm{P}_N^k - \pi\|_{\mathrm{TV}} \leq \kappa_N^k, \tag{3}$$

  with $\epsilon_N = \frac{N-1}{2\mathrm{L}+N-2}, \mathrm{L} = |\tilde{w}|_\infty / \lambda(\tilde{w})$ and $\kappa_N = 1 - \epsilon_N$.
- Note that the bound (3) relies significantly on the restrictive condition that weights are uniformly bounded $|\tilde{w}|_\infty < \infty$.
- Moreover, even when this condition is satisfied, the rate κ_N can be close to 1 when the dimension d is large.
- 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.

i-SIR sampling from an energy-based model on CIFAR-10



(a) One trajectory of i-SIR algorithm.

Global samplers

- Examples: Neural Transport HMC (Hoffman et al. [2019]), Multiple Try Metropolis (Liu et al. [2000]), i-SIR (Andrieu et al. [2010])
- Able to generate more global updates, but difficult to design
- Issue: The acceptance rate of independent proposals decreases dramatically with dimensions

Main ideas

- We focus on combining local and global samplers
- Intuition: local steps interleaved between global updates increase accuracy by allowing accurate sampling in distribution tails;
- Global kernel: iterative-sampling importance resampling (i-SIR), Andrieu et al. [2010]. This kernel uses multiple proposals in each iteration;
- Local samplers: Metropolis Adjusted Langevin (MALA), Hamiltonian Monte Carlo (HMC).
- We call this combination strategy Explore-Exploit MCMC (Ex^{2}MCMC)

Ex^{2}MCMC algorithm

- Main i-SIR drawback: absence of local exploration moves;
- Idea: apply a local MCMC kernel R (*rejuvenation kernel*) after each i-SIR step;
- R has π as invariant distribution;
- Here comes Ex^{2}MCMC : Exploration steps through i-SIR , Exploitation steps through R$(x, \cdot)$;
- As our default choice we consider MALA as rejuvenation, but other ones (HMC, NUTS) are also possible.

Ex^{2}MCMC algorithm

Algorithm 1: Single stage of Ex^{2}MCMC algorithm with independent proposals

Procedure Ex^{2}MCMC $(Y_j, \Lambda, \mathrm{R})$**:**

  Input : Previous sample Y_j;
  proposal distribution Λ;
  rejuvenation kernel R;

  Output: New sample Y_{j+1};

  Set $X_{j+1}^1 = Y_j$, draw $X_{j+1}^{2:N} \sim \lambda$;

  for $i \in [N]$ **do**

    compute the normalized weights $\omega_{i,j+1} = \tilde{w}(X_{j+1}^i) / \sum_{k=1}^N \tilde{w}(X_{j+1}^k)$;

  Set $I_{j+1} = \mathrm{Cat}(\omega_{1,j+1}, \ldots, \omega_{N,j+1})$;

  Draw $Y_{j+1} \sim \mathrm{R}(X_{j+1}^{I_{j+1}}, \cdot)$.

Assumptions

A1

(i) R has π as its unique invariant distribution;
(ii) There exists a function $V : \mathsf{X} \to [1, \infty)$, 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
$\mathsf{R}V(x) \leq \lambda_{\mathsf{R},r} V(x) + \mathsf{b}_{\mathsf{R},r} \mathbb{1}_{\mathsf{V}_r}$, where $\mathsf{V}_r = \{x : V(x) \leq r\}$;

A2

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

Ex^{2}MCMC 's V-geometric ergodicity

Theorem

Let A1 and A2 hold. Then, for all $x \in \mathsf{X}$ and $k \in \mathbb{N}$,

$$\|\mathsf{K}_N^k(x, \cdot) - \pi\|_V \leq c_{\mathsf{K}_N}\{\pi(V) + V(x)\}\tilde{\kappa}_{\mathsf{K}_N}^k \,, \tag{4}$$

where c_{K_N}, $\tilde{\kappa}_{\mathsf{K}_N} \in [0, 1)$ are some constants. In addition, $c_{\mathsf{K}_N} = c_{\mathsf{K}_\infty} + O(N^{-1})$ and $\tilde{\kappa}_{\mathsf{K}_N} = \tilde{\kappa}_{\mathsf{K}_\infty} + O(N^{-1})$.

Ex^{2}MCMC 's V-geometric ergodicity

Theorem

Let A1 and A2 hold. Then, for all $x \in \mathsf{X}$ and $k \in \mathbb{N}$,

$$\|\mathsf{K}_N^k(x, \cdot) - \pi\|_V \leq c_{\mathsf{K}_N}\{\pi(V) + V(x)\}\tilde{\kappa}_{\mathsf{K}_N}^k \,, \tag{5}$$

where c_{K_N}, $\tilde{\kappa}_{\mathsf{K}_N} \in [0, 1)$ are some constants. In addition, $c_{\mathsf{K}_N} = c_{\mathsf{K}_\infty} + O(N^{-1})$ and $\tilde{\kappa}_{\mathsf{K}_N} = \tilde{\kappa}_{\mathsf{K}_\infty} + O(N^{-1})$.

Toy example: Gaussian mixture

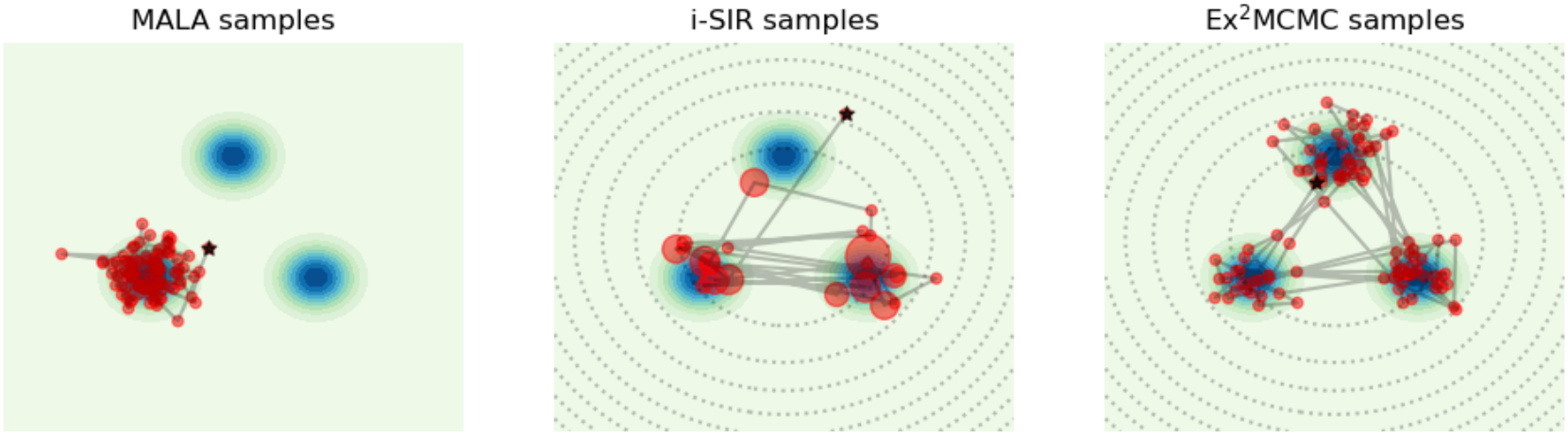


Figure: Single chain mixing visualization, 3 gaussians mixture, $d = 2$. The target density is given by

$$p_\beta(x) \propto \sum_{i=1}^{3} \beta_i \exp\left\{-\|x - \mu_i\|^2/(2\sigma^2)\right\}, \tag{7}$$

where we set all $\beta_i = 1/3$.

Toy example: Gaussian mixture (continue)

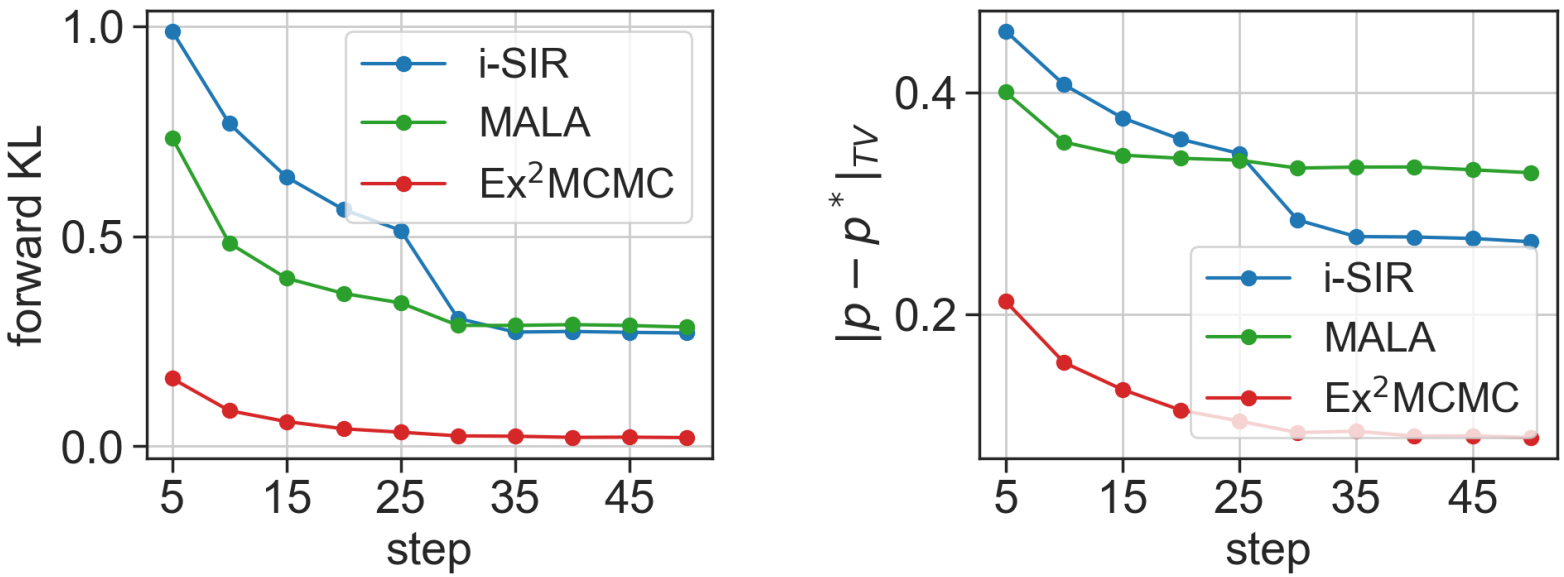


Figure: Set mixing weights to $\beta = (\beta_1, \beta_2, \beta_3) = (2/3, 1/6, 1/6)$. Quantitative analysis of parallel chains, $M = 500$ chains KDE

Toy example: Gaussian mixture (continue)

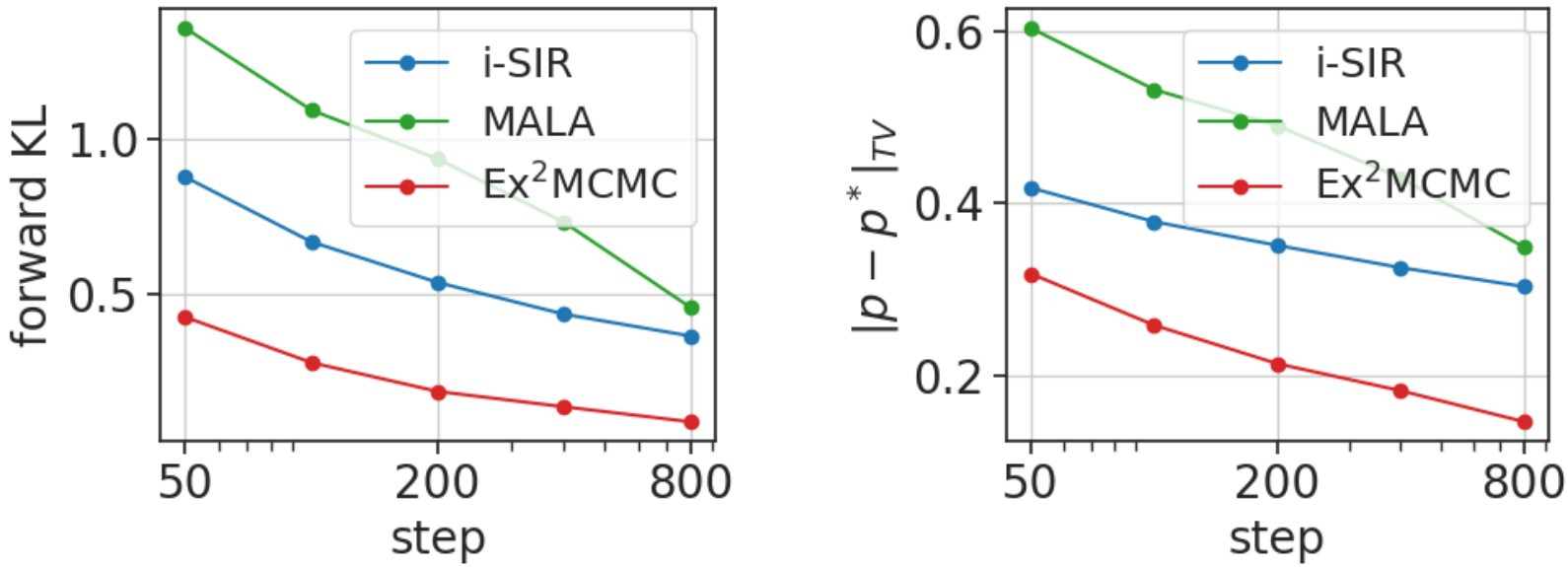


Figure: Set mixing weights to $\beta = (\beta_1, \beta_2, \beta_3) = (2/3, 1/6, 1/6)$. Quantitative analysis during for single chains statistics, $M = 100$ trajectories average

Adaptive modifications of Ex^{2}MCMC

- Consider family of proposals $\{\lambda_\theta\}, \theta \in \mathbb{R}^D$, chosen to match the target distribution $\tilde{\pi}$;
- Let $T\colon \mathbb{R}^d \to \mathbb{R}^d$ be smooth and invertible. Denote by $T\#\Lambda$ the distribution of $Y = T(X)$ with $X \sim \lambda$;
- The corresponding density is given by $\lambda_T(y) = \lambda\big(T^{-1}(y)\big)\, \mathrm{J}_{T^{-1}}(y)$, where J_T denotes the Jacobian determinant of T;

Adaptive proposals: learning procedure

- Disperancy measure: linear combination of forward and backward KL divergence (generalizations to [Papamakarios et al., 2021] possible);
- Forward and backward KL:

$$\mathcal{L}^f(\theta) = \int \log \frac{\pi(x)}{\lambda_\theta(x)} \pi(x) \mathrm{d}x \,,$$

$$\mathcal{L}^b(\theta) = \int \log \frac{\lambda(x)}{\pi\big(T_\theta(x)\big)\, \mathrm{J}_{T_\theta}(x)} \lambda(x) \mathrm{d}x \,.$$

- Given a sample $Y_k \sim \pi\,, Z_k \sim \lambda$, $k \in \{1, \ldots, K\}$, the gradients $\nabla\mathcal{L}^f$ and $\nabla\mathcal{L}^b$ can be estimated as

$$\widehat{\nabla\mathcal{L}^f}(Y^{1:K}, \theta) = -\frac{1}{K} \sum_{k=1}^{K} \nabla \log \lambda_\theta(Y_k)\,,$$

$$\widehat{\nabla\mathcal{L}^b}(Z^{1:K}, \theta) = -\frac{1}{K} \sum_{k=1}^{K} \nabla \log\big(\tilde{\pi}(T_\theta(Z_k)\, \mathrm{J}_{T_\theta}(Z_k)\big).$$

- Following Gabrié et al. [2021], we consider

$$\widehat{\mathcal{L}}(Y^{1:K}, Z^{1:K}, \theta) = \alpha\widehat{\mathcal{L}^f}(Y^{1:K}, \theta) + \beta\widehat{\mathcal{L}^b}(Z^{1:K}, \theta)\,.$$

FIEx^{2}MCMC algorithm with adaptive proposals

Algorithm 2: Single stage of FIEx^{2}MCMC. Steps of Ex^{2}MCMC are done in parallel with common values of proposal parameters θ_j. Step 4 updates the parameters using the gradient estimate obtained from all the chains.

Input : weights θ_j, batch $Y_j^{1:K}$

Output: new weights θ_{j+1}, batch $Y_{j+1}^{1:K}$

1 **for** $k \in [K]$ **do**

2 $\quad Y_{j+1,k} = \text{Ex}^2\text{MCMC}\,(Y_{j,k}, T_{\theta_j}\#\Lambda, \mathrm{R})$

3 Draw $\bar{Z}^{1:K} \sim \lambda$.

4 Update $\theta_{j+1} = \theta_j - \gamma\widehat{\nabla\mathcal{L}}(Y_{j+1}^{1:K}, \bar{Z}^{1:K}, \theta_j)$.

Practical note

In our experiments: T_θ is modelled as a normalizing flow based on RealNVP architecture (Dinh et al. [2017]).

Example: Complex geometry distributions

- Funnel distribution: for $x \in \mathbb{R}^d$, consider the density

$$p_f(x) = Z^{-1} \exp\left(-x_1^2/2a^2 - (1/2)\mathrm{e}^{-2bx_1} \sum_{i=2}^{d} \{x_i^2 + 2bx_1\} \right),$$

  Here we fix the hyperparameters $a = 2$, $b = 0.5$;

- Symmetric banana-shaped distribution: for $x \in \mathbb{R}^d$, $d = 2k$, consider the density

$$p_b(x) = Z^{-1} \exp\left(-\sum_{i=1}^{d/2} \{x_{2i}^2/2a^2 - (x_{2i-1} - bx_{2i}^2 + a^2 b)^2/2\} \right),$$

  and set the parameters $a = 5$, $b = 0.02$.

Experiments: quality metrics

Suppose that we produce samples $\{Y_t\}_{t=1}^M$, $Y_t \in \mathbb{R}^d$.

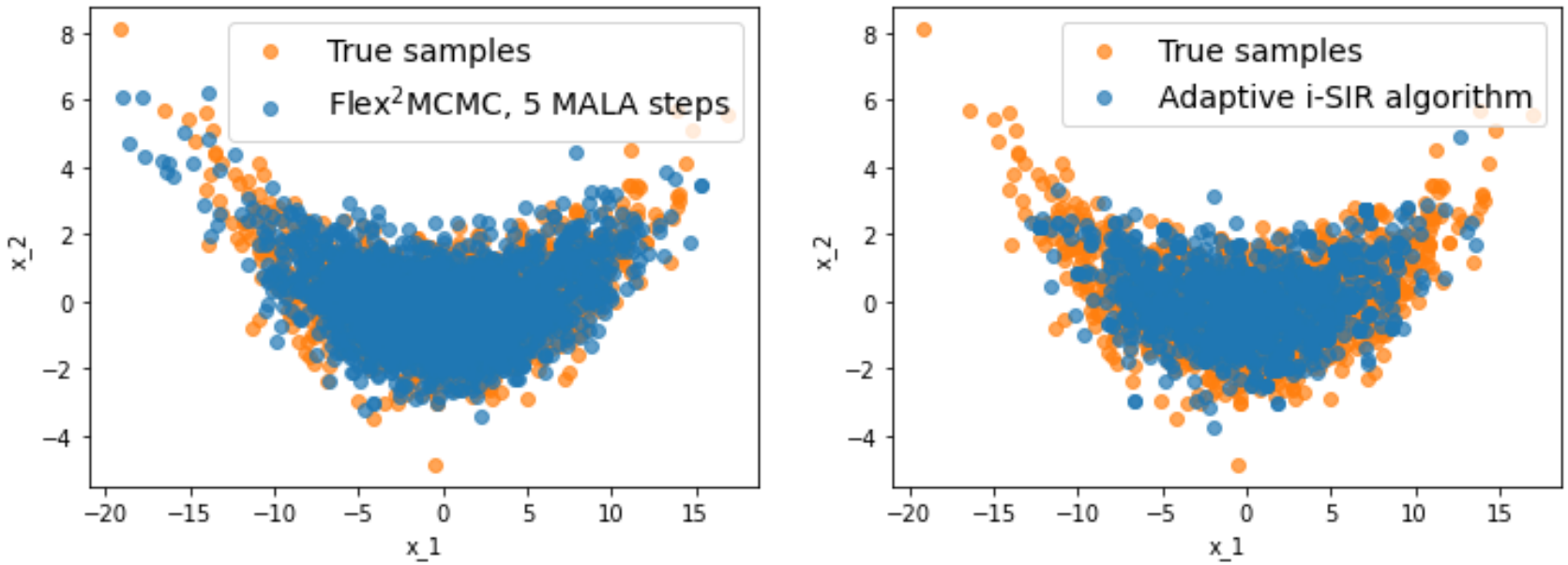
- **ESTV**: Empirical sliced total variation distance. To compute ESTV, we perform random one-dimensional projections and then perform KDE for reference and produced samples, and take TV-distance between 1-dimensional marginals;
- **ESS**: Efficient Sample Size. We define this metric as

$$\mathrm{ESS}_i = \frac{1}{1 + \sum_{k=1}^{M} \rho_k^{(i)}}, \quad \rho_k^{(j)} = \frac{\mathrm{Cov}(Y_t^{(j)}, Y_{t+k}^{(j)})}{\mathrm{Var}(Y_t^{(j)})},$$

where $\rho_k^{(j)}$ are substituted with their empirical counterparts. We report the averaged metrics

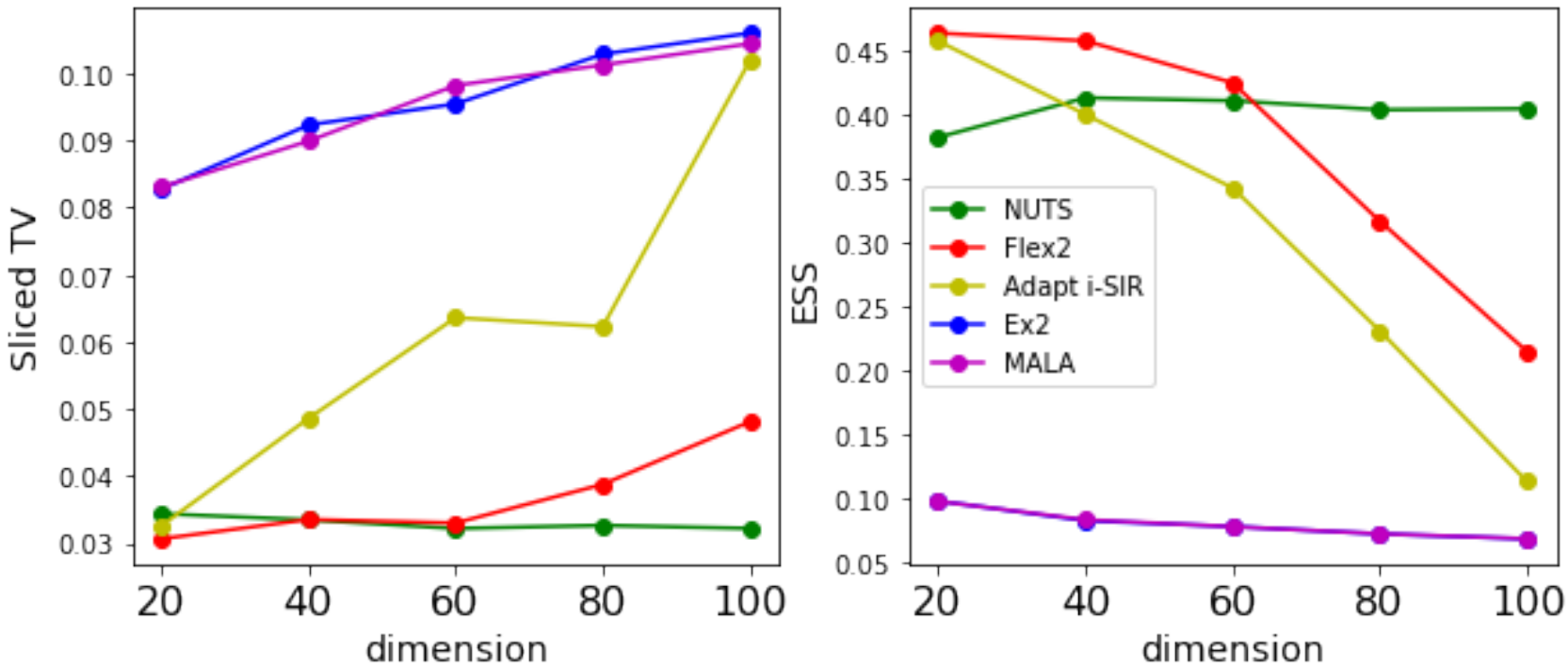
$$\mathrm{ESS} = d^{-1} \sum_{i=1}^{d} \mathrm{ESS}_i .$$

Example: Banana-shape density



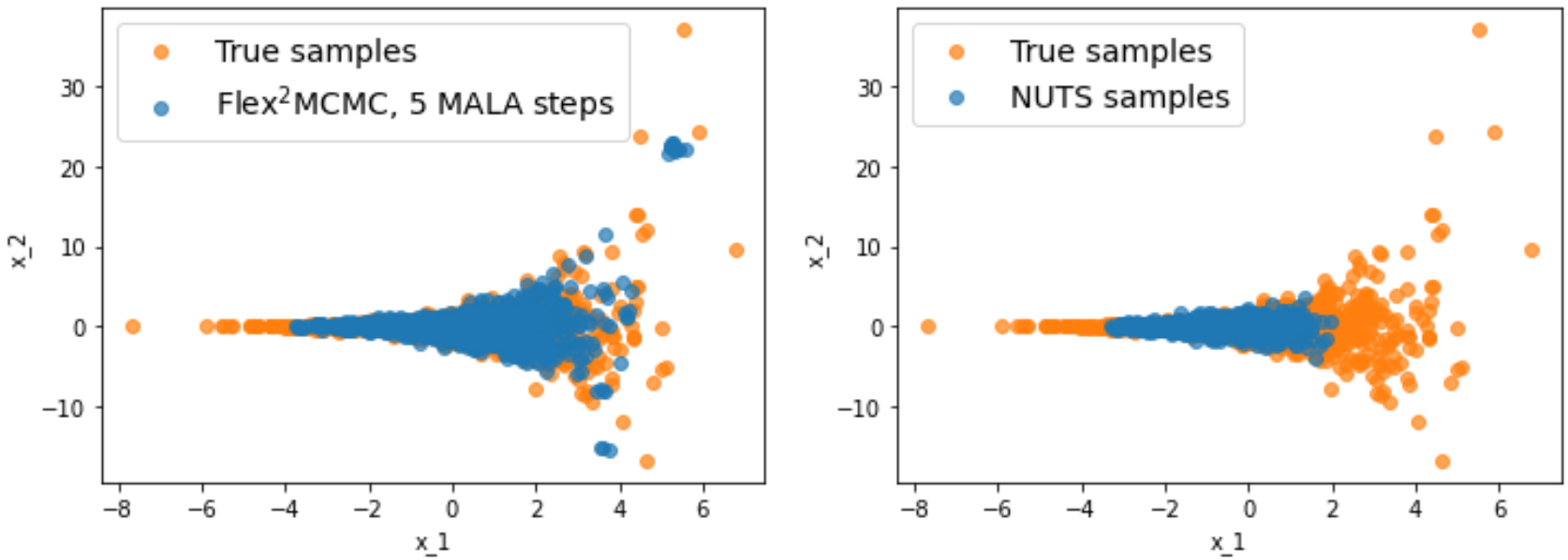
(a) $d = 100$, 2000 samples projection

Example: Banana-shape density



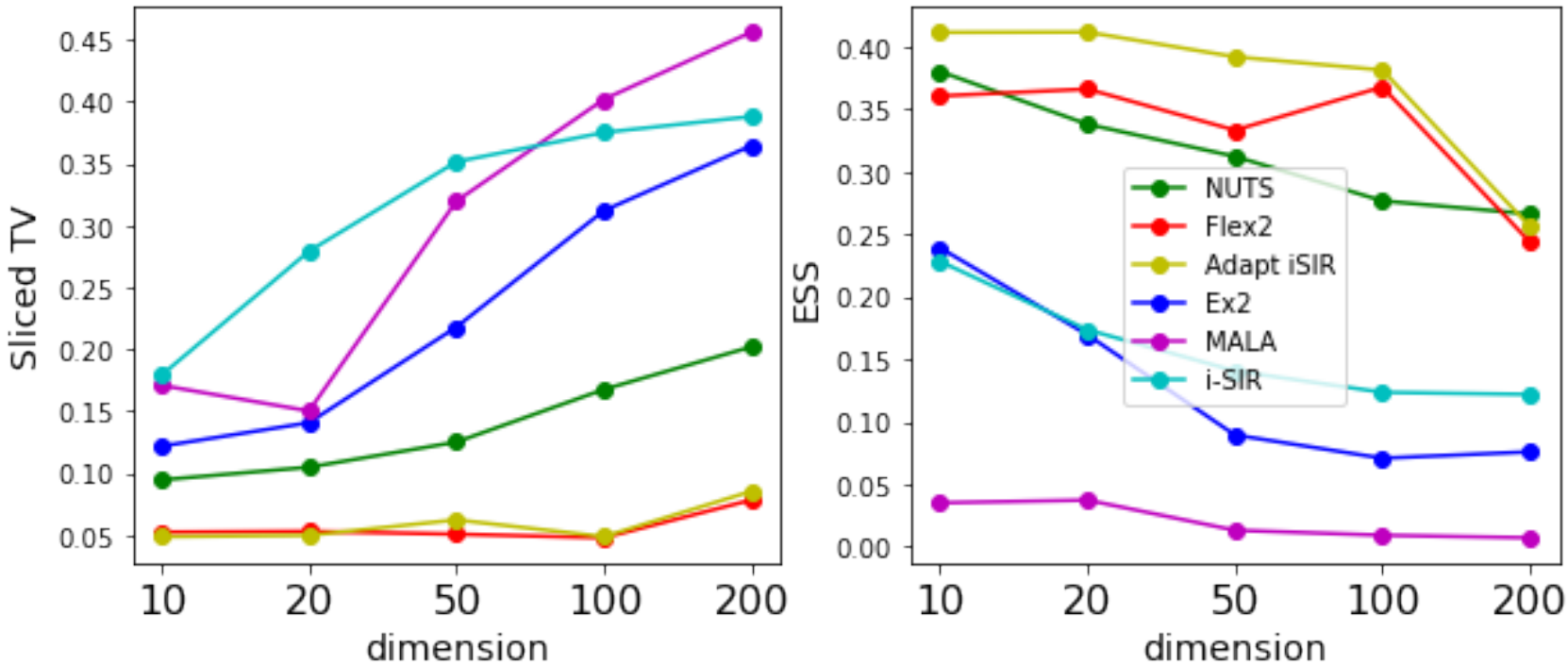
(a) Banana-shape distribution, metrics

Example: Funnel



(a) Funnel distribution, $d = 100$, 1000 samples projection

Example: Funnel



(a) Funnel distribution, metrics

GANs as Energy-based models

- Generator $G : \mathbb{R}^d \mapsto \mathbb{R}^D$: takes a latent variable z from a prior density $p_0(z)$, $z \in \mathbb{R}^d$, produces $G(z) \in \mathbb{R}^D$ in the observation space;
- Discriminator $D : \mathbb{R}^D \mapsto [0, 1]$: takes a sample in the observation space, distinguishes between real examples and fake ones;
- GAN training objective:

$$\begin{aligned} L_D &= -\mathbb{E}_{x \sim p_{\text{data}}} [\log D(x)] - \mathbb{E}_{z \sim p_z} [\log \big(1 - D(G(z))\big)] \\ L_G &= \mathbb{E}_{z \sim p_z} [\log \big(1 - D(G(z))\big)] \end{aligned} \tag{8}$$

- Consider $p_d(x)$ and $p_g(x)$ be the densities of real and fake observations, respectively;
- Optimal discriminator:

$$D^\star(x) = \frac{p_d(x)}{p_d(x) + p_g(x)} \tag{9}$$

GANs as an energy-based model

- Main drawback: information accumulated by discriminator is not used during the generation procedure;
- Let $d^\star(x) = \operatorname{logit} D^\star(x)$, therefore:

$$\frac{p_d(x)}{p_d(x) + p_g(x)} = \frac{1}{1 + \frac{p_g(x)}{p_d(x)}} = \frac{1}{1 + \exp(-d^\star(x))}$$

  Hence, we can express

$$p_d(x) = p_g(x)\mathrm{e}^{d^\star(x)}\,.$$

- Let us introduce $d(x) = \operatorname{logit} D(x)$ and consider the corresponding energy-based model

$$p_d^\star(x) = p_g(x)\mathrm{e}^{d(x)}/Z_0\,,$$

  where Z_0 is the normalizing constant. If $D(x) \approx D^\star(x)$, $p_d^\star(x)$ is close to $p_d(x)$;
- Sample from $p_d^\star(x)$ using MCMC.

GANs as an energy-based model

- Similar idea considered in Turner et al. [2019]; main issue: MCMC in pixel space is highly inefficient;
- Che et al. [2020] suggested latent-space sampling from the model

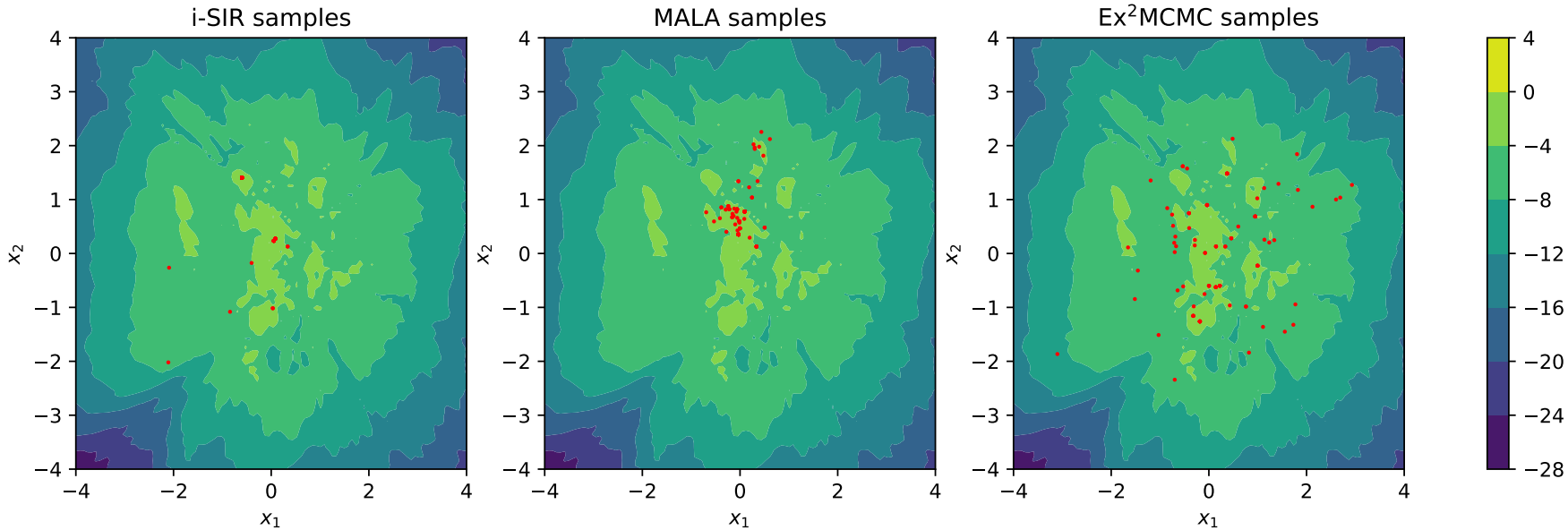
$$p_d^\star(z) = p_0(z) \exp\left\{\text{logit}\big(D(G(z))\big)\right\}, z \in \mathbb{R}^d,$$

  where $p_0(z)$ is the generator's prior distribution in the latent space;
- Note that the Wasserstein GAN also allows for an energy-based representation, with the corresponding latent distribution being equal to

$$p_W^\star(z) = p_0(z) \exp\left\{D(G(z))\right\}, z \in \mathbb{R}^d,$$

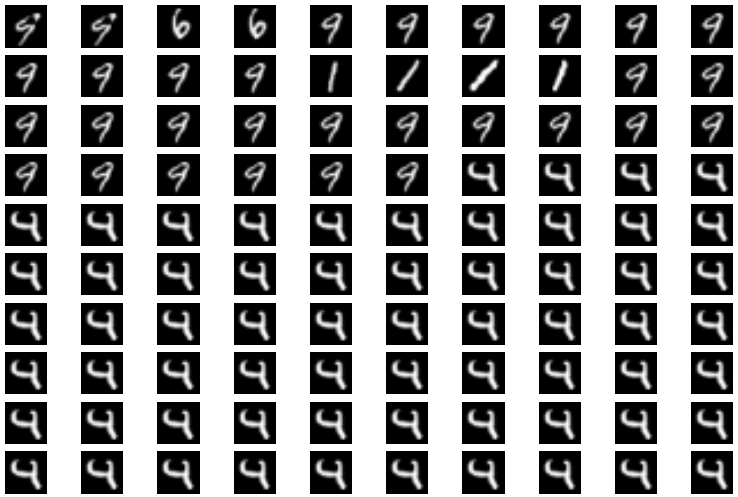
- Sampling using Langevin-based algorithms, as suggested in Che et al. [2020], can be inefficient, especially if d is large.

Results: sampling MNIST with latent dimension $d = 2$

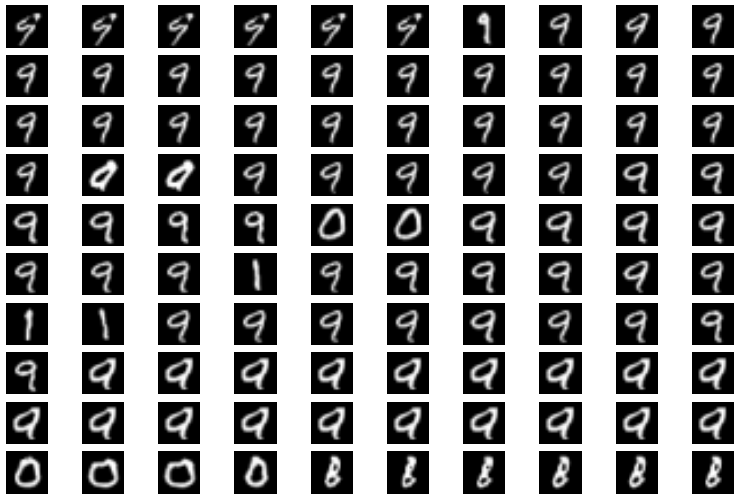


(a) JS-GAN: latent space visualizations

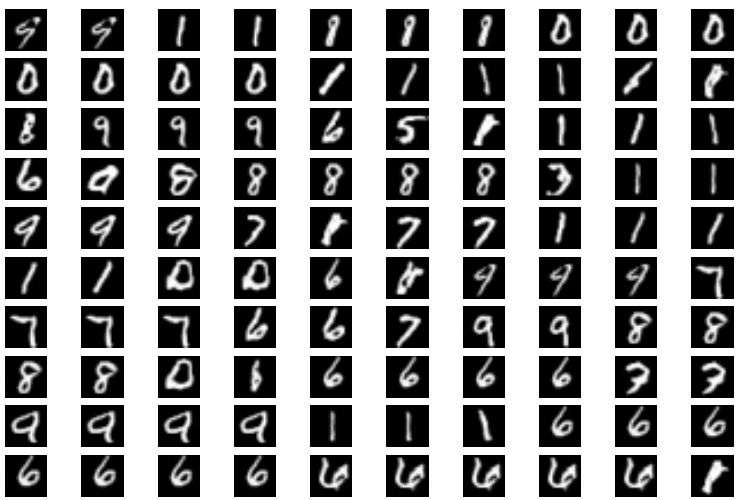
Results: MNIST visualized



(a) i-SIR samples

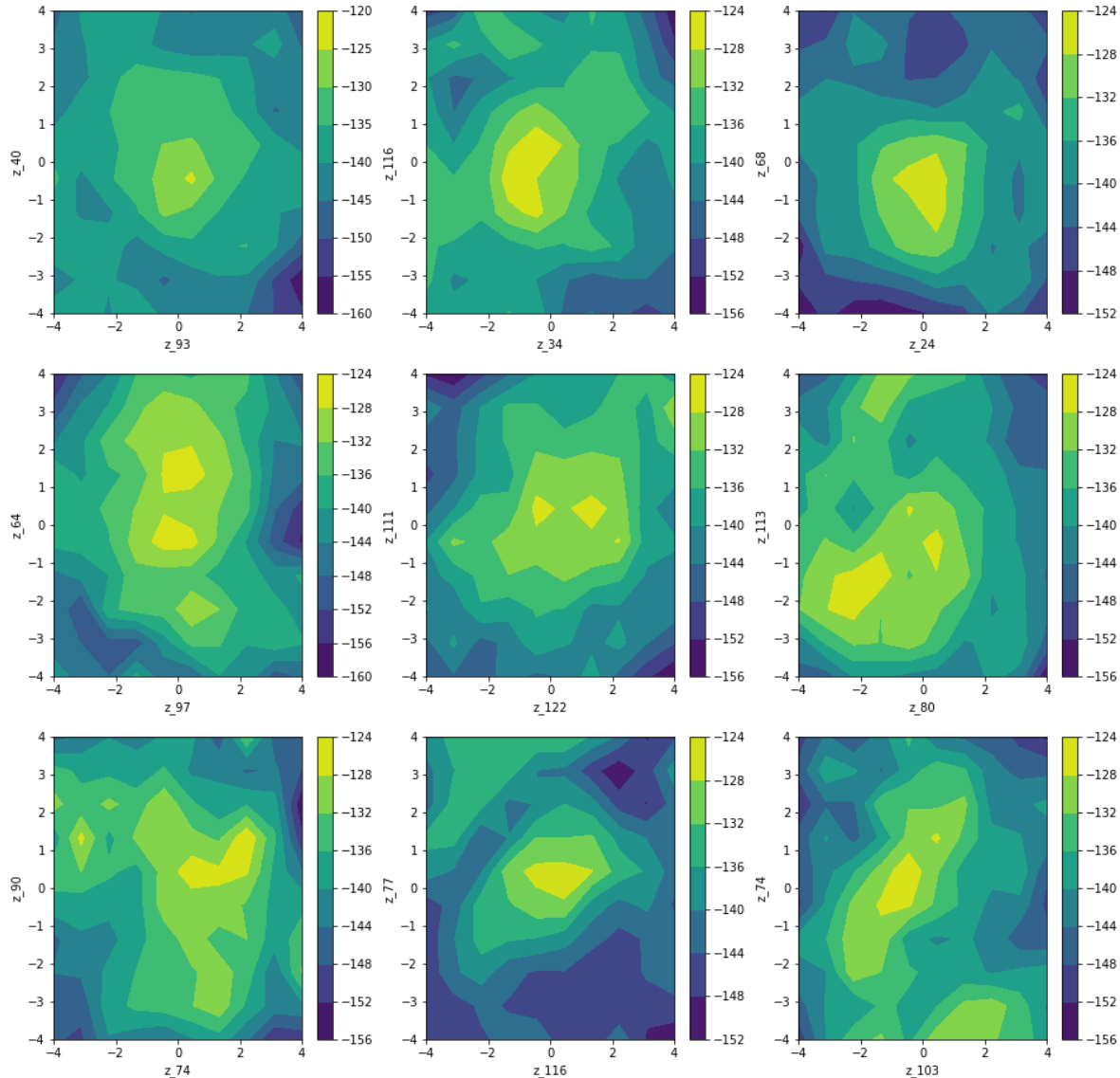


(b) MALA samples

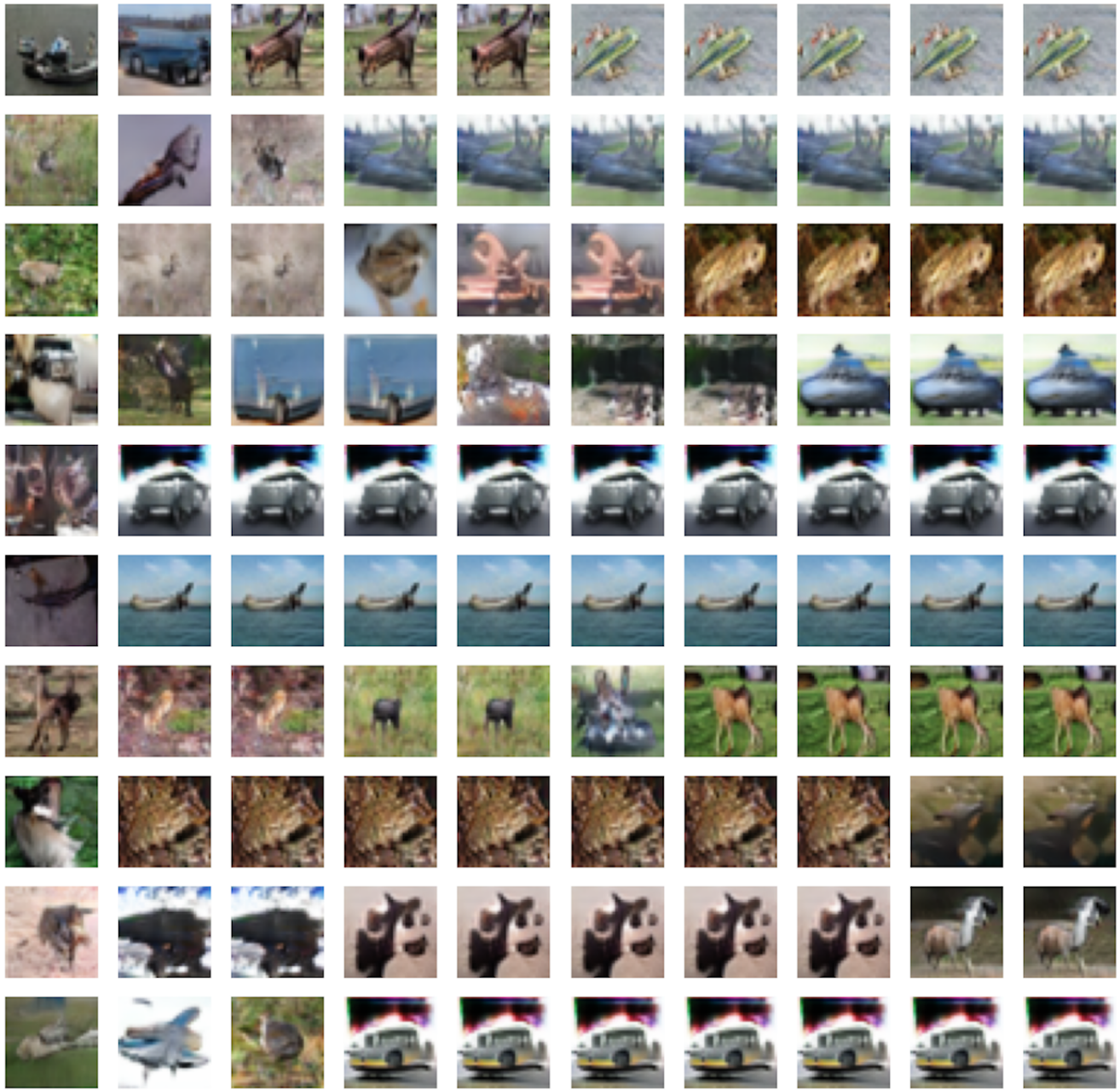


(c) Ex^{2}MCMC samples

DC-GAN energy profile, latent space



i-SIR on CIFAR-10



MALA on CIFAR-10

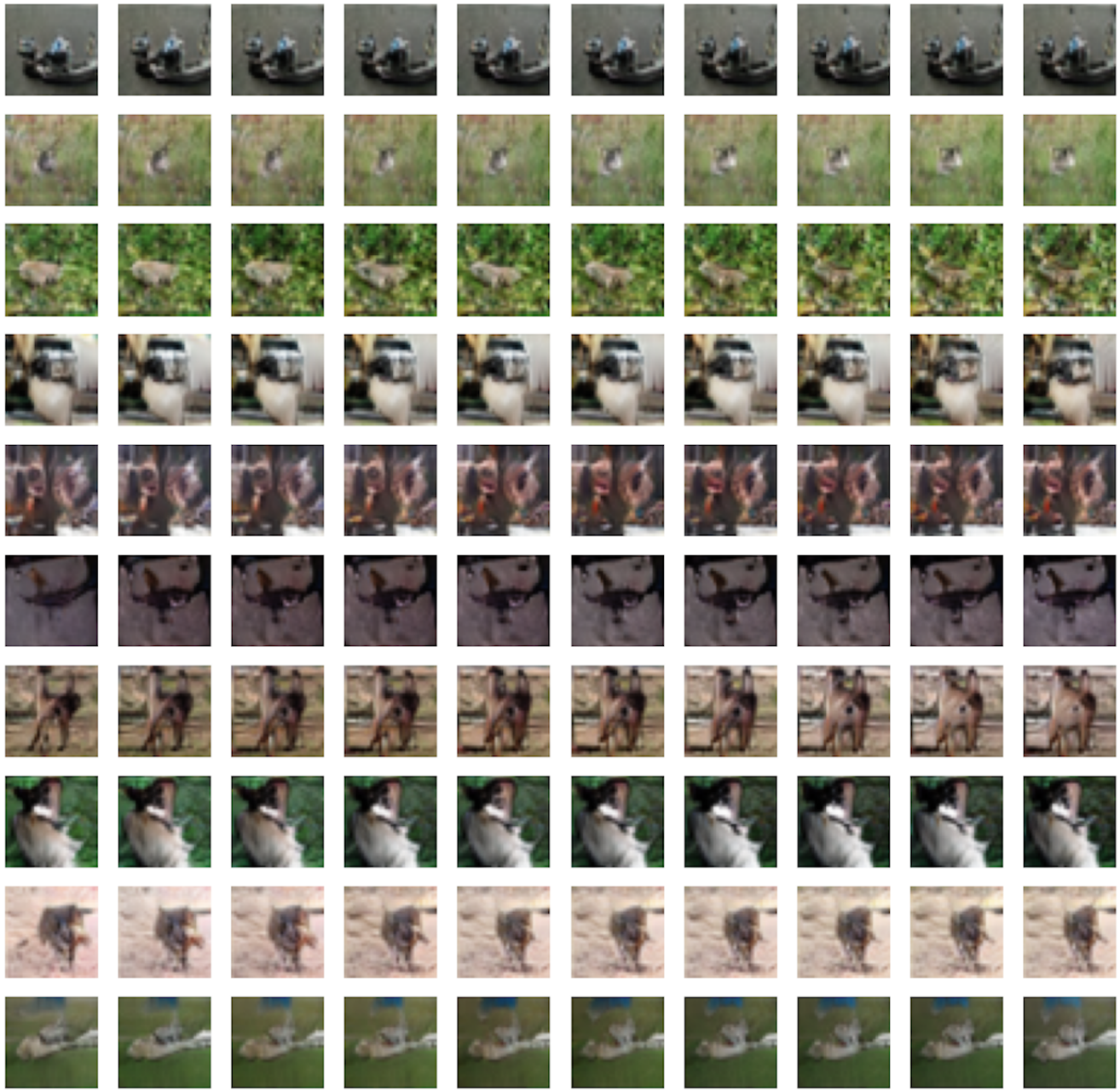
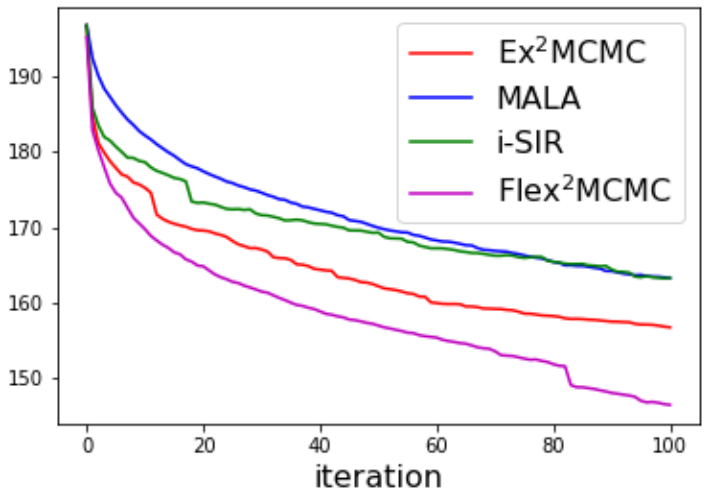


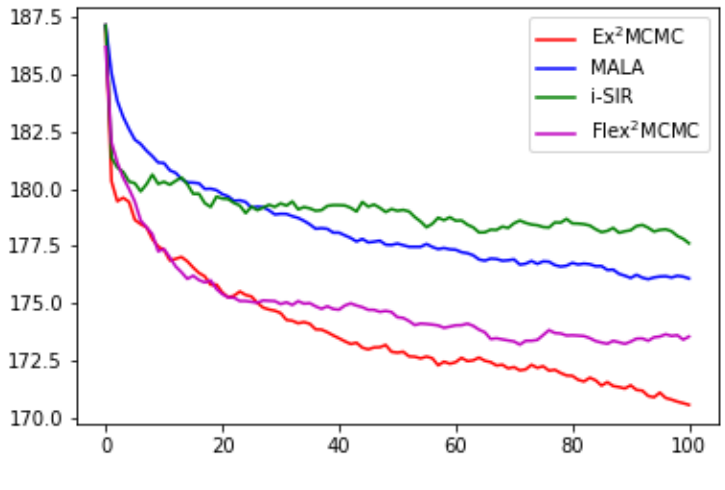
Figure: Ex^{2}MCMC samples, DC-GAN.

Figure: FIEx^{2}MCMC samples, DC-GAN.

Results: energy landscapes on CIFAR-10



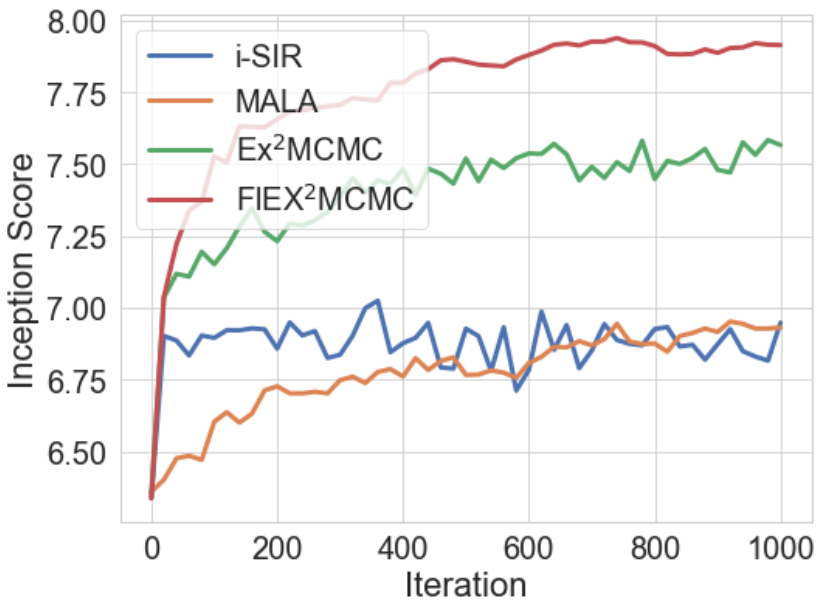
(a) DC-GAN



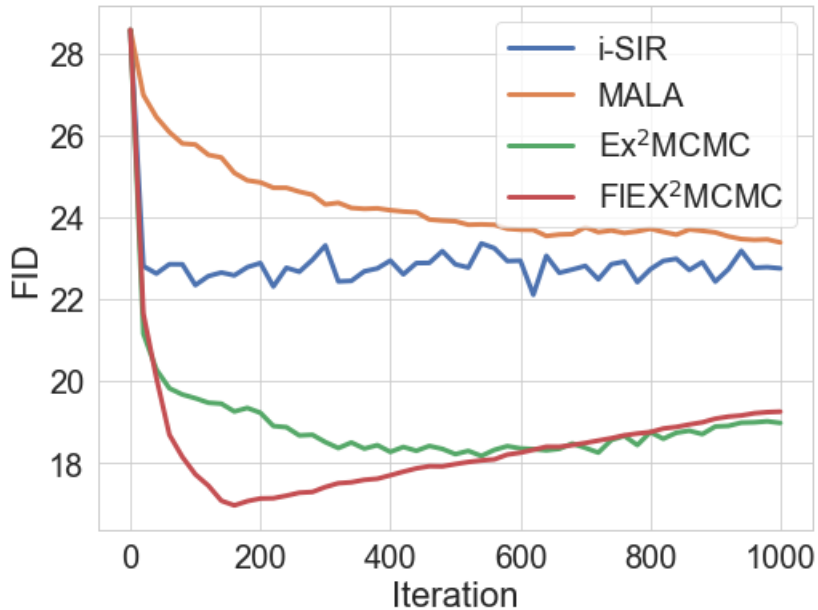
(b) SN-GAN

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

Results: FID and IS dynamics on CIFAR-10 sampling



(a) DC-GAN



(b) SN-GAN

Figure: IS and FID scores for DC-GAN on CIFAR-10 dataset.

Another ways to improve SNIS

Are there ways to further improve i-SIR ?

Indeed, one can try to recycle *all* the generated samples by incorporating all the proposed candidates $X_k^{1:N}$ into the estimator.

BR-SNIS properties

Under A1, define the constants

$$\begin{aligned} \varsigma^{bias} &= 4(\kappa[\pi,\lambda]+1+\mathrm{L}) \\ \varsigma_i^{mse} &= 4(\kappa[\pi,\lambda]\mathbb{1}_{\{0,1\}}(i)+(1+\mathrm{L})^2\mathbb{1}_{\{1,2\}}(i)), \quad i\in\{0,1,2\}. \end{aligned} \tag{10}$$

Then the following theorem holds:

Theorem 2.

Assume A1. Then for every initial distribution ξ, bounded measurable function f on $(\mathrm{X},\mathcal{X})$ such that $|f|_\infty \leq 1$, $N \geq 2$, and $k,\ell \in \mathbb{N}$,

$$\begin{aligned} \left|\mathbb{E}_\xi[\Pi_N f(X_k^{1:N})]-\pi(f)\right| &\leq \varsigma^{bias}(N-1)^{-1}\kappa_N^{k-1}, \\ \mathbb{E}_\xi[\{\Pi_N f(X_k^{1:N})-\pi(f)\}^2] &\leq \sum_{i=0}^{2}\varsigma_i^{mse}(N-1)^{-1-i/2}, \end{aligned} \tag{11}$$

Notes

- The bias decreases inversely with the number of candidates and exponentially with the number of iterations;
- The MSE is also inversely proportional to the number of candidates N.

BR-SNIS: the algorithm

- Consider an estimator formed by an average across the IS estimators $(\Pi_N f(X_k^{1:N}))_{k\in\mathbb{N}}$;
- To mitigate the bias, remove a "burn-in" period whose length k_0 should be chosen proportional to the mixing time of the Markov chain $\{Y_k,\, k \in \mathbb{N}\}$
- This yields the Rao-Blackwellised estimator for $\pi(f)$:

$$\Pi_{(k_0,k),N}(f) = (k - k_0)^{-1} \sum_{\ell=k_0+1}^{k} \Pi_N f(X_\ell^{1:N})$$

- All the importance weights included in the estimators are obtained as a by-product of the i-SIR schedule, so we do not add any computational overhead.

BR-SNIS: bias and variance

The total number of samples (generated by the proposal λ) underlying the BR-SNIS estimator is $M = (N-1)k$. Denote $\upsilon = (k - k_0)/k$ the fraction of the number of candidate pools used in the estimator, and $\mathrm{MSE}^{is}_M = (4/M)\kappa[\pi, \lambda]$.

BR-SNIS

Assume A1. Then for every initial distribution ξ, bounded measurable function f on $(\mathsf{X}, \mathcal{X})$ such that $|f|_\infty \leq 1$, and $N \geq 2$,

$$\begin{aligned} &\left|\mathbb{E}_\xi[\Pi_{(k_0,k),N}(f)] - \pi(f)\right| \leq \zeta^{bias}(\upsilon M)^{-1} 4^{-k_0/\tau_{mix,N}} \\ &\mathbb{E}_\xi[\{\Pi_{(k_0,k),N}(f) - \pi(f)\}^2] \leq \mathrm{MSE}^{is}_{\upsilon M} + \zeta^{mse}(\upsilon M)^{-1}(N-1)^{-1/2}. \end{aligned} \tag{12}$$

Moreover, for every $\delta \in (0, 1)$,

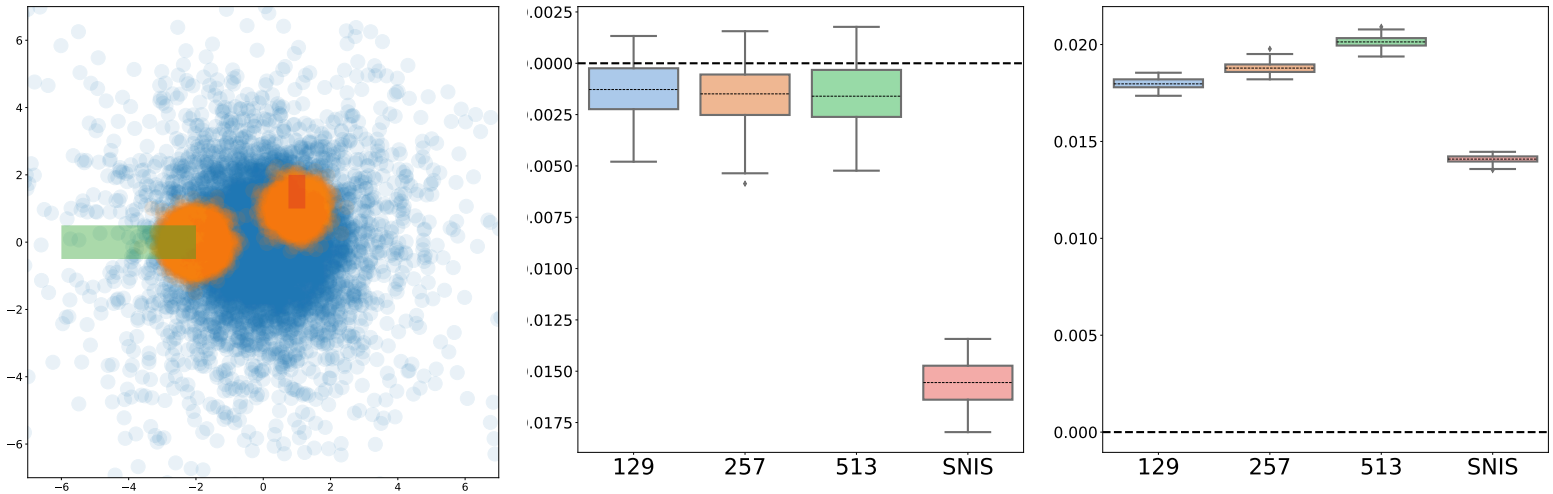
$$|\Pi_{(k_0,k),N}(f) - \pi(f)| \leq \varsigma^{hpd}(\upsilon M)^{-1/2}(\log(4/\delta))^{1/2} \tag{13}$$

with probability at least $1 - \delta$, where ς^{hpd}, ζ^{mse}, and ζ^{bias} are some computable constants.

Summary

- The bias of the BR-SNIS estimator decreases exponentially with the burn-in period k_0;
- Large k_0 comes at a price of increased overall MSE, mainly through the term $\mathrm{MSE}^{is}_{\upsilon M}$;
- A natural way to reduce the variance: use bootstrap;
- Apply a random permutation to the samples, re-compute BR-SNIS on the basis of the bootstrapped samples, then average over the bootstrapped BR-SNIS replicates. This allows for the choice $k_0 = k - 1$.

Examples: Gaussian mixture



(a) 2d projection (b) Bias (c) MSE

Figure: Comparison between SNIS and BR-SNIS for the same budget. In each boxplot the dotted line represents the **mean** value of the samples.

Target π: mixture of Gaussians in $d = 7$, proposal - Student distribution with $\nu = 3$ degrees of freedom, $f(x) = \mathbb{1}_A(x) - \mathbb{1}_B(x)$.

Results: IWAE

- Let $x \in \mathbb{R}^P$, $z \in \mathbb{R}^d$, define the joint density function $p_\theta(x, z)$. We aim to find θ maximizing

$$p_\theta(x) = \int p_\theta(x, z) \mathrm{d}z \,.$$

- Then,

$$\nabla_\theta \log p_\theta(x) = \int \nabla_\theta \log p_\theta(x, z) p_\theta(z \mid x) \mathrm{d}z, \tag{14}$$

- The conditional density $p_\theta(z \mid x) = p_\theta(x, z)/p_\theta(x)$ is intractable and can only be sampled;
- The VAE (Kingma and Welling [2014]): introduce ϕ and a family of variational distributions $q_\phi(z \mid x)$;
- Maximize ELBO:

$$\mathcal{L}(\theta, \phi) = \log p_\theta(x) - \mathrm{KL}\big(q_\phi(\cdot \mid x) \parallel p_\theta(\cdot \mid x)\big) \leq \log p_\theta(x);$$

Results: IWAE

- Consider the *importance weighted autoencoder* (IWAE). The objective of the IWAE:

$$\mathcal{L}_M(\theta, \phi) = \int \log \left(M^{-1} \sum_{i=1}^{M} \tilde{w}_{\theta,\phi,x}(z_i) \right) \prod_{\ell=1}^{M} q_\phi(z_\ell \mid x) \mathrm{d}z_i,$$

  where $\tilde{w}_{\theta,\phi,x}(z) = p_\theta(x, z)/q_\phi(z \mid x)$;

- Thus,

$$\nabla_\theta \mathcal{L}_M(\theta, \phi) = \int \sum_{i=1}^{M} \omega_{\theta,\phi,x}^{(i)} \nabla_\theta \log \tilde{w}_{\theta,\phi,x}(z_i) \prod_{\ell=1}^{N} q_\phi(z_\ell \mid x) \mathrm{d}z_\ell,$$

  where $\omega_{\theta,\phi,x}^{(i)} = \tilde{w}_{\theta,\phi,x}(z_i) / \sum_{j=1}^{M} \tilde{w}_{\theta,\phi,x}(z_j)$ are normalized importance weights;

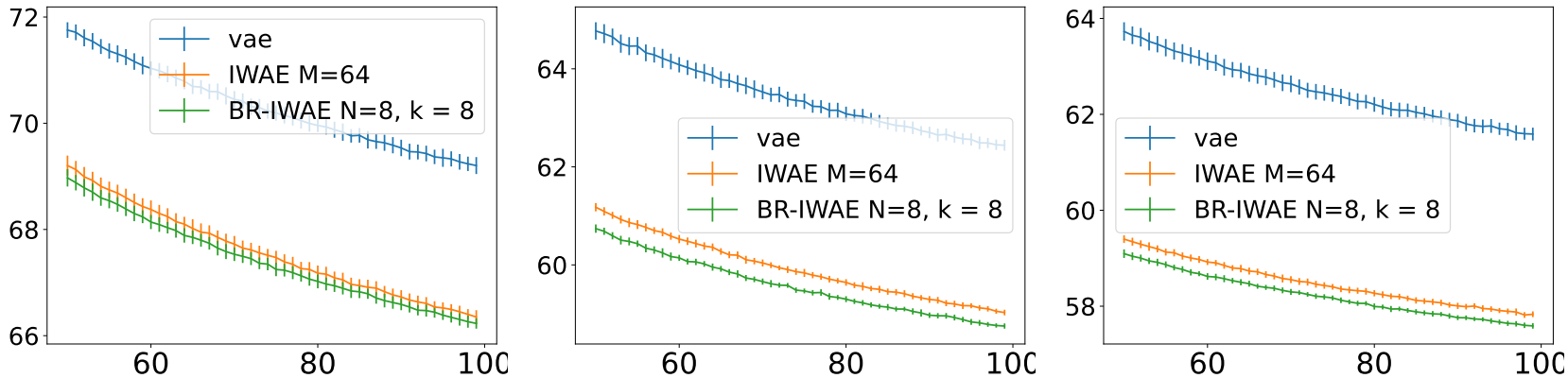
- The expression above corresponds to SNIS approximation. Thus, the optimization problem will suffer from bias.
- Proposal: use BR-SNIS for learning IWAE instead;

Results: IWAE

Latent dimension (d)	VAE	IWAE	BR-IWAE ($\boldsymbol{k = 8}$)
10	-87.40 ± 0.14	-86.44 ± 0.10	$\mathbf{-86.29 \pm 0.09}$
20	-83.55 ± 0.10	-81.81 ± 0.06	$\mathbf{-81.66 \pm 0.12}$
40	-82.90 ± 0.07	-81.05 ± 0.09	$\mathbf{-81.01 \pm 0.05}$

Table: Comparison of the mean log likelihood over the MNIST validation set (Higher is better).

Results: IWAE



(a) Dimension 10 (b) Dimension 20 (c) Dimension 40

Figure: Per epoch training loss (ELBO) for the last 40 epochs. Confidence intervals are calculated as $1.96\sigma/\sqrt{n}$ over 10 ($n = 10$) different seeds.

Papers available at:

- https://arxiv.org/abs/2207.06364 -BR-SNIS paper;
- https://arxiv.org/abs/2111.02702 - Ex^{2}MCMC paper;

Both to appear at NeurIPS-2022.

References

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. ISSN 08834237, 21688745. URL http://www.jstor.org/stable/26408299.

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

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

Tong Che, Ruixiang ZHANG, Jascha Sohl-Dickstein, Hugo Larochelle, Liam Paull, Yuan Cao, and Yoshua 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. URL https://proceedings.neurips.cc/paper/2020/file/90525e70b7842930586545c6f1c9310c-Paper.pdf.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy 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. URL https://openreview.net/forum?id=HkpbnH9lx.

Marylou Gabrié, Grant M. Rotskoff, and Eric Vanden-Eijnden. Adaptive Monte Carlo augmented with normalizing flows. *arXiv preprint arXiv:2105.12603*, 2021.

Matthew D Hoffman, Pavel Sountsov, Joshua V. Dillon, Ian Langmore, Dustin Tran, and Srinivas Vasudevan. NeuTra-lizing Bad Geometry in Hamiltonian Monte Carlo Using Neural Transport. In *1st Symposium on Advances in Approximate Bayesian Inference, 2018 1–5*, 2019. URL http://arxiv.org/abs/1903.03704.

Diederik P Kingma and Max Welling. Stochastic gradient vb and the variational auto-encoder. In *Second International Conference on Learning Representations, ICLR*, volume 19, page 121, 2014.

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

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji