PATH AUXILIARY PROPOSAL FOR MCMC IN DISCRETE SPACE

Anonymous authors
Paper under double-blind review

ABSTRACT

Energy-based Model (EBM) offers a powerful approach for modeling discrete structure, but both inference and learning of EBM are hard as it involves sampling from discrete distributions. Recent work shows Markov Chain Monte Carlo (MCMC) with the informed proposal is a powerful tool for such sampling. However, an informed proposal only allows local updates as it requires evaluating all energy changes in the neighborhood. In this work, we present a path auxiliary algorithm that uses a composition of local moves to efficiently explore large neighborhoods. We also give a fast version of our algorithm that only queries the evaluation of energy function twice for each proposal via linearization of the energy function. Empirically, we show that our path auxiliary algorithms considerably outperform other generic samplers on various discrete models for sampling, inference, and learning. Our method can also be used to train deep EBMs for high dimensional discrete data.

1 INTRODUCTION

Many real-world problems involve discrete structured data modeling, such as syntax trees for natural language processing(Tai et al., 2015), graphical model for molecules(Gilmer et al., 2017), etc. A powerful approach for modelling the distribution over structured data is Energy Based Models(LeCun et al., 2006) (EBMs). EBMs define the distribution with an unnormalized energy function, which allows great flexibility to fit the target distribution. However, this flexibility also results in the difficulties in inference and learning, as they require sampling from the EBM (Andrieu et al., 2003; Hinton, 2002), where the partition function is intractable in many cases.

Markov Chain Monte Carlo (MCMC) algorithms are one of the most widely used methodologies to sample from intractable distributions (Robert & Casella, 2013). The efficiency for MCMC depends drastically on the proposal distribution. For example, in continuous space, Metropolis-adjusted Langevin algorithm (MALA) exploits the gradient of the target in single step walk and biases the proposal distribution toward high probability region (Rossky et al., 1978; Roberts & Rosenthal, 1998; Welling & Teh, 2011); Hamiltonian Monte Carlo (HMC) employs multi-step walk and explores the distribution more efficiently(Neal, 2004; Girolami & Calderhead, 2011; Hoffman et al., 2014). These methods substantially improve the performance of the MCMC algorithm in theory and in practice. However, their proposal distributions are derived as discretization of continuous diffusion process and it is still not clear how to appropriately extend such methods into discrete space(Zanella, 2020).

An easy method in discrete space is random walk. However, in high dimensional space, random walk is typically very inefficient as it has a low accept rate. Recently, Zanella (2020) proposed a general framework called pointwise informed proposal (PIP) that shows promising results on sampling from discrete distributions. PIP utilizes the energy change in the neighborhood of current state to propose a new state. Following this work, Grathwohl et al. (2021) propose a more efficient sampler that uses Taylor series to estimate the energy change in the neighborhood. However, both methods only focus on proposing from a small neighborhood, for example, a 1-Hamming ball. This is due to the computational expense of evaluating the energy change or approximation error of Taylor series for a large neighborhood. As a result, the samples in Markov chain will have strong correlation or even be trapped at local optimum, which deteriorates the sampling efficiency.

In order to efficiently explore a larger neighborhood, we propose a path auxiliary algorithm. In each step of the path, we employ a local proposal distribution to make a sequence of small movements to reach a new state in long distance. A typical challenge for such multi-step proposal is that the accept ratio could decrease very fast when the number of steps increases. Surprisingly, we show that the accept ratio of our path auxiliary proposal is *independent* of the path and only determined by the information at the current state and the proposed state. As a result, our algorithm is able to maintain a high accept ratio with multiple local steps. We can provably show that the locally balancing functions are asymptotically optimal for our sampler. We also introduce an efficient version of the algorithm that uses linearization and can be applied to smooth target distributions.

We empirically evaluate our methods in inference, sampling, and learning on various discrete EBMs. We demonstrate that our methods significantly improve the efficiency on parity model, weighted permutation model, Ising model, Restricted Boltzmann Machine, and Factorized Hidden Markov Model. Our method can also learn competitive deep EBMs on discrete image data.

2 BACKGROUND

Energy Based Model: Let $\mathcal{X}$ be a finite state space. An EBM defines an energy function $f : \mathcal{X} \to \mathbb{R}$ with target distribution $\pi(x) = e^{-f(x)}/Z$, where $Z = \sum_{z\in\mathcal{X}} e^{-f(z)}$ (LeCun et al., 2006; Wainwright & Jordan, 2008; Du & Mordatch, 2019). The unnormalized energy function provides great flexibility to characterize complex distribution. However, this can also make the partition function Z intractable to calculate and exacerbate the difficulties of learning and inference.

Metropolis-Hastings: MCMC is a commonly used methodology to sample from an intractable distribution. Metropolis-Hastings (MH) is one of the most commonly used framework for MCMC(Metropolis et al., 1953; Hastings, 1970). Given the current state x, a proposal distribution $Q(x, \cdot)$ gives a new state y, then MH algorithm accept y with probability of $\min\{1, \frac{\pi(y)Q(y,x)}{\pi(x)Q(x,y)}\}$ to guarantee the Markov chain is π-reversible. The efficiency of MH algorithm highly depends on the selection of proposal distribution Q.

Peskun Ordering: Peskun ordering provides a method to compare the efficiency for two MCMC algorithms (Peskun, 1973; Tierney, 1998). Let P_1, P_2 be π-reversible Markov transition kernels on $\mathcal{X}$ such that $P_1(x, y) > cP_2(x, y)$ for all $x \neq y$ for a fixed $c > 0$, then we say P_1 is c-times more efficient than P_2 as the following holds: 1) $\text{var}_\pi(h, P_1) \leq \frac{1}{c}\text{var}_\pi(h, P_2) + \frac{1-c}{c}\text{var}_\pi(h), \quad \forall h : \mathcal{X} \to \mathbb{R}$; 2) $\text{Gap}(P_1) \geq c\,\text{Gap}(P_2)$. A smaller *asymptotic variance* $\text{var}_\pi(h, P)$ means a better estimation for the expectation of h and a larger *spectral gap* $\text{Gap}(P)$ means a faster convergence of the Markov chain.

Pointwise Informed Proposals and Locally Balancing Function: A pointwise informed proposal (PIP) is a MH algorithm in discrete space (Zanella, 2020). PIP uses proposal distribution $Q_g(x, y) = g(\frac{\pi(y)}{\pi(x)})I(x, y)/Z_g(x)$, where the $I(x, y) = 1_{\{y\in\mathcal{N}(x)\}}$ is the membership indicator w.r.t a symmetric neighborhood $\mathcal{N}(\cdot)$, and $Z_g(x) = \sum_{z\in\mathcal{X}} g(\frac{\pi(z)}{\pi(x)})I(x, z)$. The weight function g determines the efficiency of PIP. Zanella (2020) show that the family of locally balancing functions $\mathcal{G} = \{g : g(t) = tg(\frac{1}{t}), \forall t > 0\}$ is asymptotically optimal for PIP w.r.t. Peskun ordering. Empirically, Zanella (2020) shows $g(t) = \sqrt{t}$ and $g(t) = \frac{t}{t+1}$ have the best performance. Despite being powerful, PIP requires the calculatation of the weight $g(\frac{\pi(z)}{\pi(x)})$ for all states in the neighborhood. Hence, only a small neighborhood, usually a 1-Hamming ball, is used in existing methods.

3 PATH AUXILIARY PROPOSAL

Problems occur in point-wise informed proposal (PIP) when merely considering a small neighborhood, especially for distributions with many local optima. For example, consider a parity distribution where the state space $\mathcal{X} = \{0, 1\}^p$, and the energy function $f(x) = U$ if the number of 1s in x is odd, otherwise $f(x) = 0$. When a 1-Hamming ball neighborhood is used, by symmetry, a PIP will have a uniform probability to propose a new state from neighborhood. Then, the expected time to leave a low energy state is $O(e^U)$, which could be very inefficient when U is large. When a r-Hamming ball larger neighborhood is used, a PIP can efficiently escape a low energy state. How-

ever, the neighborhood will contain $O(n^r)$ states, which could be computationally prohibitive when n or r is large.

To address such problems with PIP, we propose a path auxiliary sampler. Instead of directly sample from a large neighborhood, path auxiliary sampler sequentially samples new state from a local proposal distribution Q_0. The computation cost at each step is still manageable as Q_0 still samples locally. Then the composition of small movements forms a path that can lead to a new state that is distant from the current state. The complexity of the sampling grows linearly instead of exponentially w.r.t. r. In this section, we first present the framework of our algorithm. Then, we discuss the choice of the weight function for path auxiliary sampler. And finally, we introduce a fast version of path auxiliary sampler.

3.1 Proposal via auxiliary path

We first define an auxiliary path. Given the neighborhood function $\mathcal{N}$, the set of auxiliary paths on $\mathcal{X}$ is defined as

$$\Sigma(\mathcal{X}, \mathcal{N}) := \{(\sigma, L) : \sigma_i \in \mathcal{X}, i = 0, ..., L; \sigma_l \in \mathcal{N}(\sigma_{l-1})), l = 1, ..., L\} \tag{1}$$

To obtain the auxiliary path, we employ a PIP $Q_0(\cdot, \cdot)$ to make local movements. We also sample path length from a prior $\alpha(\cdot)$ to assure our chain is aperiodic. The path auxiliary sampler is defined as follows

1. Sample a path length L from a prior $\alpha(L)$.
2. Denote the current state $x_t = \sigma_0$, sample $\sigma_l \sim Q_0(\sigma_{l-1}, \cdot)$ for $l = 1, ..., L$.
3. Accept $x_{t+1} = \sigma_L$ in probability $A(x, \sigma, L) = \min\left\{1, \frac{\pi(\sigma_L)\prod_{l=1}^{L} Q_0(\sigma_l, \sigma_{l-1})}{\pi(\sigma_0)\prod_{l=1}^{L} Q_0(\sigma_{l-1}, \sigma_l)}\right\}$, else $x_{t+1} = x_t$.

Theorem 1. *The path auxiliary proposal transition rule described above satisfies the detailed balance and induces a reversible Markov chain with π as its invariant distribution.*

3.2 Benefit from Auxiliary Path

We first analyze the performance of path auxiliary sampler on parity distribution mentioned above. Assume we use a uniform prior $\alpha(1) = \alpha(2) = \frac{1}{2}$. If we sample $L = 2$, then we will transit back to a new state with same energy as the current state, hence the accept ratio will be 1. As a result, we can always escape from a state in $O(1)$ steps in expectation.

Having a path auxiliary sampler to sample from L-Hamming ball at each step is more efficient than performing L steps of MH sampling in a 1-Hamming ball. To mathematically justify it, we show the accept ratio comparison between the path auxiliary sampler and 1-Hamming ball PIP sampler with the same Q_0 for proposal. Consider a path (σ, L), we compare the accept ratio below:

$$\min\left\{1, \frac{\pi(\sigma_L)\prod_{l=1}^{L} Q_0(\sigma_l, \sigma_{l-1})}{\pi(\sigma_0)\prod_{l=1}^{L} Q_0(\sigma_{l-1}, \sigma_l)}\right\} = \min\left\{1, \prod_{l=1}^{L} \frac{\pi(\sigma_l)Q_0(\sigma_l, \sigma_{l-1})}{\pi(\sigma_{l-1})Q_0(\sigma_{l-1}, \sigma_l)}\right\} \tag{2}$$

$$\geq \prod_{l=1}^{L} \min\left\{1, \frac{\pi(\sigma_l)Q_0(\sigma_l, \sigma_{l-1})}{\pi(\sigma_{l-1})Q_0(\sigma_{l-1}, \sigma_l)}\right\} \tag{3}$$

We can notice that the accept ratio for path auxiliary sampler is the product of the probability for single-step method before the minimum operator. As a result, an auxiliary path can have a high accept ratio as long as the product is large. On the contrary, in the multiple local MH case, the transition can probably be blocked at some steps in the path with low accept ratios.

3.3 Balanced Proposal

Although the product in equation 2 allows the Markov chain to escape from a local optimum when some steps in the path have high accept ratio, it can also exponentially decrease the accept ratio w.r.t. the path length L when every step has a low accept ratio. Hence, it is important to select a good local proposal Q_0. In this section, we show that the locally balancing function Zanella (2020) is asymptotically optimal for path auxiliary sampler. We first define a sub-class of weight function named as ideal function:

Definition 1. *$\mathcal{F}$ is the set of ideal function. We say a function $f \in \mathcal{F}$ if and only the following conditions holds: 1) $f : \mathbb{R}_+ \to \mathbb{R}_+$; 2) $f(1) = 1$; 3) $f(t)$ is monotonically increasing; 4) $f(t)f(\frac{1}{t})t \geq 1$*

The next theorem shows that locally balancing function $\mathcal{G}$ is asymptotically better than ideal function $\mathcal{F}$ in Peskun Ordering.

Theorem 2. *Consider the state space in Cartesian products $\mathcal{X} = \times_{i=1}^{n} \mathcal{X}_i$, where each $\mathcal{X}_i$ is a finite space with M elements, and the neighborhood is defined as 1-Hamming ball. Let d_n be the maximum degree in conditional independence graph. If 1) $\lim_{n\to\infty} \frac{d_n}{n} = 0$; 2) the target distribution satisfies $\frac{\pi(y)}{\pi(x)} \leq C < \infty, \forall y \in \mathcal{N}(x)$; 3) the path length prior α is bounded by U. Then for any $f \in \mathcal{F}$, we have a function $g \in \mathcal{G} = \{g(\cdot) : g(t) = tg(\frac{1}{t})\}$ that is asymptotically more efficient than f, which implies $\mathcal{G}$ is asymptotically optimal.*

The idea to prove this theorem is to use $\bar{f}(t) := \sqrt{f(t)f(\frac{1}{t})t}$. By definition, we have:

$$\bar{f}(t) = \sqrt{f(t)f(\frac{1}{t})t} = t\sqrt{f(t)f(\frac{1}{t})\frac{1}{t}} = t\bar{f}(\frac{1}{t}) \tag{4}$$

which means $\bar{f} \in \mathcal{G}$. Then, the theorem is proved by showing the π-reversible Markov transition kernel $P_{\bar{f}}(x, y) \geq P_f(x, y)$ asymptotically holds. The proof is given in Sec A.3

Although Definition 1 has some constraints, we can believe it already contains almost all useful weight function for PIP. Condition 1), 2), and 3) are natural for a weight function. For condition 2), we can always divide a constant $f(1)$ to satisfy it without changing the proposal distribution in PIP. For condition 3), we should be able to assume our proposal distribution has more weights on states that have larger probability in target distribution. Condition 4) is more like a technical requirement but it is easy to satisfy as long as $f(t)$ does not decrease faster than t when $t < 1$. Since the constraints for ideal function $\mathcal{F}$ are general enough for weight function, we conclude that Theorem 2 shows locally balancing function is asymptotically optimal for path auxiliary sampler.

Property 1. *When using a locally balancing function $g \in \mathcal{G}$ as the weight function, the accept ratio for path auxiliary sampler can be written as:*

$$A(x, y, \sigma, L) = \min\left\{1, \frac{Z_g(x)}{Z_g(y)}\right\} \tag{5}$$

The path independent form in Property 1 makes the accept ratio only depends on the intrinsic structure of the target distribution. Intuitively, if the energy function is smooth enough, we can imagine that the accept ratio in equation 5 will be high when y is close to x. By selecting $g(t) = \sqrt{t} \in \mathcal{G}$, we give our path auxiliary sampler (PAS) in Algorithm 1.

3.4 Scalable path auxiliary Algorithm

Although path auxiliary proposal only samples from a small neighborhood for each step in the path σ, it still requires $O(nL)$ evaluations of the probability $\pi(x)$ in total, where n is the size of the neighborhood at each step. In cases where the computational bottleneck is evaluating the target probability, this cost will be expensive. Fortunately, most distributions of interest are well-structured and many of them have a differentiable energy function, such as deep EBMs. Hence, we can use the linearization as approximation. Given the current state σ_0, the linearization is

$$\tilde{f}_{\sigma_0}(z) = f(\sigma_0) + \langle \nabla f(\sigma_0), z - \sigma_0 \rangle \tag{6}$$

we can estimate $\tilde{\pi}_{\sigma_0}(y)/\tilde{\pi}_{\sigma_0}(x) = e^{-\langle \nabla f(\sigma_0), y-x \rangle}$, and define our proposal distribution $\tilde{Q}_{g,\sigma_0}(x, y) = g(\frac{\tilde{\pi}_0(y)}{\tilde{\pi}_0(x)})/\tilde{Z}_{g,\sigma_0}(x)$, and the normalizer $Z_{g,\sigma_0}(x) = \sum_{z\in\mathcal{N}(x)} g(\frac{\tilde{\pi}_0(z)}{\tilde{\pi}_0(x)})$. to sample an auxiliary path σ. After obtaining σ_L, we use the linearization $\tilde{f}_{\sigma_L}$ at σ_L to compute the accept ratio w.r.t. the auxiliary path

$$A(x, \sigma, L) = \min\left\{1, \frac{\pi(y)\prod_{l=1}^{L} \tilde{Q}_{g,\sigma_L}(\sigma_l, \sigma_{l-1})}{\pi(x)\prod_{l=1}^{0} \tilde{Q}_{g,\sigma_0}(\sigma_{l-1}, \sigma_l)}\right\} \tag{7}$$

Algorithm 1: Path Auxiliary Sampler (PAS) and the fast version (PAFS)

Input: Target Distribution π, Initial state x_0, Path Length Prior α, Weight Function $g(t) = \sqrt{t}$
Output: Sample sequence $(x_1, x_2, ...)$

repeat
  Sample path length $L \sim \alpha(L)$
  Denote $\sigma_0 = x_t$, sample $\sigma_l \sim Q_g(\sigma_{l-1}, \cdot)$ or $\sigma_l \sim \tilde{Q}_{g,\sigma_0}(\sigma_{l-1}, \cdot)$ for $l = 1, ..., L$.
  Accept $x_{t+1} = \sigma_L$ in probability $A(x, \sigma, L) = \min\{1, \frac{Z_g(x_t)}{Z_g(\sigma_L)}\}$
    or $A(x, \sigma, L) = \min\left\{1, \frac{\pi(y)\prod_{l=1}^{L}\tilde{Q}_{g,\sigma_L}(\sigma_l,\sigma_{l-1})}{\pi(x)\prod_{l=1}^{0}\tilde{Q}_{g,\sigma_0}(\sigma_{l-1},\sigma_l)}\right\}$, else $x_{t+1} = x_t$.
until *finish sampling*;

By using approximation, we only need to evaluate $\pi(y), \pi(x), \nabla f(y), \nabla f(x)$ once in every proposal, which is a significant reduction of the computational cost. We name this algorithm as path auxiliary fast sampler (PAFS) (see Algorithm 1). The decrease of the quality from this approximation can be bounded by the following theorem.

Theorem 3. *Assume the energy function $f(x)$ is differentiable, $\nabla f(x)$ is K-Lipschitz. Consider we use weight function $g(t) = \sqrt{t}$, path length prior α bounded by U, and 1-Hamming ball as neighborhood. Denote P and $\tilde{P}$ as the transition kernel by path auxiliary sampler and path auxiliary faster sampler, respectively, we have*

$$\tilde{P}(x, y) \geq e^{-K\frac{U(U+1)}{2}} P(x, y) \tag{8}$$

Using the Peskun ordering, we know PAFS is at least $e^{-K\frac{U(U+1)}{2}}$ times as efficient as PAS. When the energy function is smooth enough, we can save $O(nU)$ evaluations of energy function without losing too much proposal quality.

4 Related Works

Informed proposal for Metropolis-Hastings (MH) algorithm has been extensively studied in the continuous space(Robert & Casella, 2013). The most famous algorithms are Metropolis-adjusted Langevein algorithm (MALA) (Roberts & Rosenthal, 1998) and Hamiltonian Monte Carlo (HMC). MALA, HMC, and their variants (Girolami & Calderhead, 2011; Hoffman et al., 2014; Welling & Teh, 2011; Titsias & Papaspiliopoulos, 2018) exploit the gradient of the target distribution to bias the proposal distribution towards high probability regions. Although gradient-based methods having brought substantial improvements in continuous space, it is still unclear how to extend them to discrete space. A number of methods try to map the discrete space to a continuous space using relaxation, apply informed methods in continuous space, and then map the new state back into discrete space(Zhang et al., 2012; Pakman & Paninski, 2013; Nishimura et al., 2017; Han & Liu, 2018). Such methods work in some scenarios, but a key challenge is the embedding of discrete space into continuous space can destroy the inherent discrete structure, resulting in highly multi-modal and irregular target distribution in continuous spaces.

Recently, several approaches use point-wise informed proposal (PIP) for discrete distributionTitsias & Yau (2017); Zanella (2020). PIP evaluates the neighborhood of the current state to bias the proposal distribution. Zanella (2020) shows that a family of locally balancing function is asymptotically optimal for PIP and Power & Goldman (2019) extends it to Markov jumping process. When the target distribution is smooth enough, Grathwohl et al. (2021) employs a Taylor's series to approximate the target distribution and further improve the sampling efficiency. Though PIP algorithms successfully show orders of magnitude improvements in efficiency compared to other informed methods on many discrete spaces when using same neighborhood, they are not able to explore a large neighborhood as they need to evaluate all states in neighborhood. An extension of GWG (Grathwohl et al., 2021) can partially address this problem via sampling multiple dimensions to modify in one step. However, such procedure can easily lead to a backtrack when modifies too many dimensions in one step and decreases its efficiency.

5 EXPERIMENTS

5.1 SAMPLERS UNDER CONSIDERATION

In this section, we empirically evaluate the sampling efficiency of path auxiliary sampler (PAS) and path auxiliary fast sampler (PAFS). We choose the path length prior α as a uniform distribution on $\{1, ..., 2X-1\}$ and denote the corresponding samplers as PAS-X and PAFS-X.

We compare our methods with five types of baselines: random walk sampler (RW), Gibbs sampler (Gibbs), Hamming ball sampler (HB), locally balanced sampler (LB), Gibbs with gradient sampler (GWG). RW is an informed proposal with $g(t) = 1$ that uniformly propose new state from neighborhood. Gibbs partitions the dimension of a state x into two groups x_u and x_{-u}, then updates the state from conditional distribution $p(x_u|x_{-u})$. We denote it as Gibbs-X, where X refers to the dimension of x_u. HB is a two-stage Gibbs sampler on the extended state space (x, x') (Titsias & Yau, 2017). We use HB-10-1, where 10 is the block size, and 1 is the hamming ball size. LB (Zanella, 2020) is implemented as a single-step PIP with $g(t) = \sqrt{t}$. GWG (Grathwohl et al., 2021) is a fast approximation of LB, which uses first order Taylor's series of the energy function to estimate the probability ratio in neighborhood, and sample the index to modify. The original paper mainly explores modifying 1 dimension per step, but GWG can be easily extend to sampling multiple indices as indicated in the original paper. We denote the extended algorithm as GWG-X, where X indicates that the number of indices to modify per step is uniformly sampled from $\{1, ..., 2X-1\}$.

5.2 INFERENCE ON ENERGY BASED MODEL

Parity Model: We first demonstrate the benefit of path auxiliary sampler versus single-step sampler in a inference task that estimates the mean μ of a parity distribution. A parity distribution has state space $\mathcal{X} = \{0, 1\}^p$ and energy function

$$f(x) = ((\textstyle\sum_{i=1}^{p} x_i) \bmod 2)U \tag{9}$$

i.e. a state has energy U if it has an odd number of 1s, otherwise 0. The neighborhood is defined as 1-Hamming ball. We run the simulation with $p = 100$ and $U = \{1, 3, 5\}$, and report the estimation

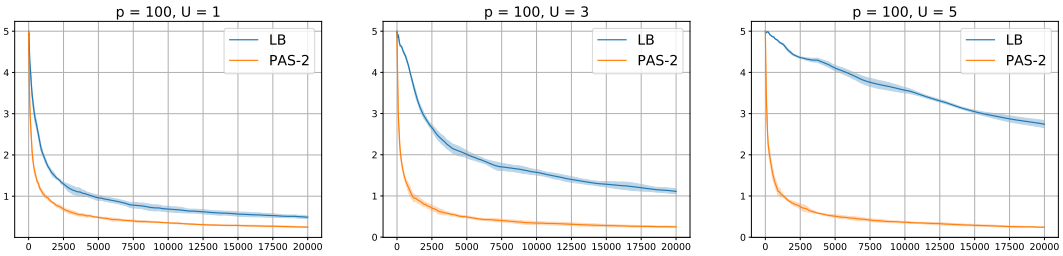

Figure 1: Estimation Error of Distribution Mean on Parity Model

error $E_n := \|\hat{\mu}_n - \mu\|_2$, where $\hat{\mu}_n = \sum_{i=1}^{n} X_i$ is the sample mean of the Markov chain at step n. For each setting and method, we run 5 chains for 20,000 steps, and we plot the mean and standard deviation of the estimation error in figure 1. We can observe that the efficiency of single step sampler decreases exponentially when U increases. On the contrary, the path auxiliary sampler can always escape from the local optima in parity distribution and estimate the distribution mean efficiently.

Weighted Permutation Model: We consider an optimization task on weighted permutation model. The state space $\mathcal{X} = S_p$ is a symmetric group, i.e. for any $\rho \in \mathcal{X}$, ρ is a permutation of $1, 2, ..., p$. The energy function is defined as:

$$f(\rho) = \textstyle\sum_{i=1}^{n} w_{i,\rho(i)} \tag{10}$$

Following Zanella (2020), the weight $\{w_{ij}\}_{i,j=1}^{p}$ are i.i.d. sampled from Gaussian$(0, \sigma^2)$. The neighborhood is determined by one exchange of the permutation. We use MCMC algorithms to find good permutations with small energy. We use $p = 100$, $\sigma = 3, 5, 10$, and run 5 simulations for each configurations. Since our formulation of weighted permutation model does not have gradient, we did not compare with GWG. From figure 2, we can see that LB sampler is trapped at local optimum, especially when σ is large and the distribution is sharp. However, PAS can always efficiently find good solutions via auxiliary path.

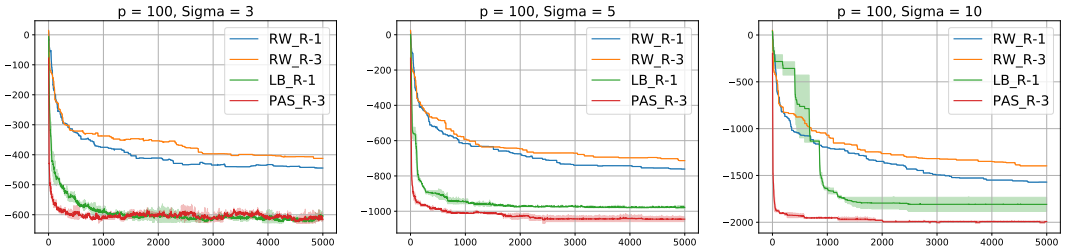

Figure 2: Energy Trace on Weighted Permutation Model

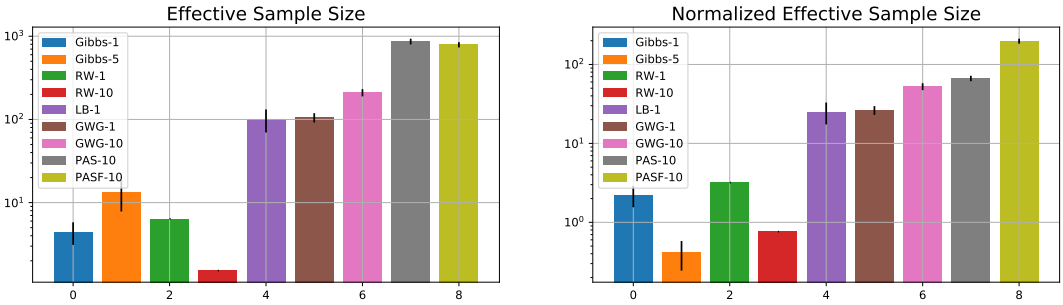

Figure 3: Sampling on 200×200 Ising model

5.3 Sampling on Energy Based Model

Lattice Ising Model: Consider the state space $\mathcal{X} = \{-1, 1\}^{V_p}$, where (V_p, E_p) is the $p \times p$ square lattice graph. For each $x \in \mathcal{X}$, the energy function is defined as:

$$f(x) = -\sum_{i \in V_p} \alpha_i x_i - \lambda \sum_{(i,j) \in E_p} x_i x_j \tag{11}$$

$\alpha_i \in \mathbb{R}$ are bias terms representing the property of x_i and λ is a global interaction term. For each sampler, we run 5 Markov chains with 1,000,000 steps. We report the effective sample size (ESS)[1] to compare the proposal quality. Since different samplers have different complexities, we also report the normalized effective sampler size (NESS), which is ESS divided by number of evaluations of energy function per step, to compare the proposal efficiency. From figure 10, we can see that PAS leads the ESS, and PAFS leads the NESS. We can also notice the ESS of PAFS is only slightly less than PAS. The reason is that the energy function of Ising model is very close to linear. Thus, PAFS only loses very limited quality from its approximation. More details are given in Appendix B.1.

Restricted Boltzmann Machine: RBM is bipartite latent-variable model, whose energy function is defined as:

$$f(x) = \log(1 + e^{w^T x + c}) + b^T x \tag{12}$$

where $\{W, b, c\}$ are parameters and $x \in \{0, 1\}^D$. We follow Grathwohl et al. (2021) to train a RBM with 500 hidden units on the MNIST dataset using contrastive divergence(Hinton, 2002). We generate samples via various MCMC samplers on the trained RBM. Besides reporting ESS, we also estimate the maximum mean discrepancy (MMD) (Gretton et al., 2012) w.r.t. a set of "ground truth" samples generated by structure known Block-Gibbs sampler. Figure 4 shows PAFS has large ESS and can efficiently match the "ground truth" samples.

Factorized Hidden Markov Model: FHMM is a statistical model that use latent variables in $\mathcal{X} = \{0, 1\}^{N \times K}$ to characterize time series data $y \in \mathbb{R}^N$. Denote $p(x)$ for hidden variables, and $p(y|x)$

[1]Computed follows Tensorflow Probability

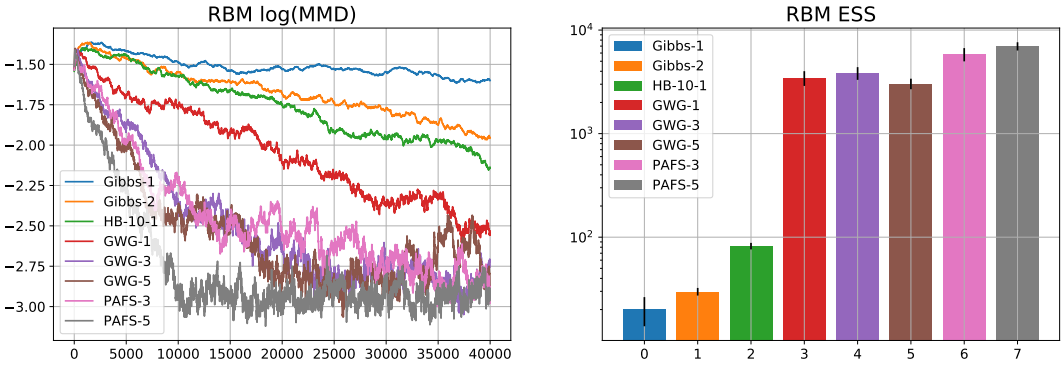

Figure 4: Sampling on RBM trained on MNIST

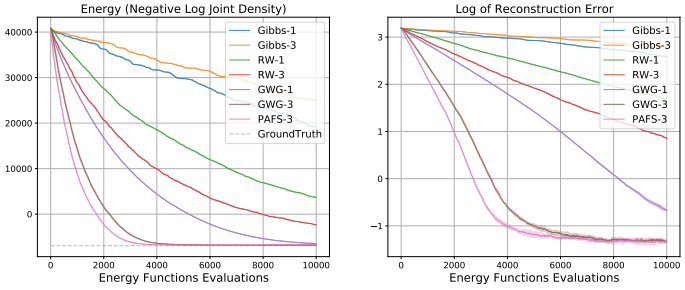

Figure 5: Sampling on Factorized Hidden Markov Model

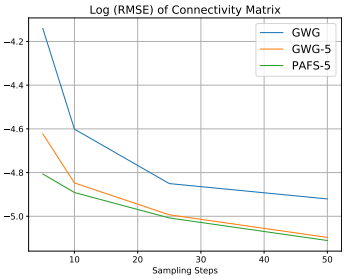

Figure 6: Learning Ising model

for likelihood:

$$p(x) = \prod_{n=1}^{N} p(x_{n,1}) \prod_{k=2}^{K} p(x_{n,k}|x_{n,k-1}) \tag{13}$$

$$p(y|x) = \prod_{n=1}^{N} \text{Gaussian}(y_n; wx_n + b, \sigma^2) \tag{14}$$

Given data y, we use MCMC to sample x from the posterior $p(x|y)$ and compare the mixing for different samplers. We choose parameters $N = 1000, K = 10$, $\mathbb{P}(x_{n,1} = 1) = 0.05$, $\mathbb{P}(X_{n,k} = x_{n,k-1}) = 0.85$, $w \in \text{Gaussian}(0, I_K)$, $b \in \text{Gaussian}(0, 1)$, and $\sigma^2 = 0.25$. We run each chain 5 times and report the mean and std for the energy (negative log joint density) $E(\hat{x}) = -\log p(\hat{x})p(y|\hat{x})$ and the reconstruction error $\|y - w\hat{x}\|_2$ w.r.t. the number of evaluations of energy function. From figure 5 we can see that both GWG and PAFS gain significant improvements in mixing by using a larger neighborhood.

The results show that a larger neighborhood in PAS and PAFS can substantially improve the sampling efficiency w.r.t. both ESS and mixing. Also, we notice that GWG has significant improvement in mixing time but not in ESS. The reason is that GWG samples all the indices based on the linearization at σ_0. During mixing, this approximation does not hurt much, since modifying a lot of dimensions can reduce the energy. However, when the chain reaches stationary distribution, only changing a small subset of dimensions is preferable in PIP. As a result, GWG has a high probability to backtrack, which means selecting an index twice and resulting in no movement.

5.4 Learning on Energy Based Model

Learning an EBM is a challenge task. Consider the target distribution is π and our energy function f_θ is parameterized by θ. The gradient for the likelihood of $\pi_\theta(x) \propto e^{-f_\theta(x)}$ is:

$$\nabla_\theta \log p(x) = \mathbb{E}_\pi[\nabla_\theta f_\theta(x)] - \mathbb{E}_{\pi_\theta}[\nabla_\theta f_\theta(x)] \tag{15}$$

Dataset	VAE (MLP)	VAE (Conv)	EBM (GWG)	EBM (Gibbs)	EBM (PAFS)	RBM	DBN
Static MNIST	-86.05	-82.41	-80.01	-117.17	**-79.58**	-86.39	-85.67
Dynamic MNIST	-82.42	-80.40	-80.51	-121.19	**-79.59**	-	-
Omniglot	-103.52	-97.65	-94.72	-142.06	**-90.75**	-100.47	-100.78
Caltech Silhouettes	-112.08	-106.35	-96.20	-163.50	**-84.56**	-	-

Table 1: Evaluation of different discrete models on the held-out test set.

The first expectation can be estimated using the data from true distribution. The second expectation requires samples from the current model, which is typically obtained via MCMC. Hence, the success of training an EBM relies on efficient MCMC algorithms to get an accurate estimation of the second expectation.

Ising model We first learn an Ising model following the setting in Grathwohl et al. (2021), where the energy function $f(x) = \theta x^T Jx$. Given a set of samples $\{x_i\}$, the task is to learn an EBM via recovering the connectivity matrix J. We generate a 25×25 2D cyclic lattice with $\theta = 0.25$ and sample training data with a long-run Gibbs chain. We train the models to maximize the likelihood of samples using persistent contrastive divergence (PCD)(Tieleman & Hinton, 2009) and report the root mean squared error (RMSE) between the inferred connectivity matrix $\hat{J}$ and the true matrix J. figure 6 shows increasing the neighborhood size can effectively help learn the Ising model. Also, GWG-5 has similar performance as PAFS-5 indicating that, using PCD, the learning efficiency of Ising model is mainly determined by the mixing of the Markov chain.

Deep EBM We evaluate path auxiliary sampler by learning a deep EBM. The experiment follows the setting in (Grathwohl et al., 2021). We train deep EBMs paramterized by Residual Networks(He et al., 2016) on small binary image datasets using PCD(Tieleman & Hinton, 2009) with a replay buffer(Du & Mordatch, 2019). We compare our methods with Variational Autoencoders Kingma & Welling (2013), GWG (Grathwohl et al., 2021), RBM and a Deep Belief Network(Hinton, 2009). We estimate the likelihoods using Annealed Importance Sampling(Neal, 2001) and report the results in table.1. The results for VAE is taken from Tomczak & Welling (2018), RBM and DBN are taken from Burda et al. (2015), GWG is taken from Grathwohl et al. (2021). We can see that our approach improves the log-likelihoods for deep EBMs on all datasets.

6 Discussion and Conclusion

In this work, we present our path auxiliary sampler for MCMC. Our approach introduces an auxiliary path for proposal, which allows to efficiently explore a large neighborhood. We demonstrate our approach on a wide range of experiments for inference, sampling, and learning on EBMs. The results show that our approach considerably outperforms existing generic samplers.

In this work, we use a uniform prior α for path length to assure the chain is aperiodic. In general, we believe a more sophisticated rule to stop the auxiliary path, e.g. $\alpha(\cdot|x)$ that depends on the current state, can further improve the sampler. By analogy to the continuous case, informed proposal uses the local energy change as the gradient to bias the proposal distribution. Single step PIP use MH adjustment after every proposal is similar to MALA(Roberts & Rosenthal, 1998), and our path auxiliary sampler, use MH adjustment after a sequence of movements, is more similar to HMC(Neal, 2004). Our future work is to figure out whether there exists a good stop rule for path auxiliary sampler that automatically adjust the path length as in Hoffman et al. (2014) for HMC.

In PAFS, we use linearization to estimate the energy function. One challenge is that a function can have the same values but different gradients in discrete space. This phenomenon indicates not only the energy function but also its gradient are important in learning EBMs in discrete space. We will explore this topic in future work.

In summary, informed proposal has shown good results for inference, sampling, and learning EBMs in discrete spaces. Our path auxiliary sampler provides an approach allowing informed proposal to efficiently explore large neighborhoods. We believe there is considerable room for future works to improve the sampling methods in discrete space.

REFERENCES

Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to mcmc for machine learning. *Machine learning*, 50(1):5–43, 2003.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Accurate and conservative estimates of mrf log-likelihood using reverse annealing. In *Artificial Intelligence and Statistics*, pp. 102–110. PMLR, 2015.

Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*, 2019.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. PMLR, 2017.

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

Will Grathwohl, Kevin Swersky, Milad Hashemi, David Duvenaud, and Chris J Maddison. Oops i took a gradient: Scalable sampling for discrete distributions. *arXiv preprint arXiv:2102.04509*, 2021.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Jun Han and Qiang Liu. Stein variational gradient descent without gradient. In *International Conference on Machine Learning*, pp. 1900–1908. PMLR, 2018.

W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

Geoffrey E Hinton. Deep belief networks. *Scholarpedia*, 4(5):5947, 2009.

Matthew D Hoffman, Andrew 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.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

Radford M Neal. Annealed importance sampling. *Statistics and computing*, 11(2):125–139, 2001.

Radford M Neal. Improving asymptotic variance of mcmc estimators: Non-reversible chains are better. *arXiv preprint math/0407281*, 2004.

Akihiko Nishimura, David Dunson, and Jianfeng Lu. Discontinuous hamiltonian monte carlo for sampling discrete parameters. *arXiv preprint arXiv:1705.08510*, 853, 2017.

Ari Pakman and Liam Paninski. Auxiliary-variable exact hamiltonian monte carlo samplers for binary distributions. *arXiv preprint arXiv:1311.2166*, 2013.

Peter H Peskun. Optimum monte-carlo sampling using markov chains. *Biometrika*, 60(3):607–612, 1973.

Samuel Power and Jacob Vorstrup Goldman. Accelerated sampling on discrete spaces with non-reversible markov processes. *arXiv preprint arXiv:1912.04681*, 2019.

Christian Robert and George Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

Gareth O Roberts and Jeffrey S Rosenthal. Optimal scaling of discrete approximations to langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(1): 255–268, 1998.

Peter J Rossky, JD Doll, and HL Friedman. Brownian dynamics as smart monte carlo simulation. *The Journal of Chemical Physics*, 69(10):4628–4633, 1978.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015.

Tijmen Tieleman and Geoffrey Hinton. Using fast weights to improve persistent contrastive divergence. In *Proceedings of the 26th annual international conference on machine learning*, pp. 1033–1040, 2009.

Luke Tierney. A note on metropolis-hastings kernels for general state spaces. *Annals of applied probability*, pp. 1–9, 1998.

Michalis K Titsias and Omiros Papaspiliopoulos. Auxiliary gradient-based sampling algorithms. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(4):749–767, 2018.

Michalis K Titsias and Christopher Yau. The hamming ball sampler. *Journal of the American Statistical Association*, 112(520):1598–1611, 2017.

Jakub Tomczak and Max Welling. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, pp. 1214–1223. PMLR, 2018.

Martin J Wainwright and Michael Irwin Jordan. *Graphical models, exponential families, and variational inference*. Now Publishers Inc, 2008.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688. Citeseer, 2011.

Giacomo Zanella. Informed proposals for local mcmc in discrete spaces. *Journal of the American Statistical Association*, 115(530):852–865, 2020.

Yichuan Zhang, Zoubin Ghahramani, Amos J Storkey, and Charles Sutton. Continuous relaxations for discrete hamiltonian monte carlo. *Advances in Neural Information Processing Systems*, 25: 3194–3202, 2012.

A PROOFS

A.1 PROOF FOR THEOREM 1

Proof. Denote $K(x, y)$ as the probability that x transit to a different state y, then we have:

$$\pi(x)K(x,y) \tag{16}$$

$$= \pi(x)\sum_L \alpha(L) \sum_{\substack{(\sigma,L)\in\Sigma(\mathcal{X},\mathcal{N}):\\ \sigma_0=x,\sigma_L=y}} \left[\prod_{l=1}^{L} Q_0(\sigma_{l-1},\sigma_l)\right] \min\left\{1, \frac{\pi(y)\prod_{l=1}^{L} Q_0(\sigma_l,\sigma_{l-1})}{\pi(x)\prod_{l=1}^{L} Q_0(\sigma_{l-1},\sigma_l)}\right\} \tag{17}$$

$$= \sum_L \alpha(L) \sum_{\substack{(\sigma,L)\in\Sigma(\mathcal{X},\mathcal{N}):\\ \sigma_0=x,\sigma_L=y}} \min\left\{\pi(\sigma_0)\prod_{l=1}^{L} Q_0(\sigma_{l-1},\sigma_l), \pi(\sigma_L)\prod_{l=1}^{L} Q_0(\sigma_l,\sigma_{l-1})\right\} \tag{18}$$

Since the path (σ, L) is symmetric w.r.t. its two ends, when σ is a path from x to y, σ is also a path from y to σ. Hence, if we exchange x, y, the summation is still applied on same paths. This implies the detailed balance condition $\pi(x)K(x, y) = \pi(y)K(y, x)$ □

A.2 PROOF FOR LEMMA 1

Lemma 1. *$\mathcal{X}$ is a finite state space with distribution π and a neighborhood function $\mathcal{N}$ such that every state has equal number of states. $\mathcal{F}$ is a class of weight functions, such that for any $f \in \mathcal{F}$, 1) $f : \mathbb{R}_+ \to \mathbb{R}_+$, 2) $f(1) = 1$, 3) $f(t)$ is monotonically increasing, 4) $f(t)f(\frac{1}{t})t \geq 1$. Define $\bar{f}(t) = \sqrt{f(t)f(\frac{1}{t})t}$, we have:*

$$\min_{x\in\mathcal{X}} Z_f(x) \leq \max_{y\in\mathcal{X}} Z_{\bar{f}}(y) \tag{19}$$

Proof. By definition, we know that $f(t) \leq 1$ for all $t < 1$, and $\bar{f}(t) \geq 1$ for all $t > 1$. We give a construction proof. Since $\mathcal{X}$ is finite, we can find state x_1, x_2 having highest and lowest probability, respectively. Then, we have:

$$Z_f(x_1) = \sum_{z\in\mathcal{N}(x_1)} f(\frac{\pi(z)}{\pi(x_1)}) \leq |\mathcal{N}(x_1)| = |\mathcal{N}(x_2)| \leq \sum_{z\in\mathcal{N}(x_2)} \bar{f}(\frac{\pi(z)}{\pi(x_2)}) = Z_{\bar{f}}(x_2) \tag{20}$$

□

A.3 PROOF FOR THEOREM 2

We prove the theorem by using Lemma 1.

Proof. We denote $c_g := \sup_{y\in\mathcal{N}(x)} \frac{Z_g(y)}{Z_g(x)}$. Theorem 4 in Zanella (2020) shows that when condition 1) and 2) hold, $c_g \to 1$ as $n \to \infty$. Hence, all we need to show is, for all $g \in \mathcal{F}$, the transition probability:

$$P_{\bar{g}}(x,y) \geq (\frac{1}{c_g c_{\hat{g}}})^U P_g(x,y) \tag{21}$$

Consider a fixed path (σ, L), we denote $t_{j,k} = \frac{\pi(\sigma_k)}{\pi(\sigma_j)}$. Then the probability that $x = \sigma_0$ transits to $y = \sigma_L$ is:

$$P_g(x,y|\sigma) = \prod_{l=1}^{L} Q_g(\sigma_{l-1},\sigma_l) \min\left\{1, \frac{\pi(y)\prod_{l=1}^{L} Q_g(\sigma_l,\sigma_{l-1})}{\pi(x)\prod_{l=1}^{L} Q_g(\sigma_{l-1},\sigma_l)}\right\} \tag{22}$$

$$= \min\left\{\prod_{l=1}^{L} \frac{g(t_{l-1,l})}{Z_g(\sigma_{l-1})}, \prod_{l=1}^{L} t_{l-1,l}\frac{g(t_{l,l-1})}{Z_g(\sigma_l)}\right\} \tag{23}$$

By definition of c_g, we have:

$$P_g(x,y,\sigma) \leq c_g^L \frac{\min\left\{\prod_{l=1}^L g(t_{l-1,l}), \prod_{l=1}^L t_{l-1,l} g(t_{l,l-1})\right\}}{\max_{z\in\mathcal{X}^{(n)}} Z_g^L(z)} \tag{24}$$

$$P_{\bar{g}}(x,y,\sigma) \geq \frac{1}{c_{\bar{g}}^L} \frac{\min\left\{\prod_{l=1}^L \bar{g}(t_{l-1,l}), \prod_{l=1}^L t_{l-1,l} \bar{g}(t_{l,l-1})\right\}}{\min_{z\in\mathcal{X}^{(n)}} Z_{\bar{g}}^L(z)} \tag{25}$$

Use the property of $\bar{g}$, we have

$$\min\left\{\prod_{l=1}^L g(t_{l-1,l}), \prod_{l=1}^L t_{l-1,l} g(t_{l,l-1})\right\} \leq \sqrt{\prod_{l=1}^L g(t_{l-1,l}) g(t_{l,l-1}) t_{l-1,l}} = \prod_{l=1}^L \bar{g}(t_{l-1,l}) \tag{26}$$

Combining equation 24, equation 25, equation 26, we have:

$$P_{\bar{g}}(x,y,\sigma) \geq \frac{1}{c_{\bar{g}}^L} \frac{\prod_{l=1}^L \bar{g}(t_{l-1,l})}{\min_{z\in\mathcal{X}^{(n)}} Z_{\bar{g}}^L(z)} \tag{27}$$

$$\geq \frac{1}{c_{\bar{g}}^L} \frac{\min\left\{\prod_{l=1}^L g(t_{l-1,l}), \prod_{l=1}^L t_{l-1,l} g(t_{l,l-1})\right\}}{\max_{z\in\mathcal{X}^{(n)}} Z_g^L(z)} \tag{28}$$

$$\geq \frac{1}{c_{\bar{g}}^L c_g^L} P_g(x,y) \tag{29}$$

Since the inequality holds for arbitrary auxiliary path (σ, L) and the path length $L \leq U$ by condition 3), we prove the theorem. □

A.4 Proof for Property 1

Proof.

$$A(x,y,\sigma,L) = \min\left\{1, \frac{\pi(y)\prod_{l=1}^L Q(\sigma_l,\sigma_{l-1})}{\pi(x)\prod_{l=1}^L Q(\sigma_{l-1},\sigma_l)}\right\} \tag{30}$$

$$= \min\left\{1, \left(\prod_{l=1}^L \frac{\pi(\sigma_l)}{\pi(\sigma_{l-1})}\right) \frac{\prod_{l=1}^L g(\frac{\pi(\sigma_{l-1})}{\pi(\sigma_l)})/Z_g(\sigma_l)}{\prod_{l=1}^L g(\frac{\pi(\sigma_l)}{\pi(\sigma_{l-1})})/Z_g(\sigma_{l-1})}\right\} \tag{31}$$

$$= \min\left\{1, \left(\prod_{l=1}^L \frac{\pi(\sigma_l)}{\pi(\sigma_{l-1})} \frac{g(\frac{\pi(\sigma_{l-1})}{\pi(\sigma_l)})}{g(\frac{\pi(\sigma_l)}{\pi(\sigma_{l-1})})}\right) \left(\prod_{l=1}^L \frac{Z_g(\sigma_{l-1})}{Z_g(\sigma_l)}\right)\right\} \tag{32}$$

$$= \min\left\{1, \frac{Z_g(x)}{Z_g(y)}\right\} \tag{33}$$

□

A.5 Proof for Theorem 3

Proof. For any path (σ, L), we first bound the estimation error $\tilde{f}_0(z) - \tilde{f}_0(\sigma_{l-1})$, for $z \in \mathcal{N}(\sigma_{l-1})$. Since f is M-smooth, we have:

$$f(z) - f(\sigma_{l-1}) \tag{34}$$

$$\leq \langle \nabla f(\sigma_{l-1}), z - \sigma_{l-1}\rangle + \frac{K}{2}\|z - \sigma_{l-1}\|^2 \tag{35}$$

$$= \langle \nabla f(\sigma_0), z - \sigma_{l-1}\rangle + \langle \nabla f(\sigma_{l-1}) - \nabla f(\sigma_0), z - \sigma_{l-1}\rangle + \frac{K}{2}\|z - \sigma_{l-1}\|^2 \tag{36}$$

$$\leq \langle \nabla f(\sigma_0), z - \sigma_{l-1}\rangle + K(l - \frac{1}{2}) \tag{37}$$

Similarly, we also have:

$$f(z) - f(\sigma_{l-1}) \geq \langle \nabla f(\sigma_0), z - \sigma_{l-1} \rangle + K(l + \frac{1}{2}) \tag{38}$$

With this estimation, we have:

$$\tilde{Q}_0(\sigma_{l-1}, \sigma_l) = \frac{e^{-\frac{1}{2}\langle \nabla f(\sigma_0), \sigma_l - \sigma_{l-1} \rangle}}{\sum_{z \in \mathcal{N}(\sigma_{l-1})} e^{-\frac{1}{2}\langle \nabla f(\sigma_0), \sigma_l - \sigma_{l-1} \rangle}} \tag{39}$$

$$\geq \frac{e^{-\frac{1}{2}[f(\sigma_l) - f(\sigma_{l-1}) - K(l - \frac{1}{2})]}}{\sum_{z \in \mathcal{N}(\sigma_{l-1})} e^{-\frac{1}{2}[f(z) - f(\sigma_{l-1}) + K(l + \frac{1}{2})]}} \tag{40}$$

$$= Q(\sigma_{l-1}, \sigma_l) e^{-Kl} \tag{41}$$

Similarly, we can also obtain

$$\tilde{Q}_L(\sigma_l, \sigma_{l-1}) \geq Q(\sigma_l, \sigma_{l-1}) e^{K(L-l)} \tag{42}$$

Now, consider the transition kernel

$$\tilde{P}(x, y) = \sum_L \alpha(L) \sum_{\substack{(\sigma, L) \in \Sigma(\mathcal{X}, \mathcal{N}): \\ \sigma_0 = x, \sigma_L = y}} \prod_{l=1}^{L} \tilde{Q}_0(\sigma_{l-1}, \sigma_l) \min \left\{ 1, \frac{\pi(y)}{\pi(x)} \frac{\prod_{l=1}^{L} \tilde{Q}_L(\sigma_l, \sigma_{l-1})}{\prod_{l=1}^{L} \tilde{Q}_0(\sigma_{l-1}, \sigma_l)} \right\} \tag{43}$$

$$= \sum_L \alpha(L) \sum_{\substack{(\sigma, L) \in \Sigma(\mathcal{X}, \mathcal{N}): \\ \sigma_0 = x, \sigma_L = y}} \min \left\{ \prod_{l=1}^{L} \tilde{Q}_0(\sigma_{l-1}, \sigma_l), \frac{\pi(y)}{\pi(x)} \prod_{l=1}^{L} \tilde{Q}_L(\sigma_l, \sigma_{l-1}) \right\} \tag{44}$$

$$\geq \sum_L \alpha(L) \sum_{\substack{(\sigma, L) \in \Sigma(\mathcal{X}, \mathcal{N}): \\ \sigma_0 = x, \sigma_L = y}} \min \left\{ \prod_{l=1}^{L} Q(\sigma_{l-1}, \sigma_l) e^{-Kl}, \frac{\pi(y)}{\pi(x)} \prod_{l=1}^{L} Q(\sigma_l, \sigma_{l-1}) e^{-K(L-l)} \right\} \tag{45}$$

$$\geq \sum_L \alpha(L) \sum_{\substack{(\sigma, L) \in \Sigma(\mathcal{X}, \mathcal{N}): \\ \sigma_0 = x, \sigma_L = y}} e^{-K\frac{L(L+1)}{2}} \min \left\{ \prod_{l=1}^{L} Q(\sigma_{l-1}, \sigma_l), \frac{\pi(y)}{\pi(x)} \prod_{l=1}^{L} Q(\sigma_l, \sigma_{l-1}) \right\} \tag{46}$$

$$\geq e^{-K\frac{U(U+1)}{2}} P(x, y) \tag{47}$$

□

B Details for Experiments

B.1 Sampling on Ising

Following Zanella (2020), we set the interaction term $\lambda = 1$, and we set $\alpha_i = \mu + Z_i$ if i in the center of the square lattice, and $\alpha_i = -\mu + Z_i$ otherwise, where $\mu = 2$ and $Z_i \sim \text{Unif}(-3, 3)$. We generate the lattice Ising model in four sizes: $p = 50, 100, 150, 200$, and for each size we generate 5 different instances. For each instance, we run a Markov chain for 1,000,000 steps and we report the effective sample size (ESS) of summary statistics over the last 500,000 steps, where the summary statistics is the hamming distance between a fixed state. Since different sampler has distinct complexity, we also report the normalized effective sampler size (NESS), which is the ESS per 500,000 evaluations of the energy function. One thing to notice is that evaluating the neighborhood or evaluating the gradient have the same complexity as evaluating the energy function due to the structure of Lattice Ising model, hence we consider all of them as one evaluation of energy function.

The results for each problem size is here:

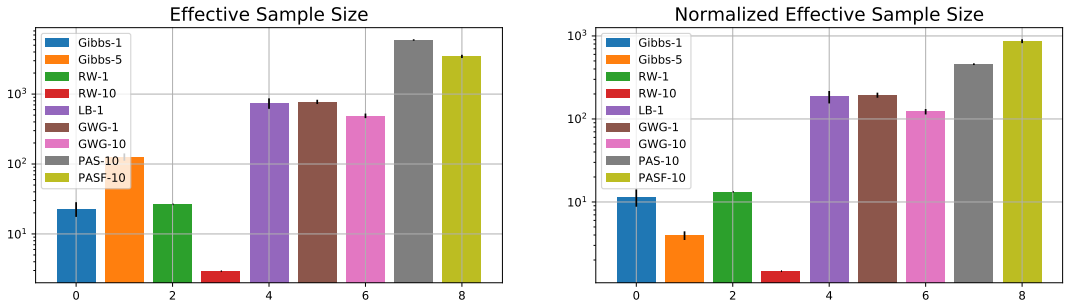

Figure 7: Sampling on 50×50 Ising model

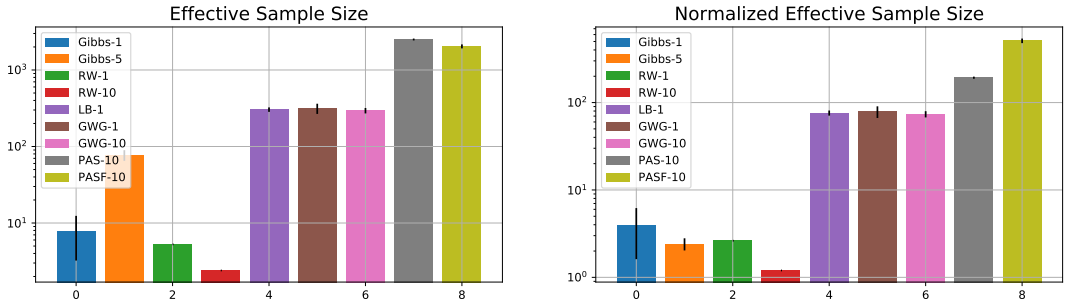

Figure 8: Sampling on 100×100 Ising model

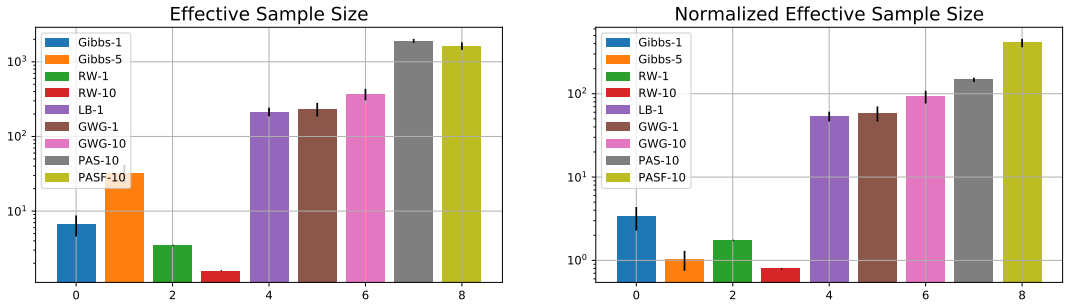

Figure 9: Sampling on 150×150 Ising model

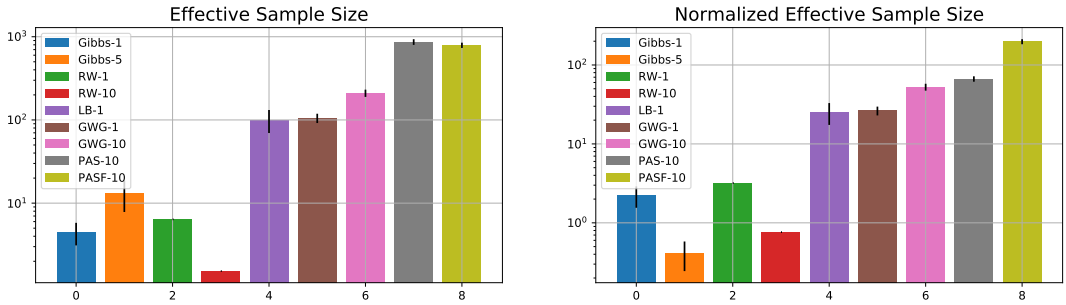

Figure 10: Sampling on 200×200 Ising model