Accelerating Discrete Langevin Samplers via Continuous Intermediates

Anonymous authors
Paper under double-blind review

Abstract

Sampling from discrete distributions remains a challenge in machine learning, with traditional Markov chain Monte Carlo (MCMC) methods such as Gibbs sampling suffering from inefficiency due to single-coordinate updates. Recent gradient-based discrete samplers have improved performance but remain constrained by the original discrete structures. To address this issue, we propose a hybrid approach that enables more global and informed proposals by introducing a continuous exploratory intermediate before the discrete update. This method, called Discrete Langevin Samplers via Continuous intermediates (cDLS), bridges the gap between discrete and continuous sampling and significantly accelerates convergence speed while maintaining theoretical guarantees. We develop variants of cDLS to ensure broad applicability, including unadjusted and Metropolis-adjusted versions. Experiments on Ising models, restricted Boltzmann machines, deep energy-based models, and Bayesian binary neural networks validate the superior performance of cDLS compared to existing methods. Our results highlight the potential of hybrid continuous-discrete exploration for advancing general discrete sampling.

1 Introduction

Discrete random variables are pervasive, ranging from text data to complex genetic data. As machine learning advances, the need for effective sampling algorithms for discrete distributions has become both urgent and essential. Markov Chain Monte Carlo (MCMC) is a classical sampling method for complex distributions, and its effectiveness highly depends on the proposal. In discrete settings, Gibbs sampling is widely used. However, it suffers in high dimensions due to updating one variable at a time. To address this, recent work has incorporated gradient information into discrete sampling, enabling more effective exploration and faster convergence (Grathwohl et al., 2021; Zhang et al., 2022; Sun et al., 2022b; 2023a;b; Xiang et al., 2023; Pynadath et al., 2024). These methods have shown promising results in both theory and practice, advancing the state of the art in high-dimensional discrete sampling.

Although gradient-based samplers assume the existence of a natural continuous extension, they operate only in discrete spaces. This limits their ability to exploit smooth structural information in the target distribution. Restricting updates to discrete points hinders large moves, slowing exploration and convergence, especially in high-dimensional settings.

To address this limitation, we propose a kind of hybrid sampler, *Discrete Langevin Samplers via Continuous Intermediates* (cDLS), which introduces a continuous intermediate step to accelerate the discrete Langevin sampler (Zhang et al., 2022). By temporarily relaxing discrete variables into a continuous space, cDLS uses gradient information more effectively, enabling smoother and more directed transitions. Conceptually, it constructs auxiliary continuous paths to accelerate exploration: starting at a discrete point, traversing a continuous intermediate guided by gradient, and returning to the discrete space.

Specifically, cDLS operates as follows. Starting from a discrete state, the sampler constructs a short continuous path guided by gradient. This continuous path captures local geometric information from the underlying energy landscape, allowing the sampler to propose transitions that are more informed and efficient than purely discrete updates. The endpoint of the continuous path is then projected back to a discrete point, completing

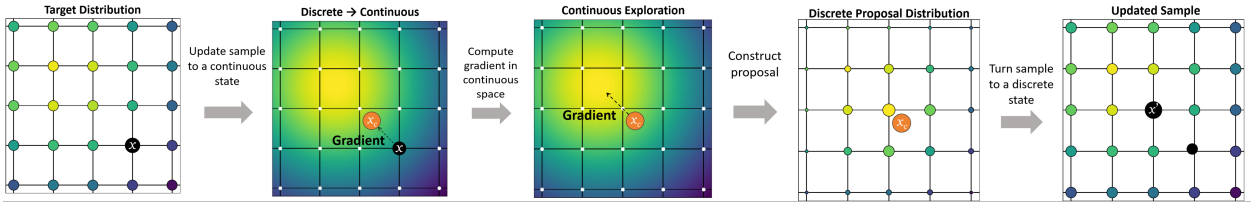


Figure 1: Overview of the proposed cDLS. Starting from a discrete state, the sampler performs a continuous update using gradient, then constructs a discrete proposal using discrete Langevin, and finally transitions to a new discrete state.

one sampling step. Compared to standard discrete samplers, this mechanism allows cDLS to make large moves and explore the space more effectively. A visualization of this process is shown in Figure 1.

We summarize our contributions as follows:

- We propose discrete Langevin samplers via continuous intermediates (cDLS), a novel hybrid sampling framework that leverages continuous space exploration to accelerate discrete Langevin sampling.
- We theoretically prove the correctness and efficiency of cDLS. We show that it achieves zero asymptotic bias for log-quadratic distributions without a Metropolis-Hastings correction, and it has small bias for distributions close to log-quadratic. Furthermore, we establish non-asymptotic convergence and inference guarantees for general discrete distributions.
- We demonstrate the effectiveness of our proposed algorithm on several experiments, including Ising models, restricted Boltzmann machines, deep energy-based models, and binary Bayesian neural networks. Notably, cDLS can achieve strong performance and low bias even without the use of Metropolis-Hastings correction.

2 Related Work

Gradient-based Discrete Sampling Efficient sampling from high-dimensional discrete distributions remains challenging due to the lack of informative proposals and poor scalability of traditional methods. Gradient-based approaches, such as Gibbs with Gradients (GWG)(Grathwohl et al., 2021) approximate locally balanced proposals (Zanella, 2020) using first-order Taylor expansion to guide coordinate selection, significantly reducing computational cost. Then, Sun et al. (2022b) introduced auxiliary path variables to extend local neighborhoods while maintaining gradient-based advantages. However, these methods still perform sequential updates and are limited by local neighborhood constraints. Inspired by Langevin dynamics in continuous space (Roberts & Stramer, 2002), Zhang et al. (2022) introduced a parallel update scheme for discrete sampling using a Langevin-like proposal. Follow-up works further extend this line by handling non-differentiable objectives (Xiang et al., 2023) and introducing adaptive scaling (Sun et al., 2023a; 2022a; Pynadath et al., 2024). In contrast, our work aims to accelerate gradient-based discrete sampling by more fully exploiting the underlying continuous structure, rather than relying solely on gradients. This introduces a new way to utilize continuous information, bridging the gap between discrete and continuous sampling approaches.

Continuous Relaxation Incorporating gradients in the proposal has been a great success in continuous space, such as the Langevin algorithm, Hamiltonian Monte Carlo (HMC), and their variants (Duane et al., 1987; Neal et al., 2011). To take advantage of this success, continuous relaxation is developed, which samples in a continuous space and then transforms the collected samples to the original discrete space (Han et al., 2020; Nishimura et al., 2020). As shown in previous work, this type of method usually does not scale to high-dimensional discrete distributions (Grathwohl et al., 2021). Relaxation-based methods conduct most sampling steps in the continuous space, with discretization applied only at the end. In contrast, our approach

only incorporates a continuous intermediate step within each discrete update. This design preserves the discrete nature of the target space throughout the sampling process. The continuous intermediates in cDLS are not used to replace discrete sampling but instead act as local exploratory guides that improve discrete proposal construction and accelerate convergence.

3 Preliminaries

We aim to sample from a discrete target distribution of the form:

$$\pi(\theta) = \frac{1}{Z}\exp(U(\theta)), \quad \theta \in \Theta,$$

where θ is a d-dimensional variable, Θ is a finite variable domain, U is the energy function, and Z is the normalizing constant for π to be a distribution. In this paper, we have the following assumptions according to the literature on gradient-based discrete sampling (Grathwohl et al., 2021; Sun et al., 2022b; Zhang et al., 2022): (1) The energy function U can be extended to a differentiable function in $\mathcal{R}^d$. (2) The domain is factorized coordinate-wise, $\Theta = \prod_{i=1}^{d} \theta_i$.

Langevin Algorithm The Langevin algorithm is a sampling method that uses gradient information with stochastic noise to generate samples from complex distributions. Given a target distribution equation 3, the Langevin dynamics are defined by the stochastic differential equation:

$$\mathrm{d}\theta_t = \nabla U(\theta_t)\mathrm{d}t + \sqrt{2}\mathrm{d}W_t \tag{1}$$

where W_t denotes the standard Brownian motion. The discretized version with stepsize α updates samples through:

$$\theta_{t+1} = \theta_t + \frac{\alpha}{2}\nabla U(\theta_t) + \sqrt{\alpha}\xi_t \tag{2}$$

where $\xi_t \sim \mathcal{N}(0, I_{d\times d})$. This algorithm efficiently explores probability spaces by combining deterministic gradient with Gaussian noise, making it widely applicable in Bayesian inference, energy-based models, and other machine learning tasks (Roberts & Stramer, 2002).

Discrete Langevin Proposal Discrete Langevin Proposal (DLP) is a counterpart of the Langevin algorithm in discrete domains (Zhang et al., 2022). Given a current state θ, the proposal for $\theta' \in \Theta$ is

$$q(\theta'|\theta) = \frac{\exp\left(-\frac{1}{2\alpha}\left\|\theta' - \theta - \frac{\alpha}{2}\nabla U(\theta)\right\|_2^2\right)}{Z_\Theta(\theta)}, \tag{3}$$

where $Z_\Theta(\theta)$ is the normalizing constant over the discrete domain. The proposal admits a coordinate-wise factorization:

$$q(\theta_i'|\theta) \propto \exp\left(\frac{1}{2}\nabla U(\theta)_i(\theta_i' - \theta_i) - \frac{(\theta_i' - \theta_i)^2}{2\alpha}\right), \quad \theta_i' \in \Theta_i. \tag{4}$$

Like other gradient-based discrete sampling methods, DLP performs updates only in discrete space, which limits its ability to exploit the smooth structural information in the target distributions.

4 Discrete Langevin Sampler via Continuous Intermediates

In this section, we propose the *Discrete Langevin Samplers via Continuous Intermediates* (cDLS), a novel discrete sampler that improves discrete Langevin sampling by leveraging continuous intermediates. The core idea is to temporarily escape the discrete domain via a gradient-informed extension in continuous space, and then return to the discrete space through a projection and a discrete update. This allows the sampler to better utilize the underlying geometry of the energy function U to facilitate more informed and exploratory updates. Unlike existing methods that operate entirely in the discrete domain, cDLS introduces a short excursion into continuous space to exploit gradient information more effectively.

4.1 Continuous Intermediates via Gradient Flow

Given the current discrete state $\theta \in \Theta$, we define a **continuous intermediate** θ^c by first taking a gradient ascent step in continuous space:

$$\theta^c_{\text{raw}} = \theta + \frac{\alpha_0}{2}\nabla U(\theta), \tag{5}$$

where $\alpha_0 > 0$ controls the magnitude of the exploration. Since θ^c_{raw} may be far from the feasible domain, we project it back into a convex region $K \subset \mathbb{R}^d$ that relaxes the original discrete domain Θ:

$$\theta^c = \text{Proj}_K(\theta^c_{\text{raw}}) = \arg\min_{y \in K} \|y - \theta^c_{\text{raw}}\|. \tag{6}$$

This projected point θ^c serves as a **continuous intermediate** from which we construct a discrete proposal. K is usually easy to decide. For example, if $\theta \in \{0,1\}^d$, then K can be $[0,1]^d$. Such a projection is natural from the intrinsic structure of discrete spaces, ensuring that the continuous intermediates remain close to the feasible region. If θ_c moves substantially away from the initial discrete distribution space, this kind of update will be meaningless. The main purpose of this step is to inject gradient flow into the sampling process without permanently leaving the discrete space.

4.2 Discrete Langevin Proposal from Continuous Intermediates

Once the continuous intermediate θ^c is obtained, we construct a discrete proposal $\theta' \in \Theta$ using a Langevin-inspired method. We adopt the structure of the Discrete Langevin Proposal (DLP) (Zhang et al., 2022), but reinterpret it as centered at θ^c:

$$q(\theta'|\theta^c) = \frac{\exp\left(-\frac{1}{2\alpha}\left\|\theta' - \theta^c - \frac{\alpha}{2}\nabla U(\theta^c)\right\|^2\right)}{Z_\Theta(\theta^c)}, \tag{7}$$

where $Z_\Theta(\theta^c)$ normalizes the distribution over the discrete domain Θ. Since Θ is coordinate-factorized according to assumption, this proposal admits a product decomposition:

$$q(\theta'|\theta^c) = \prod_{i=1}^{d} q_i(\theta'_i|\theta^c), \tag{8}$$

where each q_i is a categorical distribution defined as:

$$q_i(\theta'_i|\theta^c) \propto \exp\left(\frac{1}{2}\nabla U(\theta^c)_i(\theta'_i - \theta^c_i) - \frac{(\theta'_i - \theta^c_i)^2}{2\alpha}\right), \quad \theta'_i \in \Theta_i. \tag{9}$$

This form allows for parallelized sampling of each dimension and can be efficiently implemented. Crucially, by using θ^c instead of θ as the anchor, we incorporate richer gradient information and enable proposals that move beyond the local discrete neighborhood.

4.3 Metropolis-Hastings Correction

To ensure the resulting Markov chain has the desired stationary distribution $\pi(\theta)$, we optionally apply a Metropolis-Hastings (MH) correction (Metropolis et al., 1953; Hastings, 1970), which is usually combined with proposals to make the Markov chain reversible. Given the proposal $\theta' \sim q(\cdot|\theta^c)$, we compute the acceptance probability as:

$$A(\theta \to \theta') = \min\left(1, \exp\left(U(\theta') - U(\theta)\right) \cdot \frac{q(\theta|\theta'_c)}{q(\theta'|\theta^c)}\right), \tag{10}$$

where θ^c and θ'_c are the continuous intermediates corresponding to θ and θ', respectively. This correction is necessary when exact reversibility is required, although in practice, cDLS is often found to perform well even without MH correction due to its efficient utilization of local gradient information.

4.4 Summary of the cDLS Framework

cDLS extends the capabilities of discrete Langevin samplers by introducing a gradient-guided continuous intermediate θ^c between discrete updates. This intermediate enables broader and more informed transitions while preserving the structure of discrete sampling. We outline the sampling algorithms in Algorithm 1. The different steps compared with Zhang et al. (2022) are highlighted in blue. We call the sampler with MH correction as *Discrete Metropolis Adjusted Langevin Algorithm via Continuous intermediates* (cDMALA) and without MH correction as *Discrete Unadjusted Langevin Algorithm via Continuous intermediates*(cDULA).

5 Convergence Analysis

In this section, we present a theoretical analysis of the proposed cDLS framework, including both the asymptotic convergence of cDULA and the non-asymptotic efficiency of cDMALA. Specifically:

- In Section 5.1, we show that for log-quadratic target distributions, cDULA introduces no asymptotic bias as the stepsize $\alpha \to 0$.
- In Section 5.2, we generalize this result to distributions that are approximately log-quadratic and characterize the bias in terms of deviation from linearity.
- In Section 5.3, we analyze cDMALA and prove its uniform ergodicity, providing guarantees on the convergence rate and statistical inference.

Algorithm 1: Discrete Langevin Samplers via Continuous Intermediates

Input: Stepsize α, α_0; Initial Sample θ_0.
while *true* **do**
  for $i = 1, 2, \ldots, d$ **do**
    Explore $\theta^c \leftarrow \theta + \frac{\alpha_0}{2}\nabla U(\theta)$;
    Map θ^c as in Equation 6;
    Construct $q_i(\cdot|\theta^c)$ as in Equation 7;
    Sample $\theta'_i \sim q_i(\cdot|\theta^c)$;
  end
  `// Optionally, do the MH step`
  Compute $q(\theta'|\theta^c) = \prod_i q_i(\theta'_i|\theta^c)$;
  Explore $\theta'_c \leftarrow \theta' + \frac{\alpha_0}{2}\nabla U(\theta')$;
  Map θ'_c as in Equation 6;
  Compute $q(\theta|\theta'_c) = \prod_i q_i(\theta_i|\theta'_c)$;
  if *with MH step* **then**
    Set $\theta \leftarrow \theta'$ with probability in Equation 10;
  else
    Set $\theta \leftarrow \theta'$;
  end
end
Output: Samples $\{\theta_k\}$;

5.1 Convergence on Log-Quadratic Distributions

We begin with a special but fundamental case where the target distribution $\pi(\theta)$ has a log-quadratic form:

$$\pi(\theta) \propto \exp(\theta^T W\theta + b^T\theta), \quad \theta \in \Theta, \qquad (11)$$

where $W \in \mathbb{R}^{d\times d}$ is assumed symmetric (otherwise replace with $(W+W^T)/2$), and $b \in \mathbb{R}^d$. Such distributions frequently arise in structured models, like Ising model and Gaussian graphical models.

Theorem 1. *Assume the target distribution π is log-quadratic as in Equation 11. Let $\lambda_{\min}$ be the smallest eigenvalue of W. Assume $\alpha_0 = o(\alpha)$ and consider any $\alpha > 0$, $\alpha_0 \geq 0$. Then:*

i The Markov chain defined by the proposal $q(\cdot|\theta)$ in Equation 7 is reversible.

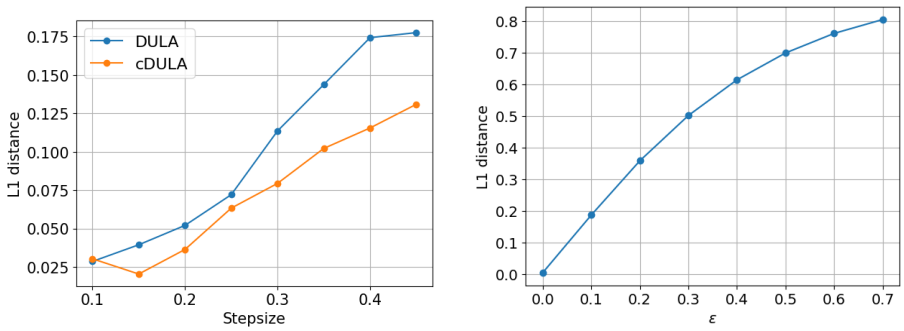
ii The stationary distribution π_{α,α_0} of the chain converges weakly to π as $\alpha \to 0$. Moreover, the convergence in L1 distance is bounded as:

$$\|\pi_{\alpha,\alpha_0} - \pi\|_1 \leq cZ \cdot \exp\left(-\frac{1}{2\alpha} - \frac{(\alpha+\alpha_0)\lambda_{\min}}{2\alpha}\right),$$

where Z is the normalizing constant of π, and $c = \exp(\alpha_0 C \cdot D/4)$ depends on α_0 and $D = \max_{\theta,\theta'\in\Theta} \|\theta' - \theta\|_\infty$.

Theorem 1 shows that the bias of cDULA vanishes exponentially as $\alpha \to 0$, implying asymptotic correctness. Moreover, for any fixed $\alpha > 0$ and $\lambda_{\min} \geq 0$, the bound is strictly tighter than in the baseline case $\alpha_0 = 0$ (i.e., DULA), by an additional factor of $\exp\left(-\frac{\alpha_0 \lambda_{\min}}{2\alpha}\right)$, up to a mild multiplicative constant. Intuitively, the continuous intermediates steer the state toward high-probability regions before the discrete update, reducing the local linearization error that drives discretization bias under quadratic energies. In practice, this allows cDULA to maintain low bias even with moderate step sizes, reducing the need for fine-tuning and lengthy trajectories.

Figure 2(a) illustrates the behavior of cDULA with varying stepsizes on a 2×2 Ising model. As expected, the discrepancy between the stationary distribution and the target π diminishes as α decreases. Additionally, cDULA demonstrates greater robustness compared to DULA under larger α, validating the theoretical claim that continuous exploration mitigates bias and enables larger, more efficient steps.



(a) (b)

Figure 2: Verification of Theorem 1 and 2. (a) cDULA with varying stepsizes on 2×2 Ising model. (b) cDULA on 1-d distribution with varying closeness ϵ to being log-quadratic.

5.2 Convergence on General Distributions

We now extend the asymptotic analysis of cDULA to a broader class of target distributions. While Theorem 1 establishes exact convergence for log-quadratic targets, real-world energy functions often deviate from strict quadratic forms. To handle such cases, we quantify how closely the gradient of the energy function U approximates a linear form.

Following Zhang et al. (2022), we assume there exists $W \in \mathbb{R}^{d \times d}$, $b \in \mathbb{R}^d$, and a small constant $\epsilon > 0$ such that

$$\|\nabla U(\theta) - (2W\theta + b)\|_1 \leq \epsilon, \quad \forall \theta \in \Theta. \tag{12}$$

This assumption states that the energy function is approximately log-quadratic in the gradient sense.

Theorem 2. *Let π be the target distribution and $\pi'(\theta) \propto \exp(\theta^T W \theta + b^T \theta)$ be the log-quadratic approximation to π as in Equation 12. Then the stationary distribution π_{α,α_0} of cDULA satisfies:*

$$\|\pi_{\alpha,\alpha_0} - \pi\|_1 \leq 2c_1(\exp(c_2\epsilon) - 1) + Z' \exp\left(-\frac{1}{2\alpha} - \frac{(\alpha + \alpha_0)\lambda_{\min}}{2\alpha}\right),$$

where c_1 depends on π' and (α, α_0), c_2 depends on Θ and $\max_{\theta,\theta'} \|\theta - \theta'\|_\infty$, and Z' is the normalizing constant for π'.

This result decomposes the total bias into two parts: The first term captures how far the true energy function U is from a quadratic form, with error scaling exponentially in ϵ. Thus, as $\epsilon \to 0$, the behavior of cDULA approaches that in Theorem 1; The second term is inherited from the previous theorem and quantifies the bias introduced by using non-zero stepsizes.

Together, these results offer practical guidance: to reduce bias, one should choose a continuous extension whose gradient is close to linear over the discrete support Θ. To validate Theorem 2, we construct a 1D distribution of the form:

$$\pi(\theta) \propto \exp\left(a\theta^2 + b\theta + 2\epsilon \sin(\theta\pi/2)\right),$$

whose gradient is $\nabla U(\theta) = 2a\theta + b + \epsilon\pi\cos(\theta\pi/2)$. This gradient deviates from linearity in proportion to ϵ. As shown in Figure 2(b), the total variation bias of cDULA increases with larger ϵ, consistent with the theoretical bound. The trend closely matches that of DULA under the same perturbation, suggesting comparable robustness in mildly non-quadratic scenarios.

5.3 Non-asymptotic Convergence

Beyond asymptotic bias, we now analyze the non-asymptotic behavior of cDMALA. Our goal is to provide theoretical guarantees for the convergence rate and support statistical inference from finite samples. We establish these results by proving uniform ergodicity of the Markov chain P under standard smoothness and local concavity assumptions, which are common in the literature on optimization and sampling Dalalyan (2017); Bottou et al. (2018).

Assumption 1. *The energy function $U(\theta) \in \mathcal{C}^2(\mathcal{R})$ has L-Lipschitz gradient. That is*

$$\|\nabla U(\theta) - \nabla U(\theta')\| \leq L\|\theta - \theta'\|. \tag{13}$$

Assumption 2. *For each $\theta \in \mathcal{R}^d$, there exists an open ball containing θ of some radius r_θ, denoted by $B(\theta, r_\theta)$, such that the function $U(\cdot)$ is m_θ-strongly concave in $B(\theta, r_\theta)$ for some $m_\theta > 0$.*

Lemma 1. *Let Assumptions 1 and 2 with $\alpha + \alpha_0 < \frac{2}{L}$ hold. Then for the Markov chain P we have, for any $\theta, \theta' \in \Theta$,*

$$p(\theta|\theta') \geq \epsilon_{\alpha,\alpha_0} \frac{\exp(\tilde{\alpha}U(\theta'))}{\sum_{\theta'} \exp(\tilde{\alpha}U(\theta'))} \tag{14}$$

where

$$\epsilon_{\alpha,\alpha_0} = \exp\left\{-(\frac{2}{\alpha} + L - \frac{(\alpha+\alpha_0)m}{4\alpha})D^2 - \frac{\alpha+\alpha_0}{2\alpha}\|\nabla U(a)\| D\right\}.$$

Theorem 3. *Under Assumptions 1 and 2, and for $\alpha + \alpha_0 < \frac{2}{L}$, the Markov chain P defined by cDMALA satisfies:*

i *P is uniformly ergodic:*

$$\|P^n - \pi\|_{TV} \leq (1 - \epsilon_{\alpha,\alpha_0})^n.$$

ii *For any real-valued test function f, the sample average satisfies:*

$$\sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n} f(X_i) - \mathbb{E}_\pi[f]\right) \xrightarrow{d} \mathcal{N}(0, \sigma_*^2),$$

for some $\sigma_* > 0$ as $n \to \infty$.

Theorem 3 guarantees geometric ergodicity and a valid Central Limit Theorem for cDMALA. Compared to DMALA, the convergence coefficient $\epsilon_{\alpha,\alpha_0}$ in cDMALA incorporates contributions from both the gradient and local curvature that are scaled by $(\alpha + \alpha_0)/2\alpha$ and $(\alpha + \alpha_0)m/4\alpha$, respectively. When the auxiliary stepsize satisfies $0 < \alpha_0 = o(\alpha)$ and the target exhibits sufficient local concavity, these enhanced terms make $\epsilon_{\alpha,\alpha_0}$ larger than the corresponding DMALA coefficient, yielding faster convergence. In this sense, the continuous intermediates effectively regularize the proposal dynamics and tighten the stepsize-dependent bound, providing a clear theoretical advantage over DMALA under the specified conditions.

6 Experiments

For both sampling tasks and learning tasks, we compare our method to Gibbs sampling, Gibbs-with-Gradient (GWG) (Grathwohl et al., 2021), Discrete Unadjusted Langevin Algorithm (DULA), Discrete Metropolis Adjusted Langevin Algorithm (DMALA) (Zhang et al., 2022), and Automatic Cyclical Scheduling (Pynadath et al., 2024), which are popular and recent gradient-based discrete samplers. All methods are implemented in `Pytorch`, and we use the official release of code from previous papers when possible. More experimental details can be found in Appendix G.

6.1 Sampling From Synthetic Distribution

To demonstrate the capability of cDLS to sample faster from the general distributions, we construct a 2-D energy landscape as follows:

$$U(\theta) = \log\left(\sum_{i=1}^{l} \exp\left(\frac{\|\theta - \mu_i\|^2}{2\sigma}\right)\right), \tag{15}$$

where $\Theta = \{0, 1, \cdots, N\}^2$, N is the maximum value for each coordinate, and a set of modes $\{\mu_1, \mu_2, \cdots, \mu_l\}$. We demonstrate the results of various samplers in Figure 3, including Total Variance distance and visualization. More experimental details can be found in Appendix G.2.

(a) DULA,TV = 5.23 (b) cDULA,TV = 3.12

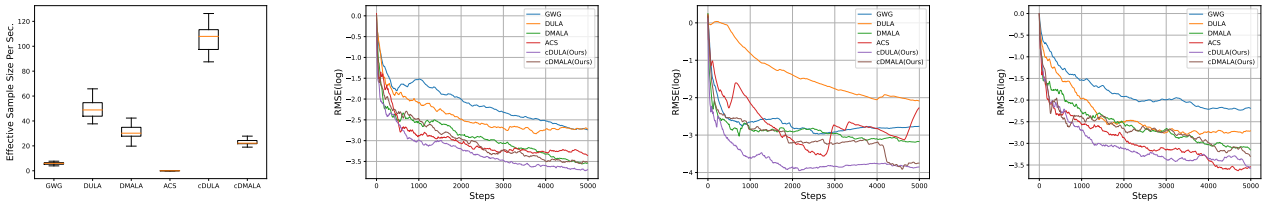
Figure 3: Synthetic Distribution Sampling. We run the DULA and cDULA sampler to compare how continuous exploration helps sample from the target distribution.

The results show that DULA focuses on a given mode, and by adding continuous space exploration, the cDULA exhibits better performance. We also visualize the sampling trajectory of DULA versus cDULA in Appendix G.2, and it can be noticed that cDULA converges to the first mode very quickly by continuous exploration; in comparison, DULA often takes an excessively long sampling time.

6.2 Sampling From Ising Models

We consider a 5 by 5 lattice Ising model with random variable $\theta \in \{-1, 1\}^d$, and $d = 5 \times 5 = 25$. The energy function is $U(\theta) = a\theta^T W\theta + b\theta$, where W is a binary adjacency matrix, $a = 0.1$ is the connectivity strength and $b = 0.2$ is the bias.

Results As seen in Figure 4, cDULA significantly outperforms other methods. In Figure 4 (a), cDULA yields the largest effective sample size (ESS) (Lenth, 2001) per second, indicating the correlation among its samples is low due to making significant updates in each step. We compare the root-mean-square error



(a) ESS (b) $a = 0.1, b = 0.2$ (c) $a = 0.2, b = 0.1$ (d) $a = 0.1, b = 0.0$

Figure 4: Ising model sampling results. (a) cDULA yields the largest effective sample size (ESS) per second and (b) achieves the lowest RMSE error among all the methods compared. (c)&(d)Also, our method maintains good performance despite the different parameters of the model. More discussion is detailed in the Appendix G.3.

(RMSE) between the estimated mean and the true mean in Figure 4 (b). This shows that exploration with continuous intermediates accelerates the convergence on this task, reaching the lowest or comparable RMSE error. In Figure 4 (c) and (d), we see that cDULA significantly outperforms baseline methods in different settings. This suggests that cDULA effectively explores the continuous space to construct more informed proposals. These results demonstrate the ability to efficiently explore and accurately characterize modes. Notably, cDLS achieved satisfactory results even without MH correction. This may be attributed to the proposal distribution constructed from continuous intermediates, which fully leverages geometric information to perform well within a limited computational budget. However, for complex scenarios, such as Section 6.4.3, which are prone to divergence, MH correction provides a strict guarantee of convergence.

6.3 Sampling From Restricted Boltzmann Machines

Restricted Boltzmann Machines (RBMs) are generative artificial neural networks, which learn an unnormalized probability over inputs,

$$U(\theta) = \sum_i \text{Softplus}(W\theta + a)_i + b^T\theta,$$

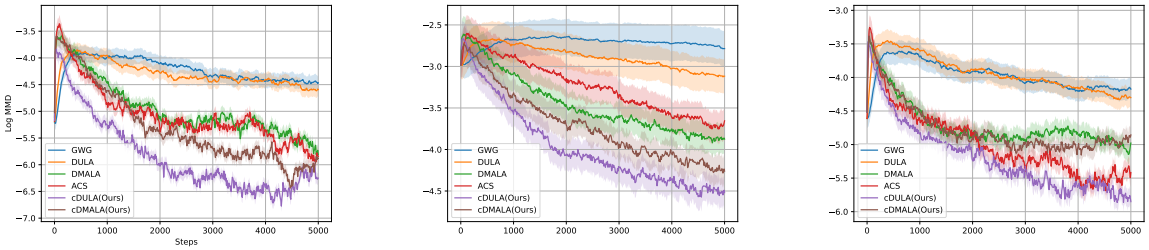
where $\{W, a, b\}$ are parameters and $\theta \in \{0, 1\}^d$. Following Grathwohl et al. (2021); Zhang et al. (2022), we randomly initialize all samplers, train $\{W, a, b\}$ with contrastive divergence (Hinton, 2002), and measure the Maximum Mean Discrepancy (MMD) between the obtained samples and those from Block-Gibbs sampling, which leverages the known structure of the RBM.

Results The results are shown in Figure 5. Our method, cDLS, especially cDULA, consistently achieves competitive or superior convergence across all datasets, with notable improvements on datasets like MNIST-like, Caltech and Omniglot. These results demonstrate the ability of the sampler to efficiently explore and accurately characterize modes. We further include the results of the generated images and runtime comparison in Appendix G.4.

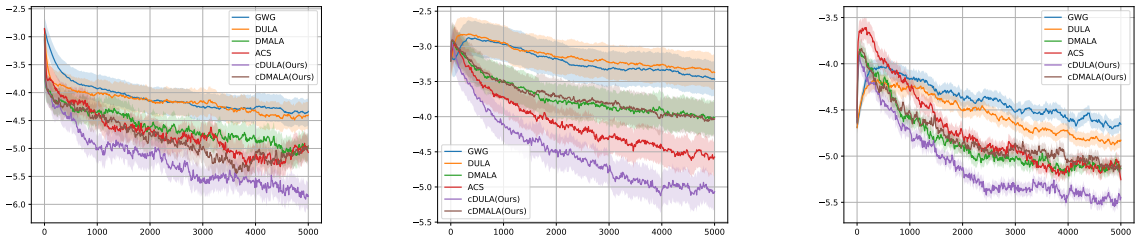
6.4 Learning Energy-based Models

Energy-based models (EBM) have achieved great success in various areas in machine learning (LeCun et al., 2006). Generally, the density of an energy-based model is defined as $f_\theta(x) = \frac{1}{Z_\theta}\exp(-E_\theta(x))$, where E_θ is a function parameterized by θ and Z_θ is the normalizing constant. Training EBM usually involves maximizing the log-likelihood, $\mathcal{L}(\theta) \triangleq \mathrm{E}_{x\sim p_{data}}[log(f_\theta(x))]$. However, direct optimization requires the computation of Z_θ, which is intractable in most scenarios. To deal with it, we usually estimate the gradient of the log-likelihood instead (Song & Kingma, 2021),

$$\nabla\mathcal{L}(\theta) = \mathbb{E}_{x\sim p_\theta}\left[\nabla_\theta E_\theta(x)\right] - \mathbb{E}_{x\sim p_{data}}\left[\nabla_\theta E_\theta(x)\right].$$



(a) MNIST (b) kMNIST (c) eMNSIT



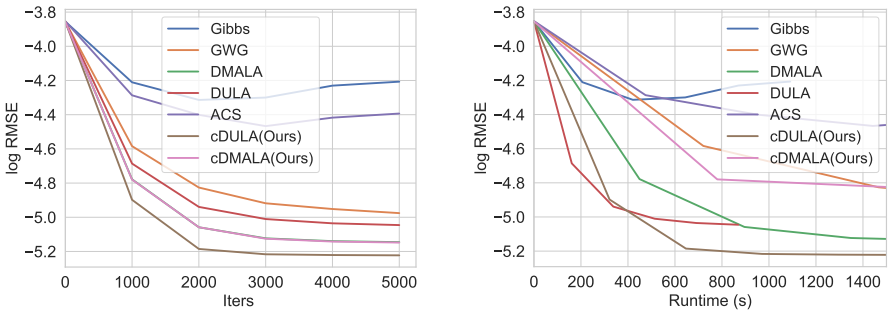
(d) Fashion (e) Caltech (f) Omniglot

Figure 5: RBM models sampling results. Our method consistently achieves competitive or superior convergence across all datasets.

Though the second term is easy to estimate from the data, the first term requires samples from p_θ. Better samplers can improve the training process of E_θ, leading to EBMs with higher performance.

6.4.1 Ising Models

As in Grathwohl et al. (2021); Zhang et al. (2022), we generate a 25 by 25 Ising model and generate data by running the Gibbs sampler. In this experiment, E_θ is an Ising model with a learnable parameter $\hat{W}$. We evaluate the samplers by computing the RMSE between the estimated $\hat{W}$ and the true W.



(a) (b)

Figure 6: Ising model learning results. cDULA outperforms all baselines, finding the true W in a short time and getting a better convergence speed.

Results Our results are summarized in Figure 6. cDULA outperforms all the compared methods, having the smallest RMSE among the baselines given the same number of iterations. Compared to the running

time, cDULA also achieves the same log-RMSE in less time, except DULA as cDULA need time to explore the continuous space.

6.4.2 Learning RBM

We demonstrate the benefits of cDLS on RBMs. We evaluate the learned model using Annealed Importance Sampling (AIS) (Neal, 2001). We compare the sampling methods of interest to Block-Gibbs (BG), which utilizes the structure of RBMs well.

Table 1: Log likelihood scores for RBM learning on test data as estimated by AIS. Our method outperforms gradient-based baselines across most datasets. ACS results are mainly taken from Pynadath et al. (2024).

Dataset	BG	GWG	DULA	DMALA	ACS	cDULA	cDMALA
MNIST	*-208.76*	-372.59	-271.17	-234.43	-249.55	**-211.24**	-346.90
eMNIST	*-337.90*	-445.68	-312.12	-349.85	-304.96	**-289.91**	-297.70
kMNIST	*-347.72*	-519.72	-381.27	-452.43	-407.39	**-380.12**	-436.04
Fashion	*-307.77*	-671.03	-471.35	-476.21	-452.37	**-406.30**	-460.93
Omniglot	*-177.83*	-338.34	**-180.66**	-267.83	-220.71	-190.47	-281.47
Caltech	*-524.34*	-593.47	-538.50	-731.18	-396.04	**-390.88**	-518.73

Results From Table 1, we note that our methods outperform the baselines on most of the datasets. Since the structure of Omniglot is quite different from the other datasets 11, it leads to insufficiency of exploration when using the same α_0. This reveals that there should be a more adaptive approach to parameter tuning for different datasets, and we leave this for future work. We include more discussion of these results and the generated images in Appendix G.5.

6.4.3 Learning EBM

We also test our methods on deep convolutional EBMs to demonstrate their efficiency. We train deep EBMs where E_θ is a ResNet (He et al., 2016) with Persistent Contrastive Divergence (Tieleman, 2008; Tieleman & Hinton, 2009) and a replay buffer (Du & Mordatch, 2019) following Grathwohl et al. (2021); Zhang et al. (2022). After training, we also adopt AIS to estimate the likelihood. From the insight of Pynadath et al. (2024), we use 10 sampling steps per iteration on all datasets except Caltech, where we adopt 30. We include more experimental details in Appendix G.6.

Results Table 2 demonstrates that our method achieves superior performance over DLP and GWG. This advantage comes from our sampler's enhanced capacity to systematically explore a broader range of modes per iteration, which accelerates the discovery of high-probability regions. Such efficient exploration directly translates into higher-quality gradient estimates during training, driving more stable and effective model updates. It is reasonable to hypothesize that cDLS with multi-step or adaptive exploration will be more advantageous in such complex experiments (Loshchilov & Hutter, 2017; Zhang et al., 2020; Pynadath et al., 2024).

6.5 Binary Bayesian Neural Networks

Bayesian neural networks have been shown to provide strong predictions and uncertainty estimation in deep learning (Hernández-Lobato & Adams, 2015; Zhang et al., 2020; Liu et al., 2021a). Meanwhile, binary neural networks (Courbariaux et al., 2016; Rastegari et al., 2016; Liu et al., 2021b), i.e., the weight is in {-1, 1}, accelerate the learning and significantly reduce computational and memory costs. To combine the benefits of both worlds, we consider training a binary Bayesian neural network with discrete sampling. We conduct regression on four UCI datasets (Dua & Graff, 2017), and the energy function is defined as,

$$U(\theta) = -\sum_{i=1}^{N} \|f_\theta(x_i) - y_i\|_2^2,$$

Table 2: Log likelihood scores for EBM learning on test data as estimated by AIS. cDLS is able to achieve better results than the baselines. Previous results are mainly taken from Pynadath et al. (2024).

Dataset	Gibbs	GWG	DULA	DMALA	ACS	cDULA	cDMALA
MNIST	-117.17	-80.01	-81.20	-79.93	-79.76	-80.42	**-79.55**
Dynamic MNIST	-121.19	-80.51	-80.06	-80.13	-79.70	**-79.50**	-79.62
Omniglot	-142.06	-94.72	-127.68	-100.08	-91.32	-111.72	**-91.01**
Caltech Silhouettes	-163.50	-96.20	-114.82	-99.35	-88.34	-102.56	**-87.82**

Table 3: Experiment results with binary Bayesian neural networks on different datasets.

Dataset	Training Log-likelihood (↑)						
	Gibbs	GWG	DULA	DMALA	ACS	cDULA	cDMALA
COMPAS	**-0.3102**$\pm$0.0029	-0.3104$\pm$0.0022	-0.3295$\pm$0.0049	-0.3122$\pm$0.0011	-0.3163$\pm$0.0002	-0.3431$\pm$0.0001	-0.3132$\pm$0.0033
News	-0.2132$\pm$0.0013	-0.2121$\pm$0.0023	-0.2116$\pm$0.0013	-0.2109$\pm$0.0001	-0.2099$\pm$0.0071	-0.2107$\pm$0.0001	**-0.2098**$\pm$0.0033
Adult	-0.3631$\pm$0.0007	-0.3249$\pm$0.0010	-0.3051$\pm$0.0001	-0.2950$\pm$0.0042	-0.3040$\pm$0.0014	**-0.2900**$\pm$0.0002	-0.3008$\pm$0.0030
Blog	-0.3746$\pm$0.0021	-0.3247$\pm$0.0004	-0.2705$\pm$0.0023	-0.2603$\pm$0.0031	-0.2654$\pm$0.0043	**-0.2601**$\pm$0.0006	-0.2607$\pm$0.0009
Dataset	**Test RMSE (↓)**						
	Gibbs	GWG	DULA	DMALA	ACS	cDULA	cDMALA
COMPAS	0.4795$\pm$0.0034	0.4774$\pm$0.0037	0.4848$\pm$0.0013	0.4775$\pm$0.0029	0.4750$\pm$0.0039	**0.4674**$\pm$0.0029	0.4789$\pm$0.0036
News	0.0964$\pm$0.0022	0.0975$\pm$0.0046	0.0994$\pm$0.0011	0.0946$\pm$0.0030	0.0992$\pm$0.0099	0.0932$\pm$0.0001	**0.0905**$\pm$0.0009
Adult	0.4284$\pm$0.0022	0.4044$\pm$0.0004	0.3900$\pm$0.0016	0.3853$\pm$0.0003	0.3919$\pm$0.0072	**0.3793**$\pm$0.0002	0.3869$\pm$0.0035
Blog	0.4040$\pm$0.0009	0.3564$\pm$0.0057	0.3186$\pm$0.0037	0.3130$\pm$0.0019	0.3168$\pm$0.0030	**0.3077**$\pm$0.0040	0.3099$\pm$0.0004

where $D = \{x_i, y_i\}_i^N$ is the training dataset, and f_θ represents a two-layer neural network with `Tanh` activation and 500 hidden neurons.

Results From Table 3, we observe that our methods significantly outperform other gradient-based discrete samplers on all datasets except COMPAS, as there are fewer data samples. It could also be found in Zhang et al. (2022). These results demonstrate that our methods converge fast for high-dimensional distributions, due to the ability to explore the parameter space using gradient, and suggest that our methods are compelling for training low-precision Bayesian neural networks where weights are discrete. Additionally, our method shows the superior generalization for data, as RMSE on test datasets is significantly decreased while the training likelihood is almost equal.

7 Conclusion

In our paper, we propose a hybrid discrete-continuous sampler for discrete spaces that extends gradient-based updates from purely discrete settings into an augmented continuous space. By allowing samples to explore along gradient directions in this expanded space, our method improves the efficiency of sampling from high-dimensional discrete distributions. For different usage scenarios, we develop several corresponding variants, which include the unadjusted and Metropolis-Hastings adjusted versions. We prove the asymptotic convergence of our sampler under log-quadratic and general distributions and non-asymptotic bounds for the MH adjusted version. Numerous experiments on many different tasks show that our method outperforms the baseline methods. As a simple and easy-to-use method, we hope that our method will improve the sampling efficiency of discrete distributions and provide a new perspective on the utilization of gradient information.

References

Leon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

Krisztian Buza. Feedback prediction for blogs. In *Data analysis, machine learning and knowledge discovery*, pp. 145–152. Springer, 2014.

Grace E Cho and Carl D Meyer. Comparison of perturbation bounds for the stationary distribution of a markov chain. *Linear Algebra and its Applications*, 335(1-3):137–150, 2001.

Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*, 2016.

Arnak S. Dalalyan. Theoretical guarantees for approximate sampling from smooth and log-concave densities. *Journal of the Royal Statistical Society Series B:Statistical Methodology*, 79(3):651–676, 2017.

Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *Advances in Neural Information Processing Systems*, 2019.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

Will Grathwohl, Kevin Swersky, Milad Hashemi, David Duvenaud, and Chris J Maddison. Oops i took a gradient: Scalable sampling for discrete distributions. *International Conference on Machine Learning*, 2021.

Jun Han, Fan Ding, Xianglong Liu, Lorenzo Torresani, Jian Peng, and Qiang Liu. Stein variational inference for discrete distributions. In *International Conference on Artificial Intelligence and Statistics*, pp. 4563–4572. PMLR, 2020.

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.

José Miguel Hernández-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International conference on machine learning*, pp. 1861–1869. PMLR, 2015.

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

S. Mattu J. Angwin, J. Larson and L. Kirchner. Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks. *ProPublica*, 2016.

Galin L. Jones. On the Markov chain central limit theorem. *arXiv Mathematics e-prints*, pp. math/0409112, 2004.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference for Learning Representations*, 2015.

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

RV Lenth. Some practical guidelines for effective sample size determination. *AMERICAN STATISTICIAN*, 55(3):187–193, 2001.

Xingchao Liu, Xin Tong, and Qiang Liu. Sampling with trusthworthy constraints: A variational gradient framework. *Advances in Neural Information Processing Systems*, 34, 2021a.

Xingchao Liu, Mao Ye, Dengyong Zhou, and Qiang Liu. Post-training quantization with multiple points: Mixed precision without mixed precision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8697–8705, 2021b.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=Skq89Scxx`.

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 et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.

Akihiko Nishimura, David B Dunson, and Jianfeng Lu. Discontinuous hamiltonian monte carlo for discrete parameters and discontinuous likelihoods. *Biometrika*, 107(2):365–380, 2020.

Patrick Pynadath, Riddhiman Bhattacharya, Arun Hariharan, and Ruqi Zhang. Gradient-based Discrete Sampling with Automatic Cyclical Scheduling. In *Neural Information Processing Systems*, 2024.

Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European conference on computer vision*, pp. 525–542. Springer, 2016.

Gareth O Roberts and Osnat Stramer. Langevin diffusions and metropolis-hastings algorithms. *Methodology and computing in applied probability*, 4(4):337–357, 2002.

Paul J Schweitzer. Perturbation theory and finite markov chains. *Journal of Applied Probability*, 5(2): 401–413, 1968.

Yang Song and Diederik P Kingma. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

Haoran Sun, Hanjun Dai, and Dale Schuurmans. Optimal scaling for locally balanced proposals in discrete spaces. In *Neural Information Processing Systems*, volume 35, pp. 23867–23880. Curran Associates, Inc., 2022a.

Haoran Sun, Hanjun Dai, Wei Xia, and Arun Ramamurthy. Path auxiliary proposal for mcmc in discrete space. In *International Conference on Learning Representations*, 2022b.

Haoran Sun, Bo Dai, Charles Sutton, Dale Schuurmans, and Hanjun Dai. Any-scale balanced samplers for discrete space. In *International Conference on Learning Representations*, 2023a.

Haoran Sun, Hanjun Dai, Bo Dai, Haomin Zhou, and Dale Schuurmans. Discrete langevin samplers via wasserstein gradient flow. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 6290–6313. PMLR, 25–27 Apr 2023b.

Tijmen Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, pp. 1064–1071, 2008.

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.

Yue Xiang, Dongyao Zhu, Bowen Lei, Dongkuan Xu, and Ruqi Zhang. Efficient informed proposals for discrete distributions via newton's series approximation. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 7288–7310. PMLR, 25–27 Apr 2023.

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

Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical stochastic gradient mcmc for bayesian deep learning. In *International Conference on Learning Representations*, 2020.

Ruqi Zhang, Xingchao Liu, and Qiang Liu. A Langevin-like sampler for discrete distributions. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 26375–26396. PMLR, 17–23 Jul 2022.

A Algorithm with Binary Variables

cDLS could be simplified when the variable domain is binary $\Theta = \{0,1\}^d$. We obtain Algorithm 2 from 1 to help understand.

Algorithm 2: Discrete Langevin Samplers via Continuous Intermediates for Binary Variables

Input: The stepsize α, α^0
while *True* **do**
end
Explore $\theta^c = \theta + \frac{\alpha_0}{2}\nabla U(\theta)$;
Map θ^c;
Compute $P(\theta|\theta^c) = \frac{\exp\left(\frac{1}{2}\nabla U(\theta^c)\odot(1-\theta-\theta^c) - \frac{(1-\theta-\theta^c)^2}{2\alpha}\right)}{\exp(\frac{1}{2}\nabla U(\theta^c)\odot(1-\theta-\theta^c) - \frac{(1-\theta-\theta^c)^2}{2\alpha}) + \exp(\frac{1}{2}\nabla U(\theta^c)\odot(\theta-\theta^c) - \frac{(\theta-\theta^c)^2}{2\alpha})}$;
Sample $\mu \sim \text{Unif}(0,1)^d$;
$I \leftarrow \textbf{dim}(\mu \leq P(\theta|\theta^c))$;
$\theta' \leftarrow \textbf{flipdim}(I)$;
`// Optionally, do the MH step`
Compute $q(\theta'|\theta) = \prod_i q_i(\theta_i^{'}|\theta) = \prod_{i\in I} P_i(\theta|\theta^c)\prod_{i\notin I}(1-P_i(\theta|\theta^c))$;
Explore $\theta_c^{'} = \theta' + \frac{\alpha_0}{2}\nabla U(\theta')$;
Map $\theta_c^{'}$;
Construct $P(\theta'|\theta_c^{'}) = \frac{\exp\left(\frac{1}{2}\nabla U(\theta_c^{'})\odot(1-\theta'-\theta_c^{'}) - \frac{(1-\theta'-\theta_c^{'})^2}{2\alpha}\right)}{\exp(\frac{1}{2}\nabla U(\theta_c^{'})\odot(1-\theta'-\theta_c^{'}) - \frac{(1-\theta'-\theta_c^{'})^2}{2\alpha}) + \exp(\frac{1}{2}\nabla U(\theta_c^{'})\odot(\theta'-\theta_c^{'}) - \frac{(\theta'-\theta_c^{'})^2}{2\alpha})}$;
Compute $q(\theta|\theta') = \prod_i q_i(\theta_i|\theta) = \prod_{i\in I} P_i(\theta'|\theta_c^{'})\prod_{i\notin I}(1-P_i(\theta'|\theta_c^{'}))$;
Set $\theta \leftarrow \theta'$ with probability in Equation 10;
Output: samples$\{\theta_k\}$;

B Algorithm with Categorical Variables

When using one-hot vectors to represent categorical variables, our algorithm becomes

$$\text{Categorical}\left(\text{Softmax}\left(\frac{1}{2}\nabla U(\theta^c)_i^T(\theta_i^{'} - \theta_i^c) - \frac{\|\theta_i' - \theta_i^c\|_2^2}{2\alpha}\right)\right).$$

where $\theta_i^{'}$ is the one-hot vector, $\theta_c^{'}$ is the explored vector. Notably, if one-hot vectors are used, the gradient-explored vectors should extend on the coordinate with one. That is say, Equation 6 for $\theta = e_i$ should follow that

$$\theta^c = \text{argmin}_{y\in K}||y - \theta c|| = Proj_K(\theta^c)$$

Actually, if the variables are ordinal with clear ordering information, we can also use integer representation $\theta \in \{0, 1\cdots, K-1\}^d$ and follow Equation 5,6,7 to sample from discrete distributions.

C Necessity of the Projection

One direct method is without Equation 6. That is, the sample has explored its domain by gradient, then updates according to the extended continuous sample. However, it may meet some obstacles and be beyond the sample domain, which will result in a very low acceptance probability in practice. For example, in the RBM experiment, its acceptance probability is 0.001% while cDMALA's acceptance probability is 62% on average. We use a simple illustration in Fig 1.

D Proof of Theorem 1

$$\theta_c = \theta + \frac{\alpha_0}{2}\nabla U(\theta)$$
$$\theta' \sim \text{Categorical}\left(\text{Softmax}\left(\frac{1}{2}\nabla U(\theta_c)^T(\theta' - \theta_c) - \frac{||\theta' - \theta_c||_2^2}{2\alpha}\right)\right)$$

Expand the terms, we get

$$\begin{aligned}
&\frac{1}{2}\nabla U(\theta_c)^T(\theta' - \theta - \frac{\alpha_0}{2}\nabla U(\theta)) - \frac{\|\theta' - \theta - \frac{\alpha_0}{2}\nabla U(\theta)\|_2^2}{2\alpha} \\
\overset{*}{=}&\frac{1}{2}\nabla U(\theta)^T(\theta' - \theta) - \frac{\|\theta' - \theta\|_2^2}{2\alpha} + (\frac{1}{2}\nabla U(\theta_c) - \frac{1}{2}\nabla U(\theta) + \frac{\alpha_0}{2\alpha}\nabla U(\theta))^T(\theta' - \theta) \\
=&\frac{\alpha + \alpha_0}{2\alpha}\nabla U(\theta)^T(\theta' - \theta) - \frac{\|\theta' - \theta\|_2^2}{2\alpha} + \frac{\alpha_0}{4}\nabla U(\theta)^T\nabla^2 U(\theta)(\theta' - \theta) \\
=&\frac{\alpha + \alpha_0}{2\alpha}\nabla U(\theta)^T(\theta' - \theta) - \frac{\|\theta' - \theta\|_2^2}{2\alpha} + \frac{\alpha_0}{4}\nabla U(\theta)^T(\nabla U(\theta') - \nabla U(\theta)) \\
\overset{*}{=}&\frac{\alpha + \alpha_0}{2\alpha}\nabla U(\theta)^T(\theta' - \theta) - \frac{\|\theta' - \theta\|_2^2}{2\alpha} + \frac{\alpha_0}{4}\nabla U(\theta)^T(\nabla U(\theta'))
\end{aligned}$$

Here, $*$ means we drop some terms that won't affect the softmax result, i.e, not relevant to θ'. Recall that the target distribution is $\pi(\theta) = \exp(\theta^T W\theta + b^T\theta)/Z$. We have that $\nabla U(\theta) = 2W^T\theta + b^T\theta, \nabla^2 U(\theta) = 2W$. Since $\nabla^2 U(\theta)$ is a constant, we can rewrite the proposal distribution as the following:

$$\begin{aligned}
q_{\alpha,\alpha_0}(\theta'|\theta) &= \frac{\exp\left(\frac{\alpha+\alpha_0}{2\alpha}\nabla U(\theta)^T(\theta' - \theta) - \frac{\|\theta' - \theta\|_2^2}{2\alpha} + \frac{\alpha_0}{4}\nabla U(\theta)^T(\nabla U(\theta'))\right)}{\sum_x \exp\left(\frac{\alpha+\alpha_0}{2\alpha}\nabla U(\theta)^T(x - \theta) - \frac{\|x - \theta\|_2^2}{2\alpha} + \frac{\alpha_0}{4}\nabla U(\theta)^T(\nabla U(x))\right)} \\
&= \frac{\exp\left(\frac{\alpha+\alpha_0}{2\alpha}(U(\theta') - U(\theta)) - (\theta' - \theta)^T(\frac{1}{2\alpha}I + \frac{\alpha+\alpha_0}{2\alpha}W)(\theta' - \theta) + \frac{\alpha_0}{4}\nabla U(\theta)^T(\nabla U(\theta'))\right)}{\sum_x \exp\left(\frac{\alpha+\alpha_0}{2\alpha}(U(x) - U(\theta)) - (x - \theta)^T(\frac{1}{2\alpha}I + \frac{\alpha+\alpha_0}{2\alpha}W)(x - \theta) + \frac{\alpha_0}{4}\nabla U(\theta)^T(\nabla U(x))\right)}
\end{aligned}$$

Let $Z_{\alpha,\alpha_0}(\theta) = \sum_x \exp\left(\frac{\alpha+\alpha_0}{2\alpha}(U(x) - U(\theta)) - (x - \theta)^T(\frac{1}{2\alpha}I + \frac{\alpha+\alpha_0}{2\alpha}W)(x - \theta) + \frac{\alpha_0}{4}\nabla U(\theta)^T(\nabla U(x))\right)$, and $\pi_{\alpha,\alpha_0} = \frac{Z_{\alpha,\alpha_0}(\theta)\pi(\theta)}{\sum Z_{\alpha,\alpha_0}(x)\pi(x)}$, now we will show that q_{α,α_0} is reversible w.r.t π_{α,α_0}. To simplify the notation, we use $\tilde{\alpha}$ to take place (α, α_0).

We have that

$$\begin{aligned}
\pi_{\tilde{\alpha}}q_{\tilde{\alpha}}(\theta'|\theta) &= \frac{Z_{\tilde{\alpha}}(\theta)\pi(\theta)}{\sum Z_{\tilde{\alpha}}(x)\pi(x)} \cdot \frac{\exp\left(\frac{\alpha+\alpha_0}{2\alpha}(U(\theta') - U(\theta)) - (\theta' - \theta)^T(\frac{1}{2\alpha}I + \frac{\alpha+\alpha_0}{2\alpha}W)(\theta' - \theta) + \frac{\alpha_0}{4}\nabla U(\theta)^T(\nabla U(\theta'))\right)}{Z_{\tilde{\alpha}}} \\
&= \frac{\exp\left(\frac{\alpha+\alpha_0}{2\alpha}U(\theta') + \frac{\alpha-\alpha_0}{2\alpha}U(\theta)) - (\theta' - \theta)^T(\frac{1}{2\alpha}I + \frac{\alpha+\alpha_0}{2\alpha}W)(\theta' - \theta) + \frac{\alpha_0}{4}\nabla U(\theta)^T(\nabla U(\theta'))\right)}{Z \cdot \sum_x Z_{\tilde{\alpha}}(x)\pi(x)}
\end{aligned}$$

Only and only if $\alpha_0 = o(\alpha)$, it is clear that this expression is symmetric in θ and θ'. Therefore $q_{\tilde{\alpha}}$ is reversible and the stationary distribution is $\pi_{\tilde{\alpha}}$.

Now we will prove that $\pi_{\tilde{\alpha}}$ converges weakly to π as $\alpha, \alpha_0 \to 0$ ($\alpha_0 = o(\alpha)$). Notice that for any θ,

$$\begin{aligned}
Z_{\tilde{\alpha}}(\theta) &= \sum_x \exp\left(\frac{\alpha + \alpha_0}{2\alpha}(U(x) - U(\theta)) - (x - \theta)^T(\frac{1}{2\alpha}I + \frac{\alpha + \alpha_0}{2\alpha}W)(x - \theta) + \frac{\alpha_0}{4}\nabla U(\theta)^T(\nabla U(x))\right) \\
&\overset{\tilde{\alpha}\downarrow 0}{=} \sum_x \exp\left(\frac{1}{2}(U(x) - U(\theta))\right)\delta_\theta(x) \\
&= 1.
\end{aligned}$$

where $\delta_\theta(x)$ is a Dirac delta. Due to $Z_{\tilde{\alpha}}(\theta)$ is a continuous function when θ is fixed on $\mathcal{A} = \{\alpha > 0, \alpha_0 \geq 0\}$, so the limitation is well-defined. It follows that $\pi_{\tilde{\alpha}}$ converges pointwisely to $\pi(\theta)$. By Scheffe's Lemma, we attain $\pi_{\tilde{\alpha}}$ converges weakly to π.

Convergence Rate w.r.t Stepsize Let us consider the convergence rate in terms of L_1-norm.

$$\|\pi_{\tilde{\alpha}} - \pi\|_1 = \sum_\theta \left| \frac{Z_{\tilde{\alpha}}(\theta)\pi(\theta)}{\sum_x Z_{\tilde{\alpha}}(x)\pi(x)} - \pi(\theta) \right|.$$

We write out each absolute value term

$$\begin{aligned}
\left| \frac{Z_{\tilde{\alpha}}(\theta)\pi(\theta)}{\sum_x Z_{\tilde{\alpha}}(x)\pi(x)} - \pi(\theta) \right| &= \pi(\theta) \left| \frac{Z_{\tilde{\alpha}}(\theta)}{\sum_x Z_{\tilde{\alpha}}(x)\pi(x)} - 1 \right| \\
&= \pi(\theta) \left| \frac{1 + \sum_{x \neq \theta} \exp(\frac{\alpha+\alpha_0}{2\alpha}(U(x) - U(\theta)) - (x-\theta)^T(\frac{1}{2\alpha}I + \frac{\alpha+\alpha_0}{2\alpha}W)(x-\theta) + \frac{\alpha_0}{4}\nabla U(\theta)^T(\nabla U(x) - \nabla U(\theta)))}{1 + \sum_y \frac{1}{Z}\exp(U(y)) \sum_{x \neq y} \exp(\frac{\alpha+\alpha_0}{2\alpha}(U(x) - U(\theta)) - (x-\theta)^T(\frac{1}{2\alpha}I + \frac{\alpha+\alpha_0}{2\alpha}W)(x-\theta) + \frac{\alpha_0}{4}\nabla U(\theta)^T(\nabla U(x) - \nabla U(\theta)))} - 1 \right|
\end{aligned}$$

where $\tilde{\alpha} = \frac{\alpha+\alpha_0}{2\alpha}$. Since $\lambda_{min}(W)\|x\|^2 \leq x^T W x, \forall x$, it follows that

$$(x-\theta)(\frac{1}{2\alpha}I + \frac{\alpha+\alpha_0}{2\alpha}W)(x-\theta) \geq \|x-\theta\|^2(\frac{1 + (\alpha+\alpha_0)\lambda_{min}}{2\alpha})$$

We also notice that $\min_{x \neq \theta} \|x-\theta\|^2 = 1$, and

$$\|\nabla U(\theta)^T(\nabla U(x) - \nabla U(\theta))\|_2 = \|(2W^T\theta + b)^T(2W^T(x-\theta)\|_2 \leq C\|x-\theta\|_\infty = CD$$

where $D = \max_{\theta,\theta' \in \Theta} \|\theta' - \theta\|_\infty$. Thus when $\frac{Z_{\tilde{\alpha}(\theta)}}{\sum_x Z_{\tilde{\alpha}}(x)\pi(x)} - 1 > 0$, we get

$$\begin{aligned}
\left| \frac{Z_{\tilde{\alpha}}(\theta)\pi(\theta)}{\sum_x Z_{\tilde{\alpha}}(x)\pi(x)} - \pi(\theta) \right| &= \pi(\theta) \left| \frac{Z_{\tilde{\alpha}}(\theta)}{\sum_x Z_{\tilde{\alpha}}(x)\pi(x)} - 1 \right| \\
&= \pi(\theta) \left(\frac{1 + \sum_{x \neq \theta} \exp(\frac{\alpha+\alpha_0}{2\alpha}(U(x) - U(\theta)) - (x-\theta)^T(\frac{1}{2\alpha}I + \frac{\alpha+\alpha_0}{2\alpha}W)(x-\theta) + \frac{\alpha_0}{4}\nabla U(\theta)^T(\nabla U(x) - \nabla U(\theta)))}{1 + \sum_y \frac{1}{Z}\exp(U(y)) \sum_{x \neq y} \exp(\frac{\alpha+\alpha_0}{2\alpha}(U(x) - U(y)) - (x-y)^T(\frac{1}{2\alpha}I + \frac{\alpha+\alpha_0}{2\alpha}W)(x-y) + \frac{\alpha_0}{4}\nabla U(y)^T(\nabla U(x) - \nabla U(y)))} - 1 \right) \\
&\leq \pi(\theta) \left(1 + \sum_{x \neq \theta} \exp(\frac{\alpha+\alpha_0}{2\alpha}(U(x) - U(\theta)) - (\frac{1}{2\alpha}I + \frac{\alpha+\alpha_0}{2\alpha}W)\|x-\theta\|^2 + \frac{\alpha_0}{4}\nabla U(\theta)^T(\nabla U(x) - \nabla U(\theta))) - 1 \right) \\
&\leq \pi(\theta) \left(c \sum_x \exp(U(x)) \right) \cdot \exp\left(-\frac{1 + (\alpha+\alpha_0)\lambda_{min}}{2\alpha} \right) \\
&= c \cdot \pi(\theta) Z \cdot \exp(-\frac{1 + (\alpha+\alpha_0)\lambda_{min}}{2\alpha})
\end{aligned}$$

where $c = \exp(\frac{\alpha_0}{4}CD)$ is a constant. Similarly, when $\frac{Z_{\tilde{\alpha}(\theta)}}{\sum_x Z_{\tilde{\alpha}}(x)\pi(x)} - 1 < 0$, we have,

$$
\begin{aligned}
\left|\frac{Z_{\tilde{\alpha}}(\theta)\pi(\theta)}{\sum_x Z_{\tilde{\alpha}}(x)\pi(x)} - \pi(\theta)\right| &= \pi(\theta)\left(1 - \frac{1 + \sum_{x\neq\theta}\exp(\frac{\alpha+\alpha_0}{2\alpha}(U(x)-U(\theta)) - (x-\theta)^T(\frac{1}{2\alpha}I + \frac{\alpha+\alpha_0}{2\alpha}W)(x-\theta) + \frac{\alpha_0}{4}\nabla U(\theta)^T(\nabla U(x) - \nabla U(\theta)))}{1 + \sum_y \frac{1}{Z}\exp(U(y))\sum_{x\neq y}\exp(\frac{\alpha+\alpha_0}{2\alpha}(U(x)-U(y)) - (x-y)^T(\frac{1}{2\alpha}I + \frac{\alpha+\alpha_0}{2\alpha}W)(x-y) + \frac{\alpha_0}{4}\nabla U(y)^T(\nabla U(x) - \nabla U(y)))}\right) \\
&\leq \pi(\theta)\left(1 - \frac{1}{1 + \sum_y \frac{1}{Z}\exp(U(y))\sum_{x\neq y} c\exp(\frac{\alpha+\alpha_0}{2\alpha}(U(x)-U(\theta)) - \frac{1+(\alpha+\alpha_0)\lambda_{min}}{2\alpha})}\right) \\
&= \pi(\theta)\left(\frac{\sum_y \frac{1}{Z}\exp(U(y))\sum_{x\neq y} c\exp(\frac{\alpha+\alpha_0}{2\alpha}(U(x)-U(\theta)) - \frac{1+(\alpha+\alpha_0)\lambda_{min}}{2\alpha})}{1 + \sum_y \frac{1}{Z}\exp(U(y))\sum_{x\neq y} c\exp(\frac{\alpha+\alpha_0}{2\alpha}(U(x)-U(\theta)) - \frac{1+(\alpha+\alpha_0)\lambda_{min}}{2\alpha})}\right) \\
&\leq \pi(\theta)\left(\sum_y \frac{1}{Z}\exp(U(y))\sum_{x\neq y} c\exp(\frac{\alpha+\alpha_0}{2\alpha}(U(x)-U(\theta)))\right)\cdot\exp\left(-\frac{1+(\alpha+\alpha_0)\lambda_{min}}{2\alpha}\right) \\
&\leq \pi(\theta)\left(c\sum_x \exp(U(x))\right)\cdot\exp\left(-\frac{1+(\alpha+\alpha_0)\lambda_{min}}{2\alpha}\right) \\
&= c\cdot\pi(\theta)Z\cdot\exp(-\frac{1+(\alpha+\alpha_0)\lambda_{min}}{2\alpha})
\end{aligned}
$$

Therefore, the difference between $\pi_{\tilde{\alpha}}$ and π can be bounded as follows.

$$
\|\pi_{\tilde{\alpha}} - \pi\| \leq c\sum_\theta \pi(\theta)Z\cdot\exp\left(-\frac{1+(\alpha+\alpha_0)\lambda_{min}}{2\alpha}\right) = cZ\cdot\exp\left(-\frac{1+(\alpha+\alpha_0)\lambda_{min}}{2\alpha}\right).
$$

E Proof of Theorem 2

Proof We use a log-quadratic distribution that is close to π as an intermediate term to bound the bias of cDULA. Recall that π is the target distribution, $\pi^{'}$ is the log-quadratic distribution that is close to π and $\pi_{\tilde{\alpha}}$ is the stationary distribution of cDULA. We let $\pi^{'}_{\tilde{\alpha}}$ be the stationary distributions of cDULA targetting $\pi^{'}$, then by triangle inequality,

$$
\|\pi_{\tilde{\alpha}} - \pi\|_1 \leq \|\pi_{\tilde{\alpha}} - \pi'_{\tilde{\alpha}}\|_1 + \|\pi'_{\tilde{\alpha}} - \pi'\|_1 + \|\pi' - \pi\|_1.
$$

Bound of $\|\pi' - \pi\|_1$. Let the energy function of π' be $V(\theta) = \theta^T W\theta + b^T\theta$. Since Θ is a discrete space, there exists a bounded subset $\Omega \in \mathrm{R}^d$ such that Θ is a subset of Ω. By Poincar*é* inequality, we get

$$
|U(\theta) - V(\theta)| \leq C_1\cdot\|\nabla U(\theta) - (W\theta + b)\|_1 \leq C_1\epsilon, \qquad \forall\theta\in\Theta
$$

where the constant C_1 depends on Ω.

Recall that $\pi(\theta) = \frac{\exp(U(\theta))}{Z}$. Let $\pi'(\theta) = \frac{\exp(V(\theta))}{Z'}$ where Z' is the normalizing constant to make π' a distribution. Then

$$
\|\pi - \pi'\|_1 = \sum_{\theta\in\Theta}\left|\frac{\exp(U(\theta))}{Z} - \frac{\exp(V(\theta))}{Z'}\right|. \tag{16}
$$

We notice that $\forall\theta$,

$$
\frac{\exp(U(\theta))}{\exp(V(\theta))} = \exp(U(\theta) - V(\theta)) \leq \exp(|U(\theta) - V(\theta)|) \leq \exp(C\epsilon),
$$

and similarly

$$
\frac{\exp(V(\theta))}{\exp(U(\theta))} \leq \exp(C\epsilon).
$$

Therefore we have $\forall\theta$,

$$\begin{aligned}\left|\frac{\exp(U(\theta))}{Z}-\frac{\exp(V(\theta))}{Z'}\right| &= \left|\frac{\exp(U(\theta))\cdot Z'-\exp(V(\theta))\cdot Z}{Z\cdot Z'}\right| \\ &\leq (\exp(2C\epsilon)-1)\min\left\{\frac{\exp(U(\theta))}{Z},\frac{\exp(V(\theta))}{Z'}\right\} \\ &\leq (\exp(2C\epsilon)-1)\left(\frac{\exp(U(\theta))}{Z}+\frac{\exp(V(\theta))}{Z'}\right). \end{aligned} \tag{17}$$

Plugging the above in Equation equation 16, we obtain

$$\|\pi-\pi'\|_1 \leq 2\left(\exp(2C\epsilon)-1\right).$$

Bound of $\|\pi'_{\tilde{\alpha}}-\pi'\|_1$. By Theorem 1, we know

$$\|\pi'_{\tilde{\alpha}}-\pi'\|_1 \leq cZ'\cdot\exp\left(-\frac{1+(\alpha+\alpha_0)\lambda_{min}}{2\alpha}\right).$$

Bound of $\|\pi_{\tilde{\alpha}}-\pi'_{\tilde{\alpha}}\|_1$. Now we will bound $\|\pi_{\tilde{\alpha}}-\pi'_{\tilde{\alpha}}\|_1$ by perturbation bounds. Let $T_{\tilde{\alpha}}$, $T'_{\tilde{\alpha}}$ be the transition matrices of GEDLS on π and π' respectively. We consider the difference between these two matrices.

$$\left\|T_{\tilde{\alpha}}-T'_{\tilde{\alpha}}\right\|_\infty = \max_\theta\sum_{\theta'}\left|\frac{\exp\left(\frac{\alpha+\alpha_0}{2\alpha}\nabla U(\theta)^T(\theta'-\theta)-\frac{\|\theta'-\theta\|_2^2}{2\alpha}+\frac{\alpha_0}{4}\nabla U(\theta)^T(\nabla U(\theta'))\right)}{Z_\alpha(\theta)}-\frac{\exp\left(\frac{\alpha+\alpha_0}{2\alpha}\nabla V(\theta)^T(\theta'-\theta)-\frac{\|\theta'-\theta\|_2^2}{2\alpha}+\frac{\alpha_0}{4}\nabla V(\theta)^T(\nabla V(\theta'\right.}{Z'_\alpha(\theta)}\right.$$

where

$$\begin{aligned} Z_\alpha(\theta) &= \sum_x \exp\left(\frac{\alpha+\alpha_0}{2\alpha}\nabla U(\theta)^T(\theta'-\theta)-\frac{\|\theta'-\theta\|_2^2}{2\alpha}+\frac{\alpha_0}{4}\nabla U(\theta)^T(\nabla U(\theta'))\right), \\ Z'_\alpha(\theta) &= \sum_x \exp\left(\frac{\alpha+\alpha_0}{2\alpha}\nabla V(\theta)^T(\theta'-\theta)-\frac{\|\theta'-\theta\|_2^2}{2\alpha}+\frac{\alpha_0}{4}\nabla V(\theta)^T(\nabla V(\theta'))\right). \end{aligned}$$

We denote $D=\max_{\theta,\theta'\in\Theta}\|\theta'-\theta\|_\infty$. By the assumption $\|\nabla U(\theta)-\nabla V(\theta)\|_1\leq\epsilon$, we get

$$\begin{aligned}\exp\left(\nabla^T U(\theta)\nabla U(\theta')-\nabla^T V(\theta)\nabla V(\theta')\right) &\leq \exp\left(\left|\nabla^T U(\theta)\nabla U(\theta')-\nabla^T V(\theta)\nabla V(\theta')\right|\right) \\ &= \exp\left(\left|\nabla^T U(\theta)\nabla U(\theta')-\nabla^T U(\theta)V(\theta')+\nabla^T U(\theta)V(\theta')-\nabla^T V(\theta)\nabla V(\theta')\right|\right) \\ &\leq \exp\left(\left|\nabla^T U(\theta)\nabla U(\theta')-\nabla^T U(\theta)V(\theta')\right|+\left|\nabla^T U(\theta)V(\theta')-\nabla^T V(\theta)\nabla V(\theta')\right|\right) \\ &\leq \exp(M\epsilon)\end{aligned}$$

where $M=2\max_\theta\{|\nabla V(\theta)|\}$. It gets from

$$|\nabla U(\theta)| = |\nabla U(\theta)-\nabla V(\theta)+\nabla V(\theta)| \leq C_1\epsilon+|\nabla V(\theta)|.$$

Similar to Equation equation 17, we have

$$\left|\frac{\exp\left(\frac{\alpha+\alpha_0}{2\alpha}\nabla U(\theta)^T(\theta'-\theta)-\frac{\|\theta'-\theta\|_2^2}{2\alpha}+\frac{\alpha_0}{4}\nabla U(\theta)^T(\nabla U(\theta'))\right)}{Z_\alpha(\theta)}-\frac{\exp\left(\frac{\alpha+\alpha_0}{2\alpha}\nabla V(\theta)^T(\theta'-\theta)-\frac{\|\theta'-\theta\|_2^2}{2\alpha}+\frac{\alpha_0}{4}\nabla V(\theta)^T(\nabla V(\theta'))\right)}{Z'_\alpha(\theta)}\right|$$

$$\leq (\exp(N\epsilon)-1)$$

$$\cdot \min\left\{\frac{\exp\left(\frac{\alpha+\alpha_0}{2\alpha}\nabla U(\theta)^T(\theta'-\theta)-\frac{\|\theta'-\theta\|_2^2}{2\alpha}+\frac{\alpha_0}{4}\nabla U(\theta)^T(\nabla U(\theta'))\right)}{Z_\alpha(\theta)},\frac{\exp\left(\frac{\alpha+\alpha_0}{2\alpha}\nabla V(\theta)^T(\theta'-\theta)-\frac{\|\theta'-\theta\|_2^2}{2\alpha}+\frac{\alpha_0}{4}\nabla V(\theta)^T(\nabla V(\theta'))\right)}{Z'_\alpha(\theta)}\right\}$$

$$\leq (\exp(N\epsilon)-1)$$

$$\cdot\left(\frac{\exp\left(\frac{\alpha+\alpha_0}{2\alpha}\nabla U(\theta)^T(\theta'-\theta)-\frac{\|\theta'-\theta\|_2^2}{2\alpha}+\frac{\alpha_0}{4}\nabla U(\theta)^T(\nabla U(\theta'))\right)}{Z_\alpha(\theta)}+\frac{\exp\left(\frac{\alpha+\alpha_0}{2\alpha}\nabla V(\theta)^T(\theta'-\theta)-\frac{\|\theta'-\theta\|_2^2}{2\alpha}+\frac{\alpha_0}{4}\nabla V(\theta)^T(\nabla V(\theta'))\right)}{Z'_\alpha(\theta)}\right).$$

where $N=\frac{\alpha+\alpha_0}{\alpha}D+\frac{M\alpha_0}{2}$. Now we substitute it to $\left\|T_\alpha-\tilde{T}_\alpha\right\|_\infty$,

$$\left\|T-\tilde{T}\right\|_\infty \leq 2\left(\exp(N\epsilon)-1\right).$$

By the perturbation bound in Schweitzer (1968),

$$\|\pi_{\tilde{\alpha}}-\pi'_{\tilde{\alpha}}\|_1 \leq C_2\cdot\|T_{\tilde{\alpha}}-T'_{\tilde{\alpha}}\|_\infty = 2C_2\left(\exp(N\epsilon)-1\right)$$

where C_2 is a constant depending on π' and $\tilde{\alpha}$. Please note that it is also possible to use other perturbation bounds (Cho & Meyer, 2001).

Combining these three bounds, we get

$$\begin{aligned}\|\pi_\alpha-\pi\|_1 &\leq \|\pi_\alpha-\pi'_\alpha\|_1+\|\pi'_\alpha-\pi'\|_1+\|\pi'-\pi\|_1\\ &\leq 2C_2\left(\exp(N\epsilon)-1\right)+cZ'\cdot\exp\left(-\frac{1+(\alpha+\alpha_0)\lambda_{min}}{2\alpha}\right)+2\left(\exp(2C_1\epsilon)-1\right).\end{aligned}$$

We define $c_1 := 2\max(2,2C_2)$ and $c_2 := \max(2C_1, N)$, then we reach the the final result

$$\begin{aligned}\|\pi_\alpha-\pi\|_1 &\leq \|\pi_\alpha-\pi'_\alpha\|_1+\|\pi'_\alpha-\pi'\|_1+\|\pi'-\pi\|_1\\ &\leq 2c_1\cdot\left(\exp(c_2\epsilon)-1\right)+Z'\cdot\exp\left(-\frac{1+(\alpha+\alpha_0)\lambda_{min}}{2\alpha}\right).\end{aligned}$$

F Proof of Lemma 1 and Theorem 3

Proof. Recalling cDLS, we consider the proposal kernel as

$$q_{\tilde{\alpha}}(\theta'|\theta)\propto\exp\left\{\tilde{\alpha}\nabla U(\theta)^T(\theta'-\theta)-\frac{\|\theta-\theta'\|^2}{2\alpha}+\frac{1}{2}(\nabla U(\theta_c)-\nabla U(\theta))^T(\theta'-\theta)\right\}$$

and consider the transition kernel as

$$p(\theta'|\theta)=\min\left\{\frac{\pi(\theta')q_{\tilde{\alpha}}(\theta|\theta')}{\pi(\theta)q_{\tilde{\alpha}}(\theta'|\theta)},1\right\}q_{\tilde{\alpha}}(\theta'|\theta)+(1-L(\theta))\delta_\theta(\theta')$$

where $\delta_\theta(\theta')$ is the Kronecker delta function and $L(\theta)$ is the total acceptable probability from the point θ with

$$L(\theta)=\sum_{\theta'\in\Theta}\min\left\{\frac{\pi(\theta')q_{\tilde{\alpha}}(\theta|\theta')}{\pi(\theta)q_{\tilde{\alpha}}(\theta'|\theta)},1\right\}q_{\tilde{\alpha}}(\theta'|\theta).$$

We also define

$$Z_{\tilde{\alpha}}(\theta) = \sum_{x\in\Theta} \exp\left\{\tilde{\alpha}\nabla U(\theta)^T(x-\theta) - \frac{\|x-\theta'\|^2}{2\alpha} + \frac{1}{2}(\nabla U(\theta_c) - \nabla U(\theta))^T(x-\theta)\right\}$$

which is the normalizing constant for the proposal kernel.

Consider the term,

$$\tilde{\alpha}\nabla U(\theta)^T(\theta'-\theta) - \frac{1}{2\alpha}\|\theta-\theta'\|^2 = \tilde{\alpha}(-U(\theta)+U(\theta')) - \frac{1}{2}(\theta-\theta')^T\left(\tilde{\alpha}\int_0^1 \nabla^2 U[(1-s)\theta + s\theta']\mathrm{d}s + \frac{1}{\alpha}I\right)(\theta-\theta').$$

From Assumption 1, we have

$$\tilde{\alpha}\int_0^1 \nabla^2 U[(1-s)\theta + s\theta']\mathrm{d}s(\theta-\theta') + \frac{1}{\alpha}I \geq (\frac{1}{\alpha} - \tilde{\alpha}L)I.$$

Since $\alpha + \alpha_0 < \frac{2}{L}$, the matrix $(\frac{1}{\alpha} - \tilde{\alpha}L)I$ is positive define. We denote that

$$\begin{aligned}
p(\theta'|\theta) &= \min\left\{\frac{\pi(\theta')q_{\tilde{\alpha}}(\theta|\theta')}{\pi(\theta)q_{\tilde{\alpha}}(\theta'|\theta)}, 1\right\} q_{\tilde{\alpha}}(\theta'|\theta) + (1-L(\theta))\delta_\theta(\theta') \\
&\geq \min\left\{\frac{\pi(\theta')q_{\tilde{\alpha}}(\theta|\theta')}{\pi(\theta)q_{\tilde{\alpha}}(\theta'|\theta)}, 1\right\} q_{\tilde{\alpha}}(\theta'|\theta) \\
&\geq \min\left\{\frac{Z_{\tilde{\alpha}}(\theta)\exp(\frac{\alpha_0}{4}L\|\nabla U(\theta')\|)}{Z_{\tilde{\alpha}}(\theta')\exp(\frac{\alpha_0}{4}LD\|\nabla U(\theta)\|)}, 1\right\} q_{\tilde{\alpha}}(\theta'|\theta)
\end{aligned}$$

where $Z_{\tilde{\alpha}}$ could also be rewritten as

$$\begin{aligned}
Z_{\tilde{\alpha}}(\theta) &= \sum_{x\in\Theta} \exp\left\{\tilde{\alpha}\nabla U(\theta)^T(x-\theta) - \frac{\|x-\theta\|^2}{2\alpha} + \frac{1}{2}(\nabla U(\theta_c) - \nabla U(\theta))^T(x-\theta)\right\} \\
&= \sum_{x\in\Theta} \exp\left\{\tilde{\alpha}(-U(\theta)+U(x)) - \frac{1}{2}(\theta-x)^T\left(\tilde{\alpha}\int_0^1 \nabla^2 U[(1-s)\theta + sx]\mathrm{d}s + \frac{1}{\alpha}I\right)(\theta-x) + \frac{1}{2}(\nabla U(\theta_c) - \nabla U(\theta))^T(x-\theta)\right\}
\end{aligned}$$

This can be seen as

$$\pi(\theta)q_{\tilde{\alpha}}(\theta'|\theta) = \frac{1}{ZZ_{\tilde{\alpha}}(\theta)}\exp\left\{\tilde{\alpha}(U(\theta)+U(\theta')) - \frac{1}{2}(\theta-\theta')^T\left(\tilde{\alpha}\int_0^1 \nabla^2 U[(1-s)\theta + s\theta']\mathrm{d}s + \frac{1}{\alpha}I\right)(\theta-\theta') + \frac{1}{2}(\nabla U(\theta_c) - \nabla U(\theta))^T(\theta'-\theta)\right\}.$$

Since Assumption 2 holds true in this setting, we have an $m > 0$ such that for any $\theta \in conv(\Theta)$

$$-\nabla^2 U(\theta) \geq mI.$$

From this, one notes that

$$\exp\left(-\tilde{\alpha}U(\theta) - \frac{1}{2}(\frac{1}{\alpha} - \tilde{\alpha}m)D^2 + \frac{\alpha_0}{4}L\|\nabla U(\theta)\|\right)\sum_{x\in\Theta}\exp(\tilde{\alpha}U(x)) \leq Z_{\tilde{\alpha}}(\theta) \leq \exp(-\tilde{\alpha}U(\theta) + \frac{\alpha_0}{4}LD\|\nabla U(\theta)\|)\sum_{x\in\Theta}\exp(\tilde{\alpha}U(x))$$

where the right-hand side follows from the fact that $\alpha + \alpha_0 < \frac{2}{L}$. Therefore,

$$\frac{Z_{\tilde{\alpha}}(\theta)}{Z_{\tilde{\alpha}}(\theta')} \geq \frac{\exp\{\tilde{\alpha}(-U(\theta)+U(\theta'))\}\cdot\exp\{\frac{\alpha_0}{4}L\|\nabla U(\theta)\|\}}{\exp\{\frac{1}{2}(\frac{1}{\alpha} - \tilde{\alpha}m)D^2\}\cdot\exp\{\frac{\alpha_0}{4}LD\|\nabla U(\theta')\|\}}$$

Also note that

$$\begin{aligned}
q_{\tilde{\alpha}}(\theta'|\theta) =& \frac{\exp\left\{\tilde{\alpha}(-U(\theta)+U(\theta')) - \frac{1}{2}(\theta-\theta')^T\left(\tilde{\alpha}\int_0^1 \nabla^2 U[(1-s)\theta + s\theta']\mathrm{d}s + \frac{1}{\alpha}I\right)(\theta-\theta') + \frac{1}{2}(\nabla U(\theta_c) - \nabla U(\theta))^T(\theta'-\theta)\right\}}{\sum_{x\in\Theta}\exp\left\{\tilde{\alpha}(-U(\theta)+U(x)) - \frac{1}{2}(\theta-x)^T\left(\tilde{\alpha}\int_0^1 \nabla^2 U[(1-s)\theta + sx]\mathrm{d}s + \frac{1}{\alpha}I\right)(\theta-x) + \frac{1}{2}(\nabla U(\theta_c) - \nabla U(\theta))^T(x-\theta)\right\}} \\
\geq& \frac{\exp\left\{\tilde{\alpha}\nabla U(\theta)^T(\theta'-\theta) - \frac{\|\theta-\theta'\|^2}{2\alpha} + \frac{1}{2}(\nabla U(\theta_c) - \nabla U(\theta))^T(\theta'-\theta)\right\}}{\exp(-\tilde{\alpha}U(\theta) + \frac{1}{2}LD^2)\sum_{x\in\Theta}\exp(\tilde{\alpha}U(x))}
\end{aligned}$$

and

$$
\begin{aligned}
-\tilde{\alpha}\nabla U(\theta)^T(\theta'-\theta)+\frac{\|\theta-\theta'\|^2}{2\alpha}-\frac{1}{2}(\nabla U(\theta_c)-\nabla U(\theta))^T(\theta'-\theta) =& \tilde{\alpha}(-\nabla U(\theta)+\nabla U(a))^T(\theta'-\theta)-\nabla U(a)^T(\theta'-\theta) \\
&+\frac{\|\theta-\theta'\|^2}{2\alpha}-\frac{1}{2}(\nabla U(\theta_c)-\nabla U(\theta))^T(\theta'-\theta) \\
&\leq \tilde{\alpha}\,\|\nabla U(a)-\nabla U(\theta)\|\,D+\tilde{\alpha}\,\|\nabla U(a)\|\,D+\frac{1}{2\alpha}D^2 \\
&\leq (\tilde{\alpha}L+\frac{1}{2\alpha})D^2+\tilde{\alpha}\,\|\nabla U(a)\|\,D+\frac{1}{2}LD^2
\end{aligned}
$$

Combining the inequities, we get

$$
p(\theta'|\theta)\geq \epsilon_{\tilde{\alpha}}\frac{\exp\{\tilde{\alpha}U(\theta')\}}{\sum_{\theta'}\exp\{\tilde{\alpha}U(\theta')\}}
$$

where

$$
\epsilon_{\tilde{\alpha}}=\exp\left\{-(\frac{1}{\alpha}+\tilde{\alpha}L+L-\frac{\tilde{\alpha}m}{2})D^2-\tilde{\alpha}\,\|\nabla U(a)\|\,D\right\}
$$

To simplify, it also could be bound with $\alpha+\alpha_0<\frac{2}{L}$,

$$
\begin{aligned}
\epsilon_{\tilde{\alpha}} =& \exp\left\{-(\frac{1}{\alpha}+\tilde{\alpha}L+L-\frac{\tilde{\alpha}m}{2})D^2-\tilde{\alpha}\,\|\nabla U(a)\|\,D\right\} \\
>& \exp\left\{-(\frac{1}{\alpha}+\frac{2}{2\alpha\cdot L}L+L-\frac{\tilde{\alpha}m}{2})D^2-\tilde{\alpha}\,\|\nabla U(a)\|\,D\right\} \\
=& \exp\left\{-(\frac{2}{\alpha}+L-\frac{\tilde{\alpha}m}{2})D^2-\tilde{\alpha}\,\|\nabla U(a)\|\,D\right\}.
\end{aligned}
$$

As for Theorem 3, the proof directly follows from Lemma 1 and Cor.5 in Jones (2004).

G Additional Experiments Results and Setting Details

G.1 Ising Model For Theory Verification

To verify Theorem 1, we use a 2 by 2 Ising model $U(\theta)=a\theta^TW\theta+b\theta$ where W is the binary adjacency matrix and $a=0.1, b=0.2$. To verify Theorem 2, we set $a=1$ and $b=0.1$.

G.2 Sampling from Synthetic Distribution

We use the experiment setting of Pynadath et al. (2024). We need to define the space between the modes, the total number of modes, and the variance of each mode σ. For convenience, we take the same mode number in Pynadath et al. (2024), the space between modes as 15, and the variance for each mode σ^2 as .3. Based on this, we can calculate the maximum value for each coordinate as follows:

$$
MaxVal=(\sqrt{NumModes}+1)*SpaceBetweenModes.
$$

The mode's center could be calculated as follows:

$$
\mu_{i,j}=\frac{MaxVal}{\sqrt{NumModes}+2}(i+1)
$$

$$
\mu_{i,j}=\frac{MaxVal}{\sqrt{NumModes}+2}(j+1)
$$

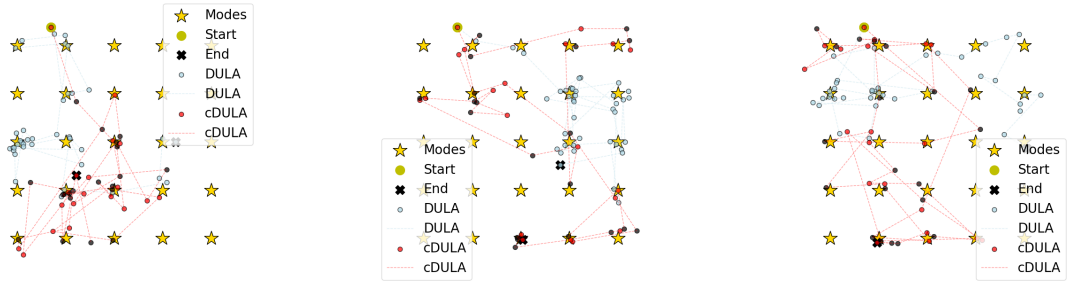


Figure 7: Sampling Trajectory with different random seeds.

G.3 Sampling from Ising Models

We adopt the experiment setting of Zhang et al. (2022). The energy of the Ising model is defined as,

$$\log(p(x)) = ax^T Jx + b^T x$$

where a controls the connectivity strength and J is an adjacency matrix whose elements are either 0 or 1. If $J = 0$, then the model becomes a factorized Bernoulli distribution. In our experiments, J is the adjacency matrix of a 2D lattice graph. cDULA and cDMALA use the same stepsize tuple (0.4,0.2). We run all methods for the same number of iterations. The results are shown in Figure 4 and **??**. We can clearly observe that our methods outperform other baselines on all connectivity strengths. These three coupling strengths respectively correspond to scenarios within the Ising model dominated by a strong external field, strong internal interactions, and the absence of an external field.

G.4 Sampling from RBMs

RBM Overview We will give a brief overview of the Blocks-Gibbs sampler used to represent the ground truth of the RBM distribution. For a more in-depth explanation, see Grathwohl et al. (2021). Given the hidden units h and the sample x, we define the RBM distribution as follows:

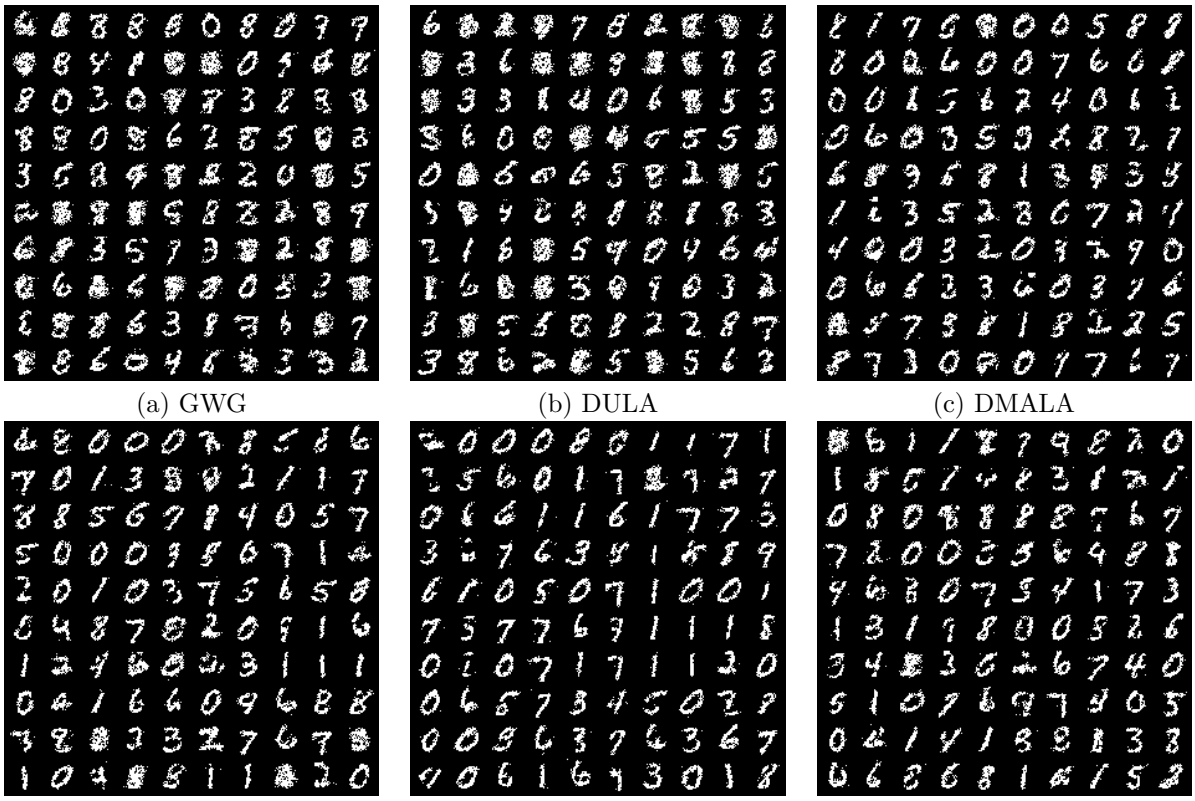
$$\log(p(x, h)) = h^T Wx + b^T x + c^T - \log(Z) \tag{18}$$

where Z is the normalizing constant for the distribution. The sample x is represented by the visible layer with units corresponding to the sample space dimension, and h represents the model capacity. It can be shown that the marginal distributions are as follows:

$$\begin{aligned} p(x|h) &= \text{Bernoulli}(Wx + c) \\ p(h|x) &= \text{Bernoulli}(W^T h + b) \\ \log(p(x)) &= \sum_i \text{Softplus}(Wx + a)_i + b^T x - \log(z). \end{aligned}$$

The Block-Gibbs sampler updates x and h alternatively, allowing for many of the coordinates to get changes at the same time, due to utilizing the specific structure of the RBM model.

Experiment Setup Similar to the experimental setup of Zhang et al. (2022); Pynadath et al. (2024), we use RBM models with 500 hidden units and 784 visible units. We adopt the same training protocol, except that we fine-tune the learning rate of Omniglot. From simpler MNIST-like data sets, we only train for a single pass through the dataset, while for more complicated datasets, we train for 3000 iterations to better represent the character and generate more realistic samplers. We include the generated images in Figure 11 to demonstrate that these models have learned the dataset reasonably well.



(a) GWG (b) DULA (c) DMALA

(d) ACS (e) cDULA (f) cDMALA

Figure 8: Generated samples from an RBM trained on MNIST. Our methods generate images that are closer to ground-truth samples.

Sampler Configuration For GWG, we use the same setting as Grathwohl et al. (2021). For DLP, we set the stepsize to 0.2 and 0.1 alternatively with MH correction and without MH correction. For cDLS, we use $\alpha = 0.2, \alpha_0 = 0.04$.

Sampling Speed While the run time can vary depending on the specific implementation of a given sampling algorithm, we illustrate the efficiency of cDULA in Figure 9. cDULA is able to capture the efficiency of DMALA without Metropolis-Hastings correction, which significantly improves the convergence speed and generates more diverse samples.

Effective sample size We find that the continuous exploration allows us to get closer to the high probability region before discrete sampling, allowing for larger steps of cDLS. In addition to accelerating the speed of convergence of the distribution, another significant benefit is the improvement in effective sample size (ESS). We run 100 chains in parallel with 5,000 iterations. Table 4 shows the ESS of the sampling of the RBM model, and it can be seen that our approach significantly improves the effective sample size.

G.5 Learning RBMs

Experiment Design Follow Pynadath et al. (2024), We use the same RBM structure as the sampling task, with 500 hidden units and 784 visible units. However, we apply the samplers of interest to the PCD algorithm introduced by Tieleman (2008). The model parameters are tuned via the Adam optimizer Kingma & Ba (2015) with a learning rate of .001.

Table 4: Effective sample size for RBM sampling. Our method outperforms gradient-based baselines across all datasets.

Dataset	GWG	DULA	DMALA	ACS	cDULA	cDMALA
MNIST	151.22±553.28	397.86±1051.08	676.07±1269.37	878.55±768.21	1948.68±1868.10	989.63±1518.10
eMNIST	143.49±513.91	312.11±939.11	262.66±755.76	478.32±372.79	1305.44±1802.20	587.28±1088.37
kMNIST	143.08±554.69	381.70±1023.72	518.35±1059.64	799.86±873.49	1601.81±1829.77	721.82±1218.88
Fashion	43.76±159.14	184.00±630.67	711.49±1232.28	977.23±1029.32	1357.86±1555.12	781.11±1387.46
Caltech	123.20±446.86	222.62±703.02	501.88±796.30	789.21±891.79	860.05±862.92	705.36±932.25
Omniglot	83.33±288.10	101.73±524.26	428.52±982.70	573.49±807.91	709.58±1395.64	504.40±1104.62

Table 5: Improvement on average acceptance probability.

Dataset	DMALA	cDMALA	Dataset	DMALA	cDMALA
MNIST	0.5307	0.6595	Fashion	0.5752	0.7025
eMNIST	0.5415	0.6706	Caltech	0.5339	0.7095
kMNIST	0.5307	0.6595	Omniglot	0.7430	0.8191

In order to evaluate the learned RBMs, we run AIS with Block-Gibbs as the sampler to calculate the log likelihood values for the models Neal (2001). We run AIS for 100,000 steps, which is adequate given the efficiency of Block Gibbs for this specific model. The results are shown in Table 1.

G.6 Learning EBMs

For all the experiments in this section, we use the stepsize α to be 0.1 for cDULA and 0.15 for cDMALA, α_0 to be 0.01 for the two.

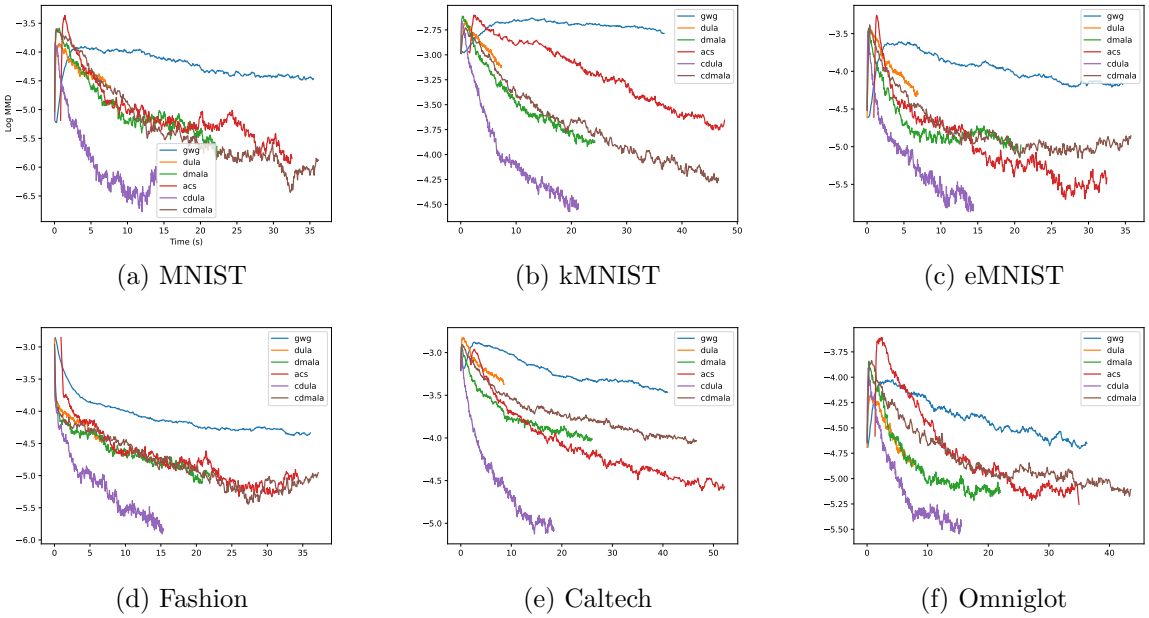
Ising Model We construct a training dataset of 2,000 instances by running 1,000,000 steps of Gibbs sampling for each instance. The model is trained by Persistent Constastive Divergence Tieleman (2008) with a buffer size of 256 samples. We also use the Adam optimizer with a learning rate of 0.001. The batchsize is 256. We train all models with an l_1 penalty with a penalty coefficient of 0.01 to encourage sparsity. The experiment setting is basically the same as Grathwohl et al. (2021); Zhang et al. (2022).

Deep EBMs We use the same EBM model architecture as Grathwohl et al. (2021); Zhang et al. (2022) and follow the same experimental design, where we implement ResNet with 8 residual blocks of 64 feature maps. We evaluate the models every 5,000 iterations by running AIS for 10, 000 steps. The reported results are from the model that performs the best on the validation set. The final reported numbers are generated by running 300, 000 iterations of AIS. All the models are trained with Adam with a learning rate of 0.0001 for 50, 000 iterations. We show the generated images with cDULA and cDMALA in Figure 10.

G.7 Binary Bayesian Neural Networks

Details of the Datasets (1) COMPAS: COMPAS J. Angwin & Kirchner (2016) is a dataset containing the criminal records of 6,172 individuals arrested in Florida. The task is to predict whether the individual will commit a crime again in 2 years. We use 13 attributes for prediction. (2) News: Online News Popularity Data Set [1] contains 39,797 instances of the statistics on the articles published by a website. The task is to predict the number of clicks in several hours after the articles are published. (3) Adult: Adult Income Dataset [2] is a dataset containing the information of US individuals from the 1994 census. The prediction

[1] https://archive.ics.uci.edu/ml/datasets/online+news+popularity
[2] https://archive.ics.uci.edu/ml/datasets/adult



(a) MNIST (b) kMNIST (c) eMNIST

(d) Fashion (e) Caltech (f) Omniglot

Figure 9: RBM Sampling with 5,000 sampling steps with results measured against time in seconds. cDULA is competitive both in terms of accuracy and efficiency.

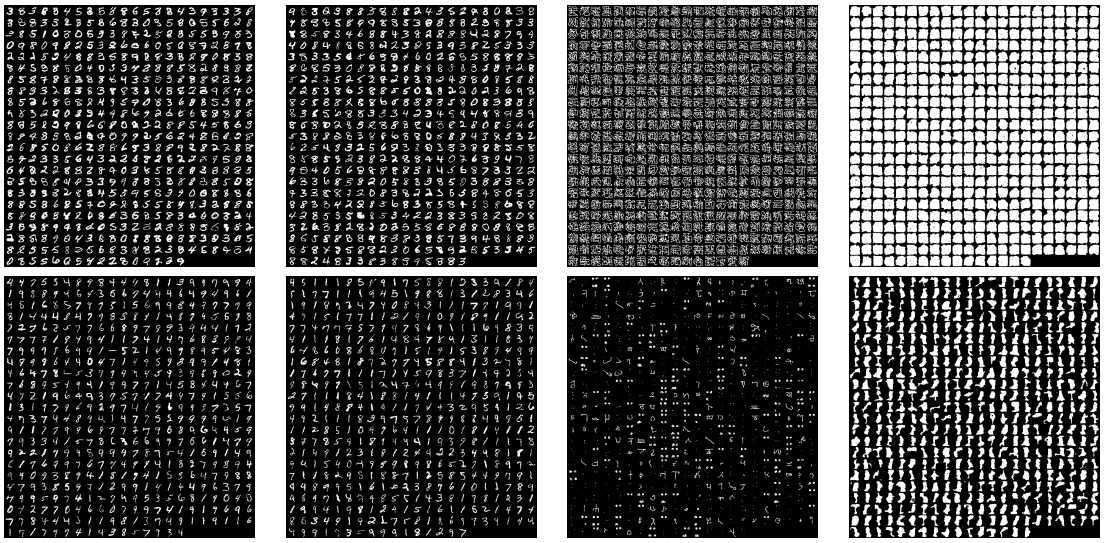


Figure 10: Left to right: Static MNIST, Dynamic MNIST, Omniglot, Caltech Silhouettes. Top: cDULA. Bottom: cDMALA.

task is to predict whether an individual makes more than 50K dollars per year. The dataset contains 44,920 data points. (4) Blog: Blog Feedback Buza (2014) is a dataset containing 54,270 data points from blog posts. The raw HTML documents of the blog posts were crawled and processed. The prediction task associated with the data is the prediction of the number of comments in the upcoming 24 hours. The feature of the dataset has 276 dimensions.

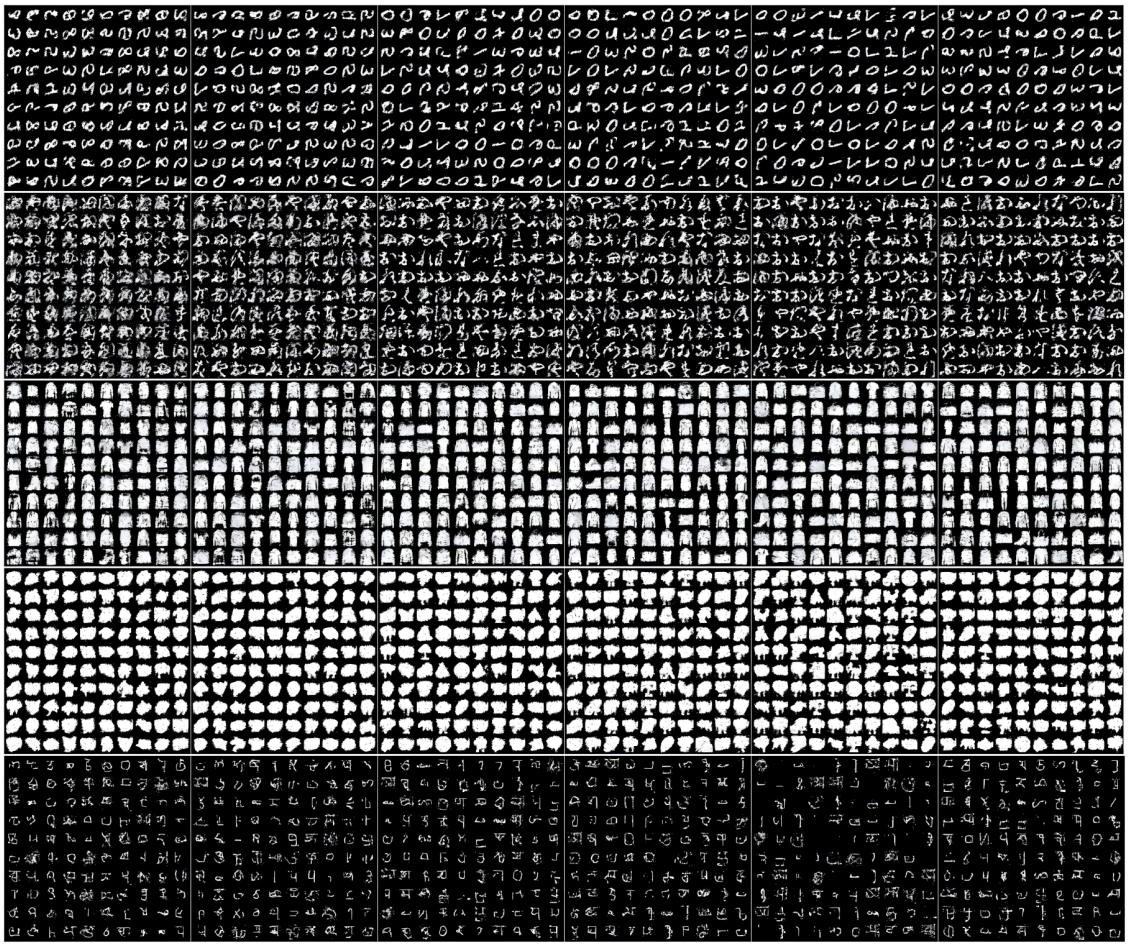


Figure 11: Images generated from RBMs trained by Constactive-Divergence with Block-Gibbs. Top to Bottom: eMNIST, kMNIST, Fashion, Caltech, Omniglot; Left to right: GWG, DULA, DMALA, ACS, cDULA and cDMALA.

Details on Training We run 10 chains in parallel and collect the samples at the end of training. All datasets are randomly partitioned into 80% for training and 20% for testing. The features and the predictive targets are normalized to (0,1). We set $\alpha = 0.1, \alpha_0 = 0.01$ for all datasets. We use a uniform prior over the weights. We train the Bayesian neural network for 2,000 steps. We use the full-batch training for the results in Section 6 so that Gibbs, GWG and DLP are also applicable. Beyond this, we could also get a stochastic variant for cDLS.