DISCS: A Benchmark for Discrete Sampling

Katayoon Goshvadi
Google Deepmind

Haoran Sun
Georgia Tech

Xingchao Liu
UT Austin

Azade Nova
Google Deepmind

Ruqi Zhang
Purdue University

Will Grathwohl
Google Deepmind

Dale Schuurmans
Google Deepmind

Hanjun Dai
Google Deepmind

Abstract

Sampling in discrete spaces, with critical applications in simulation and optimization, has recently been boosted by significant advances in gradient-based approaches that exploit modern accelerators like GPUs. However, two key challenges hinder the further research progress in discrete sampling. First, since there is no consensus on experimental settings, the empirical results in different research papers are often not comparable. Secondly, implementing samplers and target distributions often requires a nontrivial amount of effort in terms of calibration, parallelism, and evaluation. To tackle these challenges, we propose *DISCS* (DISCrete Sampling), a tailored package and benchmark that supports unified and efficient implementation and evaluations for discrete sampling in three types of tasks: sampling for classical graphical models, combinatorial optimization, and energy based generative models. Throughout the comprehensive evaluations in *DISCS*, we acquired new insights into scalability, design principles for proposal distributions, and lessons for adaptive sampling design. *DISCS* implements representative discrete samplers in existing research works as baselines, and offers a simple interface that researchers can conveniently design new discrete samplers and compare with baselines in a calibrated setup directly.

1 Introduction

Sampling in discrete spaces has been an important problem in physics (Edwards & Anderson, 1975; Baumgärtner et al., 2012), statistics (Robert & Casella, 2013; Carpenter et al., 2017), and computer science (LeCun et al., 2006; Wang & Cho, 2019) for decades. Since sampling from a target distribution $\pi(x) \propto \exp(-f(x))$ in a discrete space $\mathcal{X}$ is typically intractable, one usually resorts to MCMC methods(Metropolis et al., 1953; Hastings, 1970). However, except for a few algorithms such as Swedesen-Wang for the Ising model (Swendsen & Wang, 1987) and Hamze-Freitas for hierachical models (Hamze & de Freitas, 2012), which exploit special structure of the underlying problem, sampling in a general discrete space has primarily relied on Gibbs sampling, which exhibits notoriously poor efficiency in high dimensional spaces.

Recently, a family of locally balanced samplers (Zanella, 2020; Grathwohl et al., 2021; Sun et al., 2021; Zhang et al., 2022), using ratio informed proposal distributions, $\frac{\pi(y)}{\pi(x)}$, have significantly improved sampling efficiency by exploiting modern accelerators like GPUs and TPUs. From the perspective of gradient flow on the Wasserstein manifold of distributions, Gibbs sampling is simply a coordinate descent algorithm, whereas locally balanced samplers perform as full gradient descent (Sun et al., 2022a). Despite the advances in locally balanced samplers, a quantitative benchmark

Submitted to the 37th Conference on Neural Information Processing Systems (NeurIPS 2023) Track on Datasets and Benchmarks. Do not distribute.

is still missing. One important reason is that there is no consensus on the experimental setting. Particularly, the initialization of energy based generative models, random seeds used in graphical models, and the protocol of hyper-parameter tuning all have a significant impact on performance. As a result, some empirical results in different research papers may not be comparable. Under this circumstance, a unified benchmark is in crucial need for boosting the research in discrete sampling.

There are two key challenges that seriously hinder the appearance of such a benchmark. First, a sampler may perform well in one target distribution while poorly in another one. To thoroughly examine the performance of a sampler, a qualified benchmark needs to collect a set of representative distributions that covers the potential applications of a discrete sampler. Second, the evaluation of discrete samplers is complicated. Although the commonly used metric ESS (Vehtari et al., 2021) can effectively reflect the efficiency of a sampler in Monte Carlo integration or Bayesian inference, it is not very informative in scenarios when the sampler guides the search in combinatorial optimization problems, or performs as a decoder in deep generative models.

To address the two challenges, we propose *DISCS*, a tailored benchmark for discrete sampling. In particular, *DISCS* consists of three groups of tasks: sampling from classical graphical models, sampling for solving combinatorial optimization problems, and sampling from deep EBMs. These tasks cover the topics of simulation and optimization, and models ranging from hand-designed graphical models to learned deep EBMs. For each task, we collect the representative problems from both synthetic and real-world applications, for example graph partitioning for distributed computing and language model for text generation. We carefully design the evaluation metrics in *DISCS*. In sampling classical graphical models tasks, *DISCS* uses the ESS as standard. In sampling for solving combinatorial optimization tasks, *DISCS* runs simulated annealing (Kirkpatrick et al., 1983) with multiple chains and report the average of the best results in each chain. In sampling from energy based generative models, *DISCS* employs domain specific ways to measure the sample quality.

DISCS offers a convenient interface for researchers to implement new discrete samplers, without worrying about parallelism, experiment loop and evaluation. *DISCS* can efficiently sweep over different tasks and configurations in parallel and thus the evaluation reported in this paper can be easily reproduced. Also, *DISCS* implements existing discrete samplers random walk Metropolis (Metropolis et al., 1953), block Gibbs, Hamming ball sampler (Titsias & Yau, 2017), LB (Zanella, 2020), GWG (Grathwohl et al., 2021), PAS (Sun et al., 2021), DMALA (Zhang et al., 2022), DLMC (Sun et al., 2022a), and is actively maintaining to add new samplers. Researchers can directly compare the results with the state-of-the-art methods.

With *DISCS*, we observe an interesting phenomenon that the locally balanced weight function $g(t) = \sqrt{t}$ performs better (worse) than $g(t) = \frac{t}{t+1}$ when Ising model has temperature higher (lower) than the critical temperature. There have been a lot of studies about how to select the locally balanced function for a locally balanced sampler (Zanella, 2020; Sansone, 2022), but the answer remains open. We hope the observations in this paper can provide some insight on this question.

We wrap the *DISCS* package as a JAX library to facilitate the research in discrete sampling. The library will be open sourced at `https://github.com/google-research/discs`. The paper is organized as follows:

- In section 2, we cover the related sampling tasks and discrete samplers.
- In section 3, we formulate the discrete sampling problem.
- In section 4, we introduce the discrete sampling tasks and evaluation metrics in *DISCS*. We also report the results for existing discrete samplers.
- In section 5, we discuss the contribution and limitations of *DISCS*.

2 Related Work

Discrete sampling has been widely used to study the physical picture of spin glasses (Hukushima & Nemoto, 1996; Katzgraber et al., 2001), solve combinatorial optimization via simulated annealing (Kirkpatrick et al., 1983), and for traning or decoding deep energy based models (Wang & Cho, 2019;

Du et al., 2020; Dai et al., 2020b). However, they primarily depend on Gibbs sampling, which could be very slow in high dimensional space.

Since the seminal work Zanella (2020), the recent years have witnessed significant progresses for discrete sampling in the both theory and practice. Zanella (2020) introduces the locally balanced proposal $q(x, y) \propto g(\frac{\pi(y)}{\pi(x)})$, where $y \in N(X)$ restricted within a small neighborhood of x and $g(\cdot) : \mathbb{R}_+ \to \mathbb{R}_+$ satisfying $g(a) = ag(\frac{1}{a})$, and prove it is asymptotically optimal. In the following works, PAS (Sun et al., 2021) and DMALA (Zhang et al., 2022) generalize locally balanced proposal to large neighborhoods by introducing an auxiliary path and mimicking the diffusion process, respectively. Inspired by these locally balanced samplers, Sun et al. (2022a) generalize the Langevin dynamics in continuous space to *discrete Langevin dynamics* (DLD) in discrete space as a continuous time Markov chain $\frac{d}{dh}\mathbb{P}(X^{t+h} = y|X^t = x) = g(\frac{\pi(y)}{\pi(x)})$, and show that previous locally balanced samplers are simulations of DLD with different discretization strategies. In the view of Wasserstein gradient flow, the Gibbs sampling can be seen as coordinate descent and DLD gives a full gradient descent. Hence, locally balanced samplers induced from DLD provides a principled framework to utilize the modern accelerators like GPUs and TPUs to accelerate discrete sampling. Besides the discretization of DLD, another crucial part to design a locally balanced sampler is estimating the probability ratio $\frac{\pi(y)}{\pi(x)}$. Grathwohl et al. (2021) proposes to used gradient approximation $\frac{\pi(y)}{\pi(x)} \approx \exp(-\langle \nabla f(x), y - x\rangle)$ and obtains good performance on various classical models and deep energy based models. When the Hessian is available, Rhodes & Gutmann (2022); Sun et al. (2023a) use second order approximation via Gaussian integral trick (Hubbard, 1959) to further improve the sampling efficiency on skewed target distributions. When the gradient is not avaiable, Xiang et al. (2023) use zero order approximation via Newton's series.

Besides designing the sampler, Sun et al. (2022b) proves that when tuning path length in PAS (Sun et al., 2021), the optimal efficiency is obtained when average acceptance rate is 0.574, and design an adaptive tuning algorithm for PAS. Sansone (2022) learn locally balanced weight function for locally balanced proposal, but how to select the weight function in a principled manner is still unclear.

3 Formulation for Sampling in Discrete Space

The sampling in discrete space can be formulated as the following problem: in a finite discrete space $\mathcal{X}$, we have an energy function $f(\cdot) : \mathcal{X} \to \mathbb{R}$. We consider a target distribution

$$\pi(x) = \frac{\exp(-\beta f(x))}{Z}, \quad Z = \sum_{z \in \mathcal{X}} \exp(-\beta f(z)), \tag{1}$$

where β is the inverse temperature. When the normalizer Z is intractable, people usually resort to Markov chain Monte Carlo (MCMC). Metropolis-Hastings (M-H) (Metropolis et al., 1953; Hastings, 1970) is a commonly used general purpose MCMC algorithm. Specifically, given a current state $x^{(t)}$, the M-H algorithm proposes a candidate state y from a proposal distribution $q(x^{(t)}, y)$. Then, with probability

$$\min\Big\{1, \frac{\pi(y)q(y, x^{(t)})}{\pi(x^{(t)})q(x^{(t)}, y)}\Big\}, \tag{2}$$

the proposed state is accepted and $x^{(t+1)} = y$; otherwise, $x^{(t+1)} = x^{(t)}$. In this way, the detailed balance condition is satisfied and the M-H sampler generates a Markov chain $x^{(0)}, x^{(1)}, ...$ that has π as its stationary distribution.

4 Benchmark for Sampling in Discrete Space

The recent development of locally balanced samplers that use the ratio $\frac{\pi(y)}{\pi(x)}$ to guide $q(x, \cdot)$ have significantly improved the sampling efficiency in discrete space. However, there is no consensus for many experimental settings and the empirical results in different research papers may not be comparable. Under this circumstance, we propose ***DISCS*** as a benchmark for general purpose

samplers in discrete space. In Section 4.1, we introduces the baselines in *DISDS*. In Section 4.2, 4.3, 4.4, we introduce the tasks considered in *DISCS* and how the discrete samplers are evaluated on these tasks. We also report the results of the baselines.

4.1 Baselines

We include both classical discrete samplers and locally balanced samplers in recent research papers as baselines in our benchmark. Specifically, *DISCS* implements

1. Random Walk Metropolis (RWM) (Metropolis et al., 1953).
2. Block Gibbs (BG), where BG-<a> denotes using block Gibbs with block size a.
3. Hamming Ball Sampler (HB) (Titsias & Yau, 2017), where HB-<a>-<b> denotes using block size a and Hamming ball size b.
4. Gibbs with Gradient (GWG) (Grathwohl et al., 2021), a locally balanced sampler that use gradient to approximation the probability ratio. For binary distribution, GWG has a scaling factor L to determine how many sites to flip per step.
5. Path Auxiliary Sampler (PAS) (Sun et al., 2021), a locally balanced sampler that has a scaling factor L to determine the path length.
6. Discrete Metropolis Adjusted Langevin Algorithm (DMALA)(Zhang et al., 2022), a locally balanced sampler that has a scaling factor α to determine the step size.
7. Discrete Langevin Monte Carlo (DLMC) (Sun et al., 2022a), a locally balanced sampler that has a scaling factor τ to determine the simulation time of DLD. DLMC has multiple choices for its numerical solver to approximate the transition matrix. *DISCS* considers the two versions used in the original paper, DLMC that uses an interpolation and DLMCf that uses Euler's forward method.

Remark: weight function All the locally balanced samplers have the flexibility to select locally balanced function. $g(t) = \sqrt{t}$ and $g(t) = \frac{t}{t+1}$ are the two most commonly used weight functions. In this paper, we will use $\sqrt{t}$ by default. When we use both of them, we use <sampler>-<func> to refer the type of the weight function.

Remark: scaling Since the scalings of the proposal distribution in RWM, PAS, DMALA, and DLMC are tunable, we considers two versions with adaptive tuning or binary search tuning for fair comparison. Sun et al. (2022b, 2023b) propose adaptive tuning algorithm for PAS and DLMC when the target distribution is factorized. In practice, we find that they also apply well for other locally balanced samplers and for more general target distributions. Hence, in this paper, we use the adaptive tuning algorithm by default to tune the scaling for locally balanced samplers. In the several exceptions where the adaptive algorithm does not apply, we will use <sampler-name>-noA to indicate the results from binary search tuning.

4.2 Sampling from Classical Graphical Models

This section covers the classical graphical models that are widely used in physics and statistics, including Bernoulli Models, Ising Models (Ising, 1924), and Factorial Hidden Markov Models (Ghahramani & Jordan, 1995). The graphical models have large flexibility, for example, the number of discrete variables, the number of categories for each discrete variable, and the temperature of the model. The performances of different samplers can heavily depends on these configurations. *DISCS* provides tools to automatically sweep over hundreds of configurations by one click. Same as the routine in Monte Carlo integration or Bayesian inference, *DISCS* uses the Effective Sample Size (ESS) to measure the efficiency for each sampler and reports the ESS normalized by the number of calling energy function and the ESS normalized by the running time.

We use Ising Models as an example in the main text, and the more results are reported in Appendix. For an Ising Model defined on a 2D grid, where the state space $\mathcal{X} = \{-1, 1\}^{p \times p}$ represents the spins on all nodes. For each state $x \in \mathcal{X}$, the energy function is defined as:

$$f(x) = -\sum_{i,j} J_{ij} x_i x_j - \sum_i h_i x_i \tag{3}$$

where J_{ij} is the internel interaction and the h_i is the external field. The configurations J and h can be set freely in *DISCS*. In the main text, we report the results using the configuration from Zanella (2020). Specifically, $J_{ij} = 0.5$, $h_i = \mu_i + \sigma_i$, where $\sigma_i \sim \text{Uniform}(-1.5, 1.5)$ and $\mu_i = 0.5$ if node i is located in a circle has the same center as the 2D grid and radius $\frac{p}{2\sqrt{2}}$, else -0.5. We consider the target distribution $\pi(x) \propto \exp(-\beta f(x))$, where β is the inverse temperature. Using *DISCS*, one can easily investigate the influence of the model dimension. In Figure 1, one can see that the traditional samplers, RWM, GB, HB, have significant decrease in ESS when the model dimension increases, while the locally balanced samplers are less affected as the ratio information $\frac{\pi(y)}{\pi(x)}$ effectively guides the proposal distribution. The overall trends basically follows the prediction from Sun et al. (2022b) that the ESS is $O(d^{-1})$ for RWM and $O(d^{-\frac{1}{3}})$ for PAS.

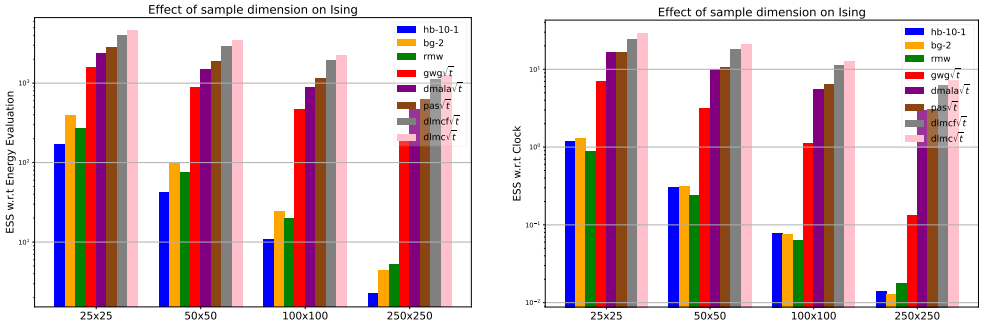


Figure 1: Results on Ising model with different dimensions

Through *DISCS*, researchers can also easily evaluate the samplers with different temperature. In Figure 2, we evaluate Ising models with inverse temperatures from 0.1607 to 0.7607. We consider Ising model without external field: $h_i \equiv 0$ and $J_{ij} \equiv 1$ as we know the critical temperature for this configuration is $\frac{2}{\log(1+\sqrt{2})}$ which means the critical point for inverse temperature $\beta = 0.4407$. From the results, we can see that

- The Ising model is harder to sample from when the inverse temperature β is closer to the critical point, which is consistent with the theory in statistical physics
- When the inverse temperature β is lower than the critical point, using weight function $g(t) = \sqrt{t}$ gives larger ESS; When the inverse temperature is larger than the critical point, using weight function $g(t) = \frac{t}{t+1}$ consistently obtains larger ESS.

The second observation implies that one should use ratio function $\frac{t}{t+1}$ for target distributions with sharp landscapes. We will revisit this conclusion in Figure 5 and Table 2.

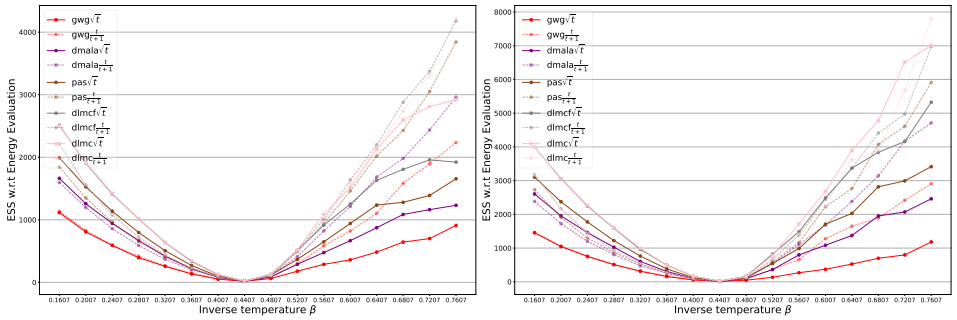


Figure 2: Performance of locally balanced samplers with different types of weight functions v.s temperature on: (left) 50×50 Ising model, (right) 100×100 Ising model

The categorical version of Ising model is Potts model, where each site of a state x_i has values in a symmetry group, instead of $\{-1, 1\}$. For simplicity, we denote the symmetry group as a set of one

hot vectors $\mathcal{C} = \{e_1, ..., e_c\}$ with $h_i \in \mathbb{R}^C, J_{ij} \in \mathbb{R}^{C\times C}$. In this way, the energy function becomes:

$$f(x) = -\sum_{i,j} x_i^\top J_{ij} x_j - \sum_i \langle h_i, x_i \rangle \tag{4}$$

In Figure 3, one can see the sampling efficiency is very robust with respect to the number of category. The result for BG-2 on Potts model with 256 categories are omitted as it takes over 100 hours.

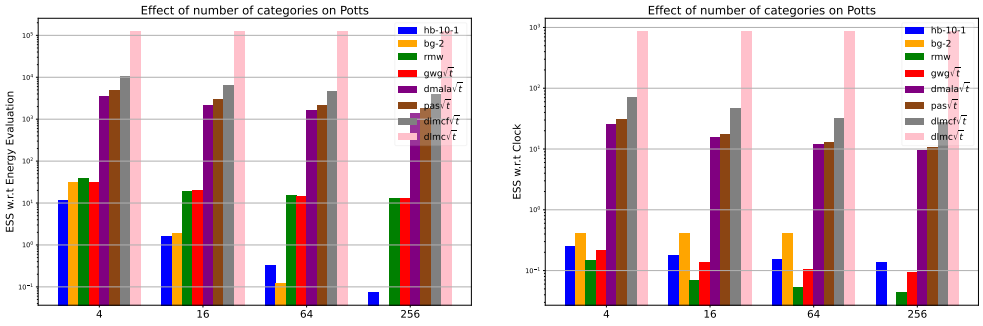


Figure 3: Results of Potts models with different number of categories

4.3 Sampling for Solving Combinatorial Optimiazation

Combinatorial optimization is a core challenge in domains like logistics, supply chain management and hardware design, and has been a fundamental problem of study in computer science for decades. Combining with simulated annealing Kirkpatrick et al. (1983), discrete sampling algorithm is a powerful tool to solve combinatorial optimization problems (Sun et al., 2023b). In expectation, a sampler with a faster mixing rate can find better solutions. Hence, the second type of tasks is sampling for solving combinatorial optimization problems. Currently, *DISCS* covers four problems: Maximum Independent Set, Max Clique, Max Cut, and Balanced Graph Partition. Without loss of generality, we consider combinatorial optimization that admit the following form:

$$\min_{x\in\mathcal{C}=\{0,1,\ldots,C-1\}^d} \quad a(x), \quad \text{s.t.} \quad b(x) = 0 \tag{5}$$

For ease of exposition, we also assume $b(x) \geq 0, \forall x \in \mathcal{C}$, but otherwise do not limit the form of a and b. To convert the optimization problem to a sampling problem, we first rewrite the constrained optimization into a penalty form via a penalty coefficient λ, then treat this as an energy function for an EBM. In particular, the energy function takes the form:

$$f(x) = a(x) + \lambda \cdot b(x) \tag{6}$$

Then, we define the probability of x at inverse temperature β by:

$$p_\beta(x) \propto \exp(-\beta f(x)) \tag{7}$$

A naive approach to this problem would be directly sampling from $p_{\beta\to\infty}(x)$, but such a distribution is highly nonsmooth and unsuitable for MCMC methods. Instead, following classical simulated annealing, we define a sequence of distributions parameterized by a sequence of decaying temperatures:

$$\mathcal{P} = [p_{\beta_0}(x), p_{\beta_1}(x), \ldots, p_{\beta_T}(x)] \tag{8}$$

where the sequence $\beta_0 < \beta_1 < \ldots < \beta_T \to \infty$ converges to a large enough value as T increases.

Example 1: Max Cut A cut on a graph $G = (V, E)$ is to find a partition of the graph nodes into two complementary sets $V = V_1 \cup V_2$, such that the number of edges in E between V_1 and V_2 is as large as possible. Max Cut is an unconstrained problem, which makes its formulation relatively simple. We can set $\mathcal{C} = \{0, 1\}$ such that $x_i = 0$ represents $i \in V_1$ and $x_i = 1$ means $x_i \in V_2$. Then we can write $a(x) = -x^\top Ax, b(x) \equiv 0$, where A is the adjacency matrix of G. By applying simulated

annealing with the same temperature schedule, we can compare the performance for each sampler. We report the results in Figure 4. The ratio is computed by dividing the cut size for the solutions obtained by running Gurobi for one hour (Dai et al., 2020a). The legends are sorted according to the optimal value they find. One can see that the PAS leads the results. Also, locally balanced samplers significantly outperforms the traditional samplers, especially when the graph size increases.

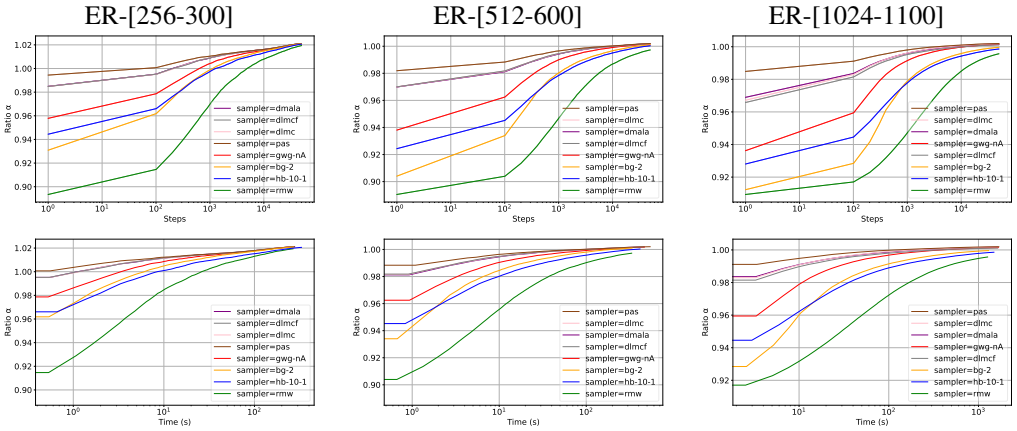


Figure 4: Results for MAXCUT on ER graphs. The ratio is computed by dividing the optimal cut size obtained from running Gurobi for 1 hour. (top) ratio with respect to number of M-H steps, (bottom) ratio with respect to running time.

Example 2: Maximum Independent Set On a graph $G = (V, E)$, an independent set $S \subset V$ means that for any $i, j \in S$, $(i, j) \notin E$. We can set $\mathcal{C} = \{0, 1\}$ such that $x_i = 0$ means $i \notin S$ and $x_i = 1$ means $i \in S$. Then we can write $a(x) = -\sum_{i \in V} x_i$ and $b(x) = \sum_{(i,j) \in E} x_i x_j$. For the penalty coefficient λ, we follow Sun et al. (2022c) to select $\lambda = 1.0001$ being a value slightly larger than 1. We run all samplers on five groups of small ER graphs with 700 to 800 nodes, each group has 128 graphs with densities varying 0.05, 0.10, 0.15, 0.20, and 0.25. We also run all samplers on 16 large ER graphs with 9000 to 11000 nodes. For each configurations, we run 32 chains with the same running time and report the average of the best results found by each chain in Table 1. One can easily see that PAS obtains the best result.

Table 1: Results for MIS on ER graphs. The set found by sampling algorithm is not necessary an independent set, we report a lower bound: set size - # pair of adjacent nodes in the set.

Sampler	ER[700-800]					ER[9000-11000]
	0.05	0.10	0.15	0.20	0.25	0.15
HB-10-1	100.374	58.750	41.812	32.344	26.469	277.149
BG-2	102.468	60.000	42.820	32.250	27.312	316.170
RMW	97.186	56.249	40.429	31.219	25.594	-555.674
GWG-nA	104.812	62.125	44.383	**34.812**	28.187	367.310
DMALA	104.750	62.031	44.195	34.375	28.031	357.058
PAS	**105.062**	**62.250**	**44.570**	34.719	**28.500**	**377.123**
DLMCf	104.450	62.219	44.078	34.469	28.125	354.121
DLMC	104.844	62.187	44.273	34.500	28.281	355.058

4.4 Sampling from Energy Based Generative Models

The discrete samplers can also play as the decoder in generative models. In particular, given a dataset $\mathcal{D} = \{X_i\}_{i=1}^N$ sampled from the target distribution π, one can train an energy function $f_\theta(\cdot)$, such that the energy based model $\pi_\theta(\cdot) \propto \exp(-f_\theta(\cdot))$ fits the dataset $\mathcal{D}$. *DISCS* provides multiple checkpoints for the energy function trained on real-world image or language datasets. Researchers can easily evaluate their samplers after loading the learned energy function.

For the models that are relatively simple, for example, Restricted Boltzmann Machine (RBM) trained on MNIST (LeCun, 1998) and fashion-MNIST (Xiao et al., 2017b), one can continue using ESS as the metric. In Figure 5, we evaluate the samplers on RBMs trained on MNIST with 25 and 200 hidden variables. One can see that 1) DLMC has the best performance, 2) when the hidden dimension is larger, the learned distribution becomes sharper, hence $\frac{t}{t+1}$ obtains better efficiency compared to $\sqrt{t}$, which is consistent with our observation in Figure 2. For more complicated deep energy based models, a sampler may fail to mix within a reasonable steps. In this case, ESS is not a good metric. To address this problem, *DISCS* provides multiple alternative measurements, including snapshots, annealed importance sampling, and domain specific scores.

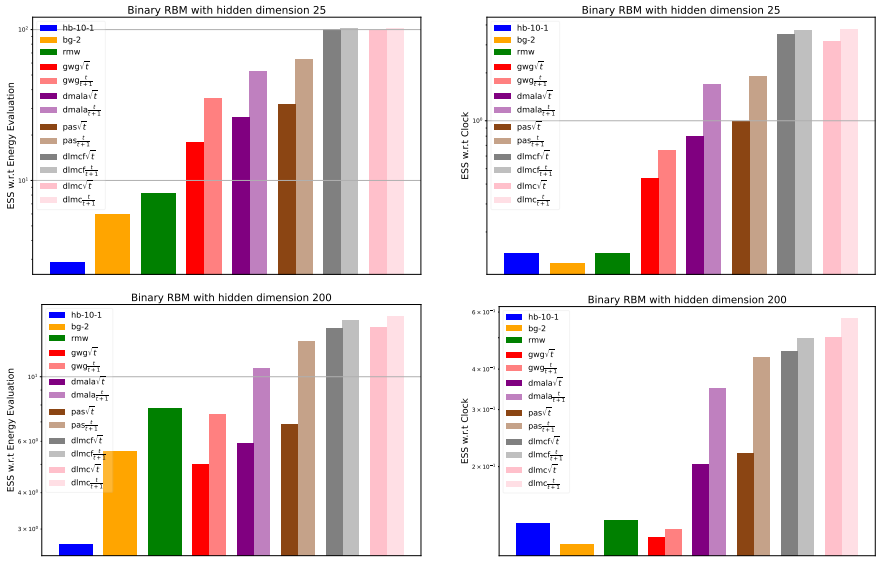
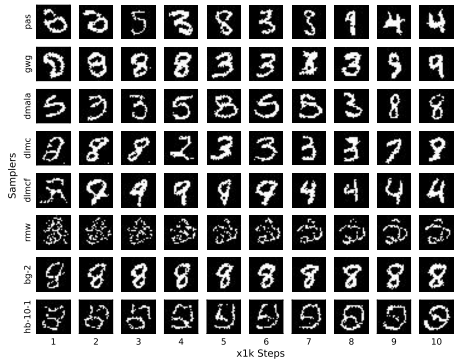


Figure 5: Results on RBMs trained on MNIST dataset. (top) RBM with 25 binary hidden variables, (bottom) RBM with 200 binary hidden variables

Snapshots After loading the checkpoint of energy based generative models, *DISCS* can generate snapshots of the sampling chains. For example, in Figure 6, we display the snapshots of sampling on a deep residual network trained on MNIST data (Sun et al., 2021) and on pretrained language model BERT[1]. One can see that locally balanced samplers generates samples with higher qualities, and can typically visit multiple modalities in the distribution.



MNIST

knowing even [MASK] details [MASK] [MASK] to understand [MASK] people involved and how they relate to one another .

RMW	knowing even the details gives lennon to understand the people involved and how they relate to one another. knowing even the details gives one to understand the people involved and how they relate to one another.
GWG	knowing even the details was necessary to understand the people involved and how they relate to one another. knowing even simple details is important to understand the people involved and how they relate to one another.
DMALA-nA	knowing even conversation details is good to understand the people involved and how they relate to one another. knowing even no details accountable, to understand the people involved and how they relate to one another.
DMALA	knowing even the details also easy to understand the people involved and how they relate to one another. knowing even the details is primitive to understand the people involved and how they relate to one another.
PAS	knowing even the details is important to understand the people involved and how they relate to one another. knowing even the details always helps to understand the people involved and how they relate to one another.
DLMCf	knowing even the details was easy to understand the people involved and how they relate to one another. knowing even the details also easy to understand the people involved and how they relate to one another.
DLMC-nA	knowing even complex details is important to understand the people involved and how they relate to one another. knowing even different details is good to understand the people involved and how they relate to one another.
DLMC	knowing even the details was good to understand the people involved and how they relate to one another. knowing even complex details allows you to understand the people involved and how they relate to one another.

BERT

Figure 6: Snapshots of energy based generative models: (left) snapshots for every 1k steps on MNIST ResNet, (right) snapshots for text filling task on BERT in Table 2

[1]loading the check point from https://huggingface.co/bert-base-uncased.

Domain Specific Scores In many deep generative tasks, the goal is to efficiently sample high-quality samples, instead of mixing in the learned energy based models. In this scenario, domain specific scores that directly evaluate the sample qualities are a better choice. For example, *DISCS* provides text filling tasks based on pre-trained language models like BERT (Wang & Cho, 2019; Devlin et al., 2018). Following the settings in prior work (Zhang et al., 2022), *DISCS* randomly sample 20 sentences from TBC (Zhu et al., 2015) and WiKiText-103 (Merity et al., 2016), mask four words in each sentence (Donahue et al., 2020), and sample 25 sentences from the probability distribution given by BERT. As a common practice in non-auto-regressive text generation, we select the top-5 sentences with the highest likelihood out of 25 sentences to avoid low-quality generation (Gu et al., 2017; Zhou et al., 2019). We evaluate the generated samples in terms of diversity and quality. For diversity, we use self-BLEU (Zhu et al., 2018) and the number of unique n-grams (Wang & Cho, 2019) to measure the difference between the generated sentences. For quality, we measure the BLEU score (Papineni et al., 2002) between the generated texts and the original dataset, which is the combination of TBC and WikiText-103. We report the quantitative results in Table 2. We do not have the results for HB and BG as they are computationally infeasible for this task with 30k+ tokens. In this task, the locally balanced sampler still outperforms RMW. Also, one can notice that the weight function $\frac{t}{t+1}$ significantly outperforms $\sqrt{t}$. The reason is that the overparameterized neural network is a low temperature system with sharp landscape. This phenomenon is consistent with the results in Figure 2.

Table 2: Quantative results on text infilling. The reference text for computing the Corpus BLEU is the combination of WT103 and TBC.

Methods	Self-BLEU (↓)	Unique n-grams (%) (↑)						Corpus BLEU (↑)
		Self		WT103		TBC		
		$n=2$	$n=3$	$n=2$	$n=3$	$n=2$	$n=3$	
RMW	92.41	6.26	9.10	18.97	26.73	19.33	26.67	16.24
GWG$\sqrt{t}$	85.93	11.22	17.14	23.16	35.56	23.58	35.56	16.75
DMALA$\sqrt{t}$	85.88	11.58	17.14	22.07	34.08	23.22	34.15	**17.06**
PAS$\sqrt{t}$	85.39	11.37	17.60	22.61	35.53	23.65	35.47	16.57
DLMCf$\sqrt{t}$	88.39	9.53	14.06	21.00	31.85	22.27	31.98	16.70
DLMC$\sqrt{t}$	85.28	12.05	17.65	24.03	36.34	24.51	36.27	16.45
GWG$\frac{t}{t+1}$	81.15	15.47	22.70	**25.62**	38.91	25.62	38.58	16.68
DMALA$\frac{t}{t+1}$	80.21	**16.36**	23.71	25.60	**39.39**	26.75	**39.72**	16.53
PAS$\frac{t}{t+1}$	81.02	15.62	22.65	25.59	39.28	26.08	39.48	16.69
DLMCf$\frac{t}{t+1}$	**80.12**	16.25	**23.76**	25.41	39.31	**26.86**	39.57	16.73
DLMC$\frac{t}{t+1}$	84.55	12.62	18.47	24.27	37.28	24.94	37.14	16.69

5 Conclusion

DISCS is a tailored benchmark for discrete sampling. It implements various discrete sampling tasks and state-of-the-art discrete samplers and enables a fair comparison. From the results, we know that DLMC leads in sampling from classical graphical models, PAS leads in solving combinatorial optimization problems, DLMCf and DMALA has the best performance on language models. We believe more efficient discrete samplers can be obtained by designing better discretization of DLD (Sun et al., 2022a). *DISCS* is a convenient tools during this process. The researcher can freely set the configurations for tasks and samplers and *DISCS* will automatically compile the program and run the processes in parallel. Besides, we observe that the choice of the locally balanced weight function should depends on the critical temperature of the target distribution. We believe this observation is insightful and will lead to a deeper understanding of locally balanced samplers.

Of course, *DISCS* does not include all existing tasks or samplers in discrete sampling, for example, the zero order (Xiang et al., 2023) and second order (Sun et al., 2023a) approximation methods. We will keep iterating *DISCS* and more features will be added in the future. We wrap *DISCS* to a JAX library. Researchers can conveniently implement customer tasks or samplers to accelerate their study and, in the meanwhile, contribute the code to *DISCS* for further improvement. We believe *DISCS* will be a powerful tools for researchers and facilitate the future research in discrete sampling.

References

Baumgärtner, A., Burkitt, A., Ceperley, D., De Raedt, H., Ferrenberg, A., Heermann, D., Herrmann, H., Landau, D., Levesque, D., von der Linden, W., et al. *The Monte Carlo method in condensed matter physics*, volume 71. Springer Science & Business Media, 2012.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. Stan: A probabilistic programming language. *Journal of statistical software*, 76(1), 2017.

Dai, H., Chen, X., Li, Y., Gao, X., and Song, L. A framework for differentiable discovery of graph algorithms. 2020a.

Dai, H., Singh, R., Dai, B., Sutton, C., and Schuurmans, D. Learning discrete energy-based models via auxiliary-variable local exploration. *arXiv preprint arXiv:2011.05363*, 2020b.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Donahue, C., Lee, M., and Liang, P. Enabling language models to fill in the blanks. *arXiv preprint arXiv:2005.05339*, 2020.

Du, Y., Li, S., Tenenbaum, J., and Mordatch, I. Improved contrastive divergence training of energy based models. *arXiv preprint arXiv:2012.01316*, 2020.

Edwards, S. F. and Anderson, P. W. Theory of spin glasses. *Journal of Physics F: Metal Physics*, 5 (5):965, 1975.

Ghahramani, Z. and Jordan, M. Factorial hidden markov models. *Advances in Neural Information Processing Systems*, 8, 1995.

Grathwohl, W., Swersky, K., Hashemi, M., Duvenaud, D., and Maddison, C. J. Oops I took a gradient: Scalable sampling for discrete distributions. *arXiv preprint arXiv:2102.04509*, 2021.

Gu, J., Bradbury, J., Xiong, C., Li, V. O., and Socher, R. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*, 2017.

Hamze, F. and de Freitas, N. From fields to trees. *arXiv preprint arXiv:1207.4149*, 2012.

Hastings, W. K. Monte Carlo sampling methods using Markov chains and their applications. 1970.

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

Hubbard, J. Calculation of partition functions. *Physical Review Letters*, 3(2):77, 1959.

Hukushima, K. and Nemoto, K. Exchange monte carlo method and application to spin glass simulations. *Journal of the Physical Society of Japan*, 65(6):1604–1608, 1996.

Ising, E. *Beitrag zur theorie des ferro-und paramagnetismus*. PhD thesis, Grefe & Tiedemann, 1924.

Katzgraber, H. G., Palassini, M., and Young, A. Monte carlo simulations of spin glasses at low temperatures. *Physical Review B*, 63(18):184422, 2001.

Kirkpatrick, S., Gelatt Jr, C. D., and Vecchi, M. P. Optimization by simulated annealing. *science*, 220 (4598):671–680, 1983.

LeCun, Y. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

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

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

Potts, R. B. Some generalized order-disorder transformations. In *Mathematical proceedings of the cambridge philosophical society*, volume 48, pp. 106–109. Cambridge University Press, 1952.

Rhodes, B. and Gutmann, M. Enhanced gradient-based mcmc in discrete spaces. *arXiv preprint arXiv:2208.00040*, 2022.

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

Sansone, E. Lsb: Local self-balancing mcmc in discrete spaces. In *International Conference on Machine Learning*, pp. 19205–19220. PMLR, 2022.

Sun, H., Dai, H., Xia, W., and Ramamurthy, A. Path auxiliary proposal for MCMC in discrete space. In *International Conference on Learning Representations*, 2021.

Sun, H., Dai, H., Dai, B., Zhou, H., and Schuurmans, D. Discrete Langevin sampler via Wasserstein gradient flow. *arXiv preprint arXiv:2206.14897*, 2022a.

Sun, H., Dai, H., and Schuurmans, D. Optimal scaling for locally balanced proposals in discrete spaces. *arXiv preprint arXiv:2209.08183*, 2022b.

Sun, H., Guha, E. K., and Dai, H. Annealed training for combinatorial optimization on graphs. *arXiv preprint arXiv:2207.11542*, 2022c.

Sun, H., Dai, B., Sutton, C., Schuurmans, D., and Dai, H. Any-scale balanced samplers for discrete space. In *The Eleventh International Conference on Learning Representations*, 2023a.

Sun, H., Goshvadi, K., Nova, A., Schuurmans, D., and Dai, H. Revisiting sampling for combinatorial optimization. In *International Conference on Machine Learning*, pp. 19205–19220. PMLR, 2023b.

Swendsen, R. H. and Wang, J.-S. Nonuniversal critical dynamics in Monte Carlo simulations. *Physical review letters*, 58(2):86, 1987.

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

Tran, T., Phung, D., and Venkatesh, S. Mixed-variate restricted boltzmann machines. In *Asian conference on machine learning*, pp. 213–229. PMLR, 2011.

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., and Bürkner, P.-C. Rank-normalization, folding, and localization: An improved r for assessing convergence of mcmc (with discussion). *Bayesian analysis*, 16(2):667–718, 2021.

Wang, A. and Cho, K. Bert has a mouth, and it must speak: Bert as a markov random field language model. *arXiv preprint arXiv:1902.04094*, 2019.

Xiang, Y., Zhu, D., Lei, B., Xu, D., and Zhang, R. Efficient informed proposals for discrete distributions via newton's series approximation. In *International Conference on Artificial Intelligence and Statistics*, pp. 7288–7310. PMLR, 2023.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017a.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017b.

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

Zhang, R., Liu, X., and Liu, Q. A Langevin-like sampler for discrete distributions. In *International Conference on Machine Learning*, pp. 26375–26396. PMLR, 2022.

Zhou, C., Neubig, G., and Gu, J. Understanding knowledge distillation in non-autoregressive machine translation. *arXiv preprint arXiv:1911.02727*, 2019.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pp. 19–27, 2015.

Zhu, Y., Lu, S., Zheng, L., Guo, J., Zhang, W., Wang, J., and Yu, Y. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 1097–1100, 2018.

A Experiment Details

The source code is open source at DISCS and the data used in this paper is avaiable at DISCS DATA.

A.1 Classical Graphical Models

For all the experiments of classical graphical models, we run 100 chains. The chains are run in parallel on 4 V100 GPUs, with each GPU handling a mini batch of 25 chains. We evaluate the performance of all the samplers and study the effect of sample shape, number of categories, locally balance function type for locally balanced samplers and the smoothness/sharpness of different models. Note that the result for BG-2 on Potts 10 and Categorical 8 model with 256 categories are omitted as it takes over 100 hours. The chain length is set as 1 million steps when studying the effect of number of categories and sample shape and in the other cases is set as 100k steps. For each experiment, as the sampling happens, all the samples of all chains are mapped separately on a randomly generated sample to a lower dimension of one. The ESS is calculated on the mapped samples after the burn-in phase i.e. after the generation of half of the chain using TensorFlow MCMC effective sample size. The ESS is averaged over all the chains and is reported over the running time and number of energy evaluation of each sampler. In the following sections, we provide the energy function we used for each of the classical graphical models.

A.1.1 Factorized Models

Factorized models are the simplest distributions in a discrete space, where each site is independent with others. Consider the category set of one hot vectors $\mathcal{C} = \{e_1, ..., e_C\}$ and the state space $\mathcal{X} = \mathcal{C}^N$. We have $|\mathcal{C}| = C$ is the number of category and N is the number of variables. The energy function of a factorized model is:

$$f(x) = \sum_{n=1}^{N} \langle x_n, \theta^n \rangle \tag{9}$$

where $\theta^d \in \mathbb{R}^C$. We denote the target distribution as Bernoulli model when $C = 2$ and Categorical model when $C > 2$. We report the results on Bernoulli models and Categorical models in Figure 7 and 8, respectively.

A.1.2 Ising Models

The Ising model (Ising, 1924) is a mathematical model of ferromagnetism in statistical mechanics. It consists of binary random variables arranged in a lattice graph $G = (V, E)$ and allows node to interact with its neighbors. The Potts model (Potts, 1952) is a generalization of the Ising model where the random variables are categorical. The energy function for Ising model and Potts model can be described as:

$$f(x) = -\sum_{n=1}^{N} \langle x_n, \theta_n \rangle - \sum_{(i,j)\in E} J_{ij}(x_i, x_j) \tag{10}$$

where we set $\theta^d \in \mathbb{R}^n$, and $J_{ij}(x_i, y_j) = 1_{\{x_i = y_j\}}$. For Ising model, we use $\theta^n \sim \text{Uniform}(-2, 1)$ for the outer part of the lattice graph, and $\theta^n \sim \text{Uniform}(-1, 2)$ for the inner part of the lattice graph. We report the results on Ising model and Potts model in Figure 9, 10.

A.1.3 Factorial Hidden Markov Model

FHMM (Ghahramani & Jordan, 1995) uses latent variables to characterize time series data. In particular, it assumes the continuous data $y \in \mathbb{R}^L$ is generated by hidden state $x \in \mathcal{C}^{L\times K}$. The probability function is:

$$p(x) = p(x_1) \prod_{l=2}^{L} p(x^t | x^{t-1}), \quad p(y|x) = \prod_{l=1}^{L} \mathcal{N}(y_t; \sum_{k=1}^{K} \langle W_k, x_{l,k} \rangle + b; \sigma^2) \tag{11}$$

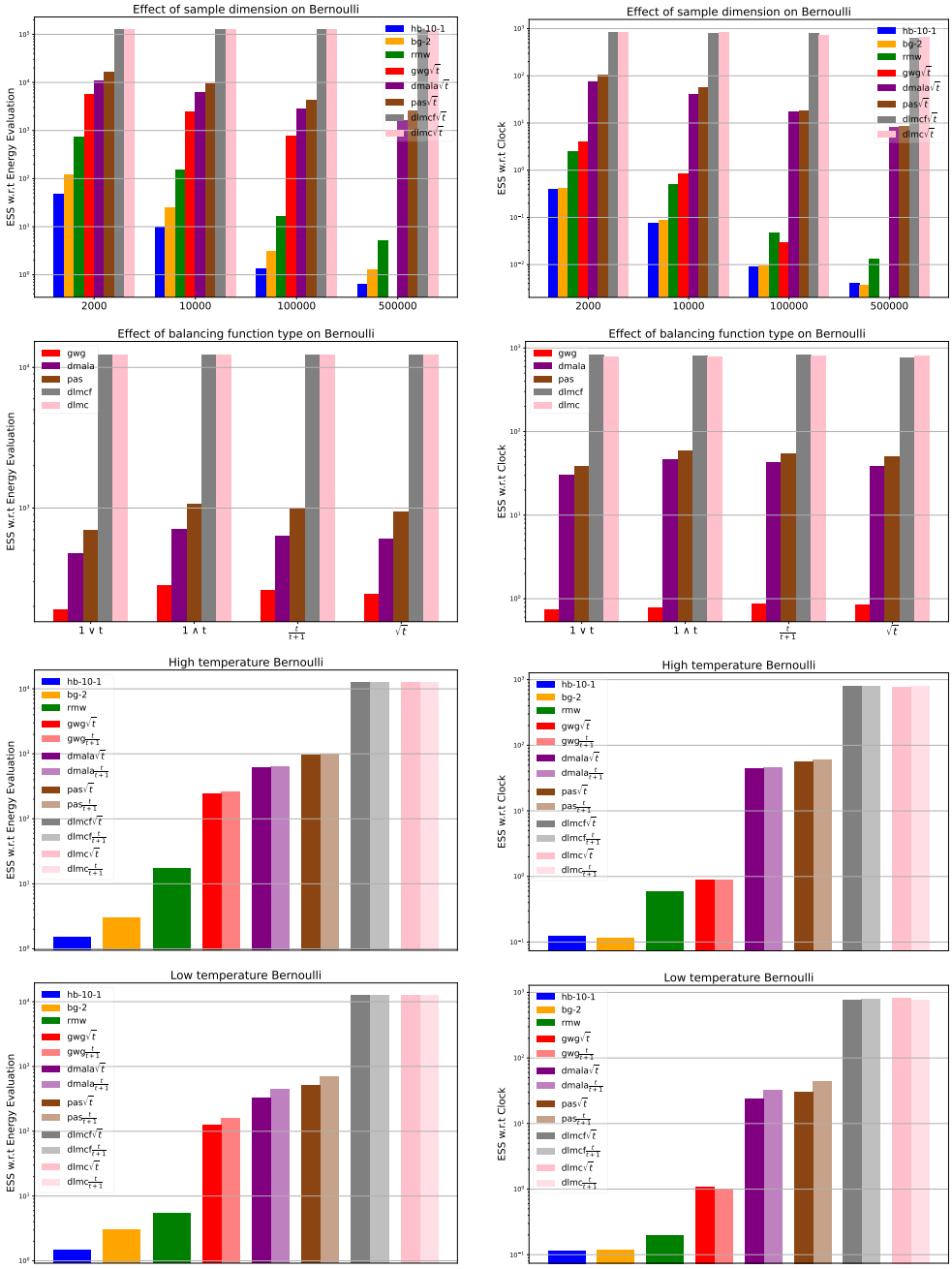


Figure 7: Results on Bernoulli Models

In particular, for binary model, we consider $\mathbb{P}(x_1 = 0) = 0.9, \mathbb{P}(x^t = x^{t-1}|x^{t-1}) = 0.8, \sigma = 2.0$. We use $L = 200, K = 50$ for high temperature setting and $L = 1000, K = 10$ in low temperature setting. For categorical model, we use $p(x_1|x_1 \neq 0)$ and $p(x^t|x^{t-1}, x^t \neq x^{t-1})$ as uniform distribution and we use $L = 200$, $K = 10$ with category number $C = 4, 8$. We report the results in Figure 11.

A.2 Combinatorial Optimization

Here we first provide the experimental details for the combinatorial optimization problems, MIS, Max Clique, Maxcut and, Balanced Graph Partition. The statistics of the synthetic datasets, including

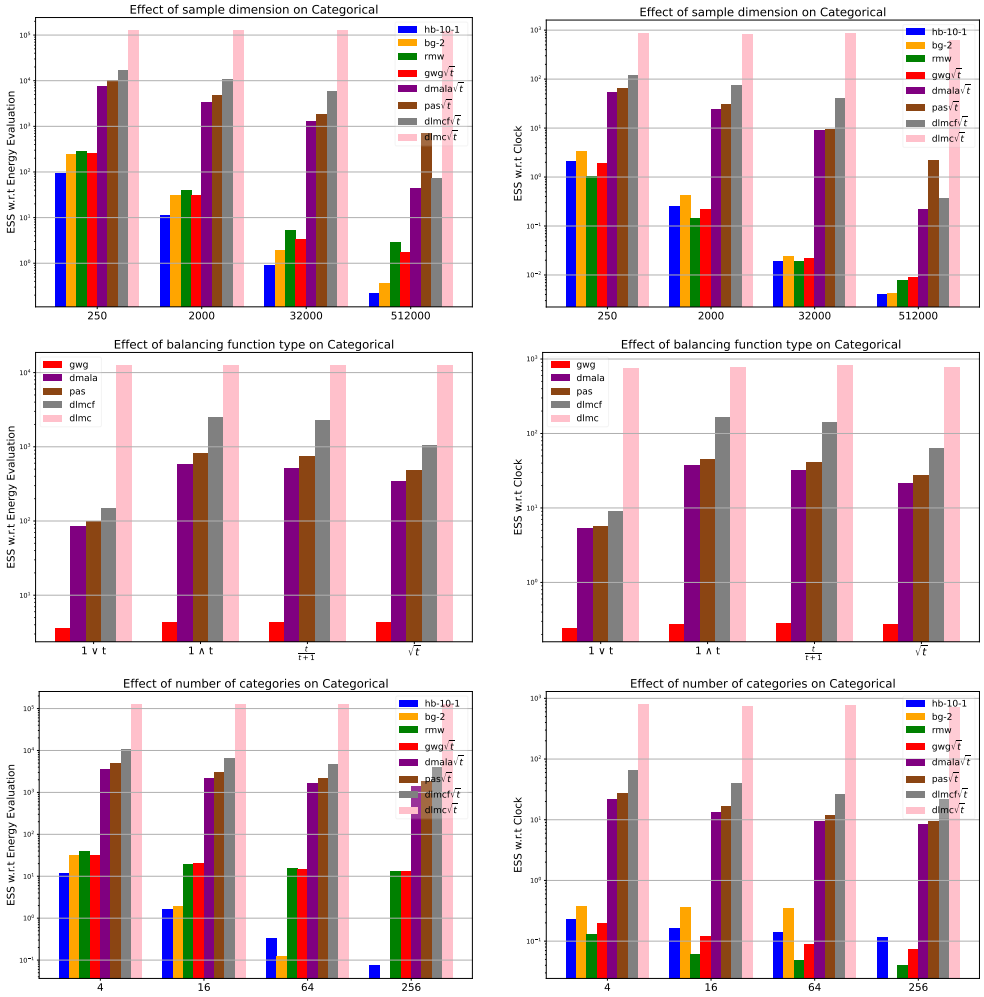


Figure 8: Results on Categorical Models

Table 3: Synthetic data statistics.

Name	MIS		Max Clique	Maxcut	
	ER-[700-800]	ER-[9000-11000]	RB	ER	BA
Max # nodes	800	10,915	475	1,100	1,100
Max # edges	47,885	1,190,799	90,585	91,239	4,384
# Test instances	128	16	500	1,000	1,000

the maximum number of nodes/edges in a graph, and the number of test instances are reported in 3. Additionally the statistics of real-world graphs are in 4. For Maxcut-ba and all Balanced Graph Partition and MIS graphs, we used 32 as the number of chains and for Maxcut-optsicom, Maxcut-er, ans all MaxClique graphs we used 16. The data used for these experiments could be found at DISCS DATA.

We run all the experiments on 8 V100 GPUs in parallel. For only Maxcut Optsicom graph, we use 2 V100 GPUs. The test instances are divided evenly between the GPUs and are run in parallel. For each experiment, we report the average of the best solution found over the number of test instances along with the end-to-end run time (in seconds) of each in tables. We report the results for all the

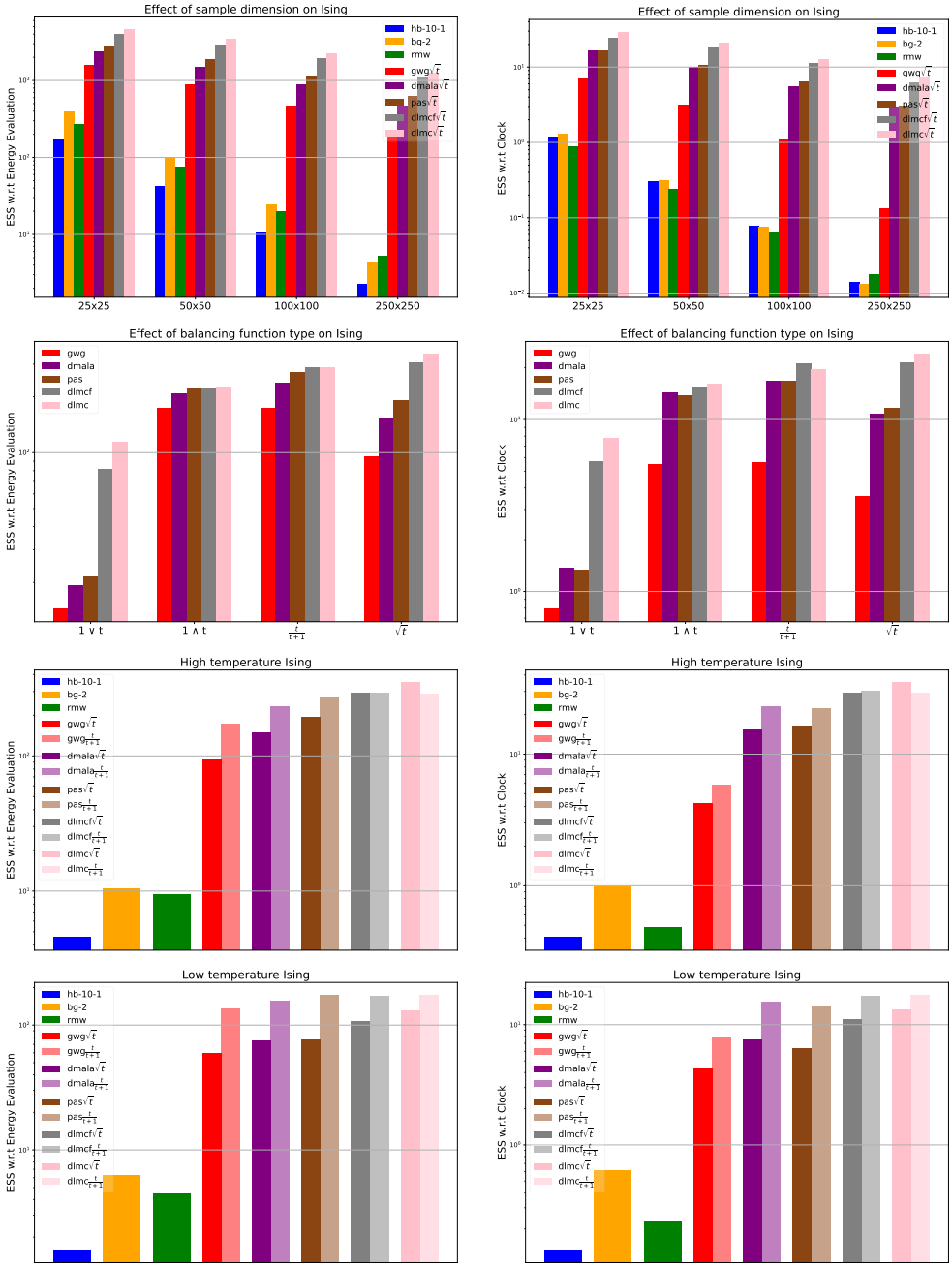


Figure 9: Results on Ising Models

samplers and plot the their solution through as the chain is being generated over M-H step and the running time.

In the following sections, we provide the actual energy function we used for each of the problems we experimented in the main paper. For a graph $G = (V, E)$ we label the nodes in V from 1 to d. The adjacency matrix is represented as A. For a weighted graph we simply let A_{ij} denote the edge weight between node i and j. For constraint problems, we follow Sun et al. (2022c) to select penalty coefficient λ as the minimum value of λ such that $x^* := \arg\min f(x)$ is achieved at x^* satisfying the original constraints. Such a choice of the coefficient guarantees the target distribution converges

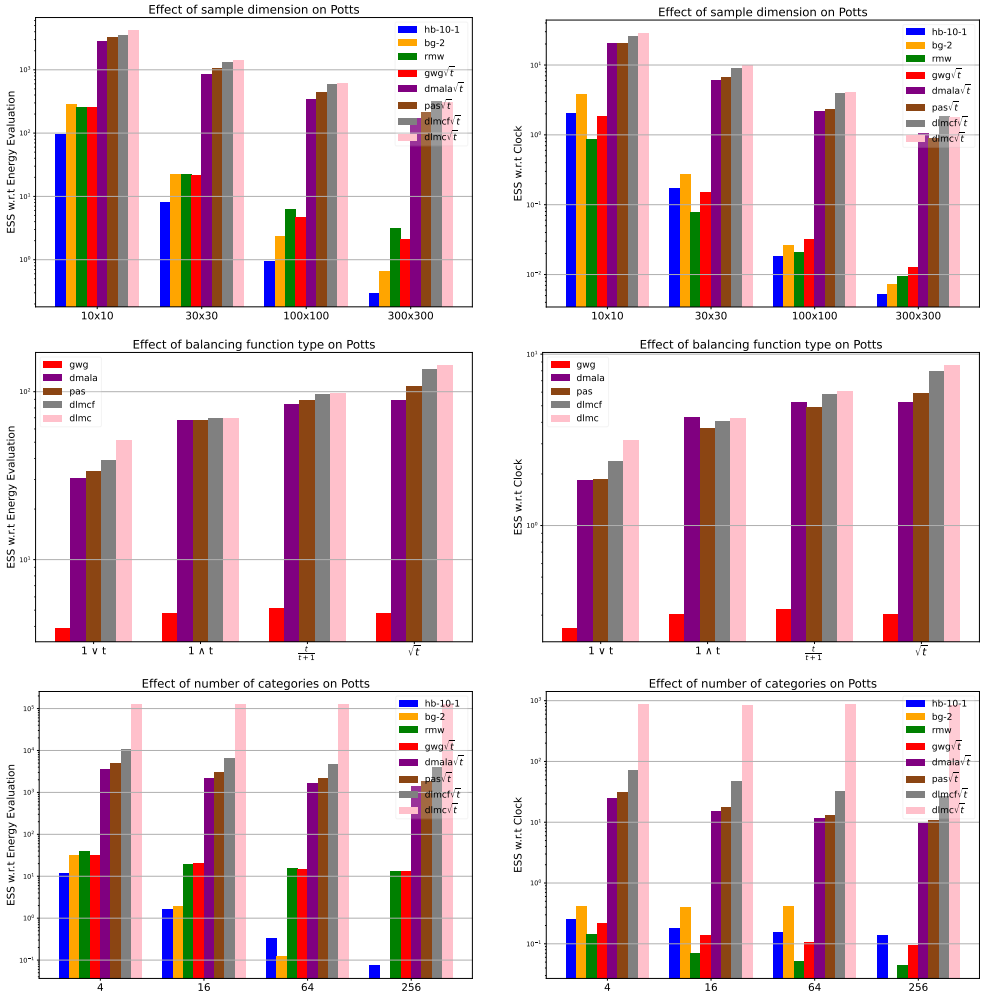


Figure 10: Potts

Table 4: Real-world data statistics.

Name	MIS	Max Clique	Maxcut	Balanced Graph Partition				
	SATLIB	Twitter	Optsicom	MNIST	VGG	ALEXNET	RESNET	INCEPTION
Max # nodes	1,347	247	125	414	1,325	798	20,586	27,114
Max # edges	5,978	12,174	375	623	2,036	1,198	32,298	40,875
# Test instances	500	196	10	1	1	1	1	1

to the optimal solution of the original CO problems while keeping the target distribution as smooth as possible.

A.2.1 MIS

The MIS has the integer programming formulation as

$$\min_{x\in\{0,1\}^d} -\sum_{i=1}^{d} c_i x_i, \quad \text{s.t. } x_i x_j = 0,\ \forall (i,j) \in E \tag{12}$$

We use the corresponding energy function in the following quadratic form:

$$f(x) := -c^T x + \lambda \frac{x^T A x}{2} \tag{13}$$

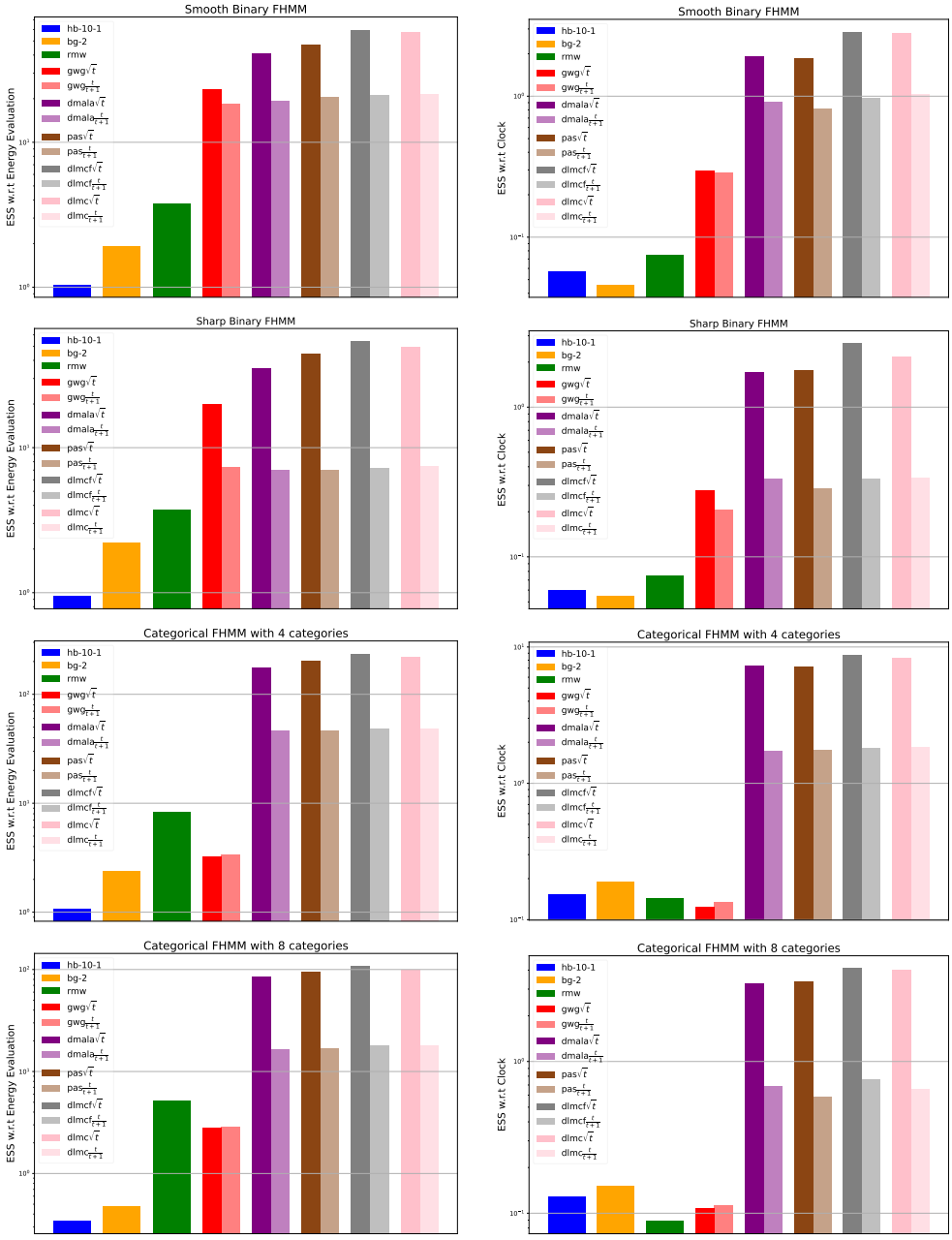


Figure 11: Results on FHMMs

In our experiments c equals to 1 and we use $\lambda = 1.0001$. In post processing, we iteratively go through all nodes x_i for $i = 1, ..., d$. If there exists $x_j = 1$ for $(x_i, x_j) \in E$, we flip its value $x_j = 0$. After post processing, the state x is guaranteed to be feasible in the original MIS problem. We provide the average of the best solutions over all number of instances along with their corresponding running time at 5. The plots of the experiments could be found at 12.

We also conduct experiments to justify the results are robust regarding the choice of the penalty coefficient. In Figure 13, we use penalty coefficient $\lambda \in \{1.001, 1.01, 1.1, 2\}$ on ER-[700-800] graphs with density $\{0.05, 0.10, 0.15, 0.20, 0.25\}$. We also use a dashed line to represent the optimal value obtained by running Gurobi-10 for 1 hour. From the results, we can observe that 1) PAS

Table 5: MIS.

Sampler	Graphs	ER[700-800]					ER[9000-11000]	SATLIB
	Density	0.05	0.10	0.15	0.20	0.25	0.15	
HB-10-1	Size	100.374	58.750	41.812	32.344	26.469	277.149	422.427
	Time(s)	426.185	390.810	684.590	414.067	429.879	15139.425	5381.857
BG-2	Size	102.468	60.000	42.820	32.250	27.312	316.170	422.200
	Time(s)	291.427	290.042	562.986	295.024	288.109	13079.125	3027.204
RMW	Size	97.186	56.249	40.429	31.219	25.594	-555.674	420.284
	Time(s)	284.092	293.517	499.577	297.140	281.772	12401.737	2955.729
GWG-nA	Size	104.812	62.125	44.383	34.812	28.187	367.310	422.971
	Time(s)	278.885	308.873	737.671	303.435	310.551	24698.296	3540.670
DMALA	Size	104.750	62.031	44.195	34.375	28.031	357.058	423.641
	Time(s)	291.271	292.131	714.614	297.848	298.732	24769.380	3465.343
PAS	Size	105.062	62.250	44.570	34.719	28.500	377.123	424.143
	Time(s)	299.004	310.765	759.372	299.569	308.475	25242.166	4826.039
DLMCF	Size	104.450	62.219	44.078	34.469	28.125	354.121	423.387
	Time(s)	291.366	301.554	726.287	302.667	300.413	24892.216	3679.425
DLMC	Size	104.844	62.187	44.273	34.500	28.281	355.058	423.479
	Time(s)	293.235	294.975	725.326	294.688	299.884	24976.312	3523.320

consistently obtains the best results, 2) locally balanced samplers have results consistently better than traditional sampler and Gurobi.

A.2.2 Max Clique

The max clique problem is equivalent to MIS on the dual graph. In our experiments c equals to 1.

$$\min_{x\in\{0,1\}^d} -\sum_{i=1}^{d} c_i x_i, \quad \text{s.t. } x_i x_j = 0,\ \forall (i,j) \notin E \tag{14}$$

The energy function is

$$f(x) := -c^T x + \frac{\lambda}{2}\left(\mathbf{1}^\top x \cdot (\mathbf{1}^\top x - 1) - x^T A x\right) \tag{15}$$

In our experiments c equals to 1 and we use $\lambda = 1.0001$. In post processing, we iteratively go through all nodes x_i for $i = 1, ..., d$. If there exists $x_j = 1$ for $(x_i, x_j) \notin E$, we flip its value $x_j = 0$. After post processing, the state x is guaranteed to be feasible in the original Max Clique problem. We provide the average of the best solutions over all number of instances along with their corresponding running time at 6. The plots of the experiments could be found at 14.

A.2.3 Maxcut

We optimize the following problem:

$$\min_{x\in\{-1,1\}^d} -\sum_{(i,j)\in E} A_{i,j}\left(\frac{1 - x_i x_j}{2}\right) \tag{16}$$

Note that for simplicity each dimension of x is selected from $\{-1, 1\}$. To represent the corresponding energy function for $x \in \{0, 1\}^d$, we have

$$f(x) := -\sum_{(i,j)\in E} A_{i,j}\left(\frac{1 - (2x_i - 1)(2x_j - 1)}{2}\right) \tag{17}$$

In our experiments A_{ij} equals to 1. Since the problem is always feasible, the post processing is identity map. We provide the average of the best solutions over all number of instances along with their corresponding running time at 7. The plots of the experiments could be found at 15.

Table 6: Max Clique.

Sampler	Results	RB	TWITTER
HB-10-1	Ratio α	0.850	0.966
	Time(s)	1724.893	6.817
BG-2	Ratio α	0.859	0.995
	Time(s)	1592.808	6.327
RMW	Ratio α	0.841	0.584
	Time(s)	1683.397	5.664
GWG-nA	Ratio α	0.878	0.999
	Time(s)	2525.801	6.032
DMALA	Ratio α	0.876	0.999
	Time(s)	2561.617	6.190
PAS	Ratio α	0.878	0.999
	Time(s)	2542.538	6.160
DLMCF	Ratio α	0.871	0.999
	Time(s)	2532.835	5.988
DLMC	Ratio α	0.875	0.999
	Time(s)	2639.588	6.124

Table 7: Maxcut.

Sampler	Results	BA							ER			OPTSICOM
		16-20	32-10	64-75	128-150	256-300	512-600	1024-1100	256-300	512-600	1024-1100	
HB-10-1	Ratio α	1.000	1.000	1.000	1.000	1.000	1.008	1.014	1.020	1.000	0.998	1.000
	Time(s)	742.568	754.613	749.626	783.278	792.338	1143.302	1890.534	331.019	416.002	1488.382	75.347
BG-2	Ratio α	1.000	1.000	1.000	1.000	1.000	1.009	1.014	1.021	1.001	0.999	1.000
	Time(s)	517.183	538.258	550.082	553.863	531.720	578.991	1157.571	269.116	337.014	1295.219	17.050
RMW	Ratio α	0.998	1.000	1.000	1.000	0.999	1.005	1.007	1.019	0.997	0.996	1.000
	Time(s)	534.215	534.615	528.641	558.608	541.302	574.778	1065.852	267.071	333.402	1266.630	58.960
GWG-nA	Ratio α	1.000	1.000	1.000	1.000	1.000	1.010	1.017	1.021	1.002	1.001	1.000
	Time(s)	522.094	531.425	578.917	551.923	545.634	724.721	1427.577	264.202	466.199	1666.021	80.124
DMALA	Ratio α	1.000	1.000	1.000	1.000	1.000	1.010	1.018	1.021	1.002	1.002	1.000
	Time(s)	531.433	538.938	568.224	549.026	544.568	750.909	1490.872	277.855	461.179	1643.135	53.509
PAS	Ratio α	1.000	1.000	1.000	1.000	1.000	1.010	1.018	1.021	1.002	1.002	1.000
	Time(s)	519.842	538.814	550.035	550.578	580.051	940.408	1917.954	278.005	543.607	1689.071	59.213
DLMCF	Ratio α	1.000	1.000	1.000	1.000	1.000	1.010	1.018	1.021	1.002	1.001	1.000
	Time(s)	521.592	526.289	545.877	557.564	533.119	765.719	1510.380	272.841	452.252	1639.539	52.552
DLMC	Ratio α	1.000	1.000	1.000	1.000	1.000	1.010	1.018	1.021	1.002	1.002	1.000
	Time(s)	531.003	550.118	543.287	544.611	542.677	765.104	1564.198	271.262	451.080	1642.223	53.368

A.2.4 Balanced graph partition

We find the following objective for balanced graph partition gives the best result:

$$f(x) := \sum_{s=1}^{k} \sum_{(i,j)\in E} \mathbb{I}\left(x_i \neq x_j \&\& (x_i = s || x_j = s)\right) + \sum_{s=1}^{k} \left(d/k - \sum_{i=1}^{d} \mathbb{I}(x_i = s) \right)^2 \tag{18}$$

where k is the number of partitions. Since the problem is always feasible, the post processing is identity map. We provide the edge cut ratio and balanceness of the best samples over all the chains at 8.

A.3 Energy Based Generative Models

A.3.1 Restricted Boltzmann Machine

The RBM is an unnormalized latent variable model, with a visible random variable $v \in \mathcal{C}^N$ and a hidden random variable $h \in \{0,1\}^M$. When v is binary, we call it a binary RBM (binRBM) and when v is categorical, we call it a categorical RBM (catRBM). The energy function of both binRBM and catRBM (Tran et al., 2011) can be written as:

$$f(v) = \sum_{h} \left[-\sum_{n=1}^{N} \langle v_n, \theta_n \rangle - \sum_{m=1}^{M} \beta_m h_m - \sum_{d,m} \langle h_m \theta_{m,d}, v_n \rangle \right] \tag{19}$$

Table 8: Balanced graph partition.

Metric	Samplers	VGG	MNIST-conv	ResNet	AlexNet	Inception-v3
Edge cut ratio ↓	HB-10-1	0.050	0.046	0.050	0.037	0.065
	BG-2	**0.048**	**0.045**	0.050	0.038	0.069
	RMW	0.054	0.046	0.092	0.052	0.117
	GWG	0.102	0.046	0.159	0.063	0.164
	DMALA	0.084	0.058	0.178	0.063	0.176
	DMALA-nA	0.059	**0.045**	0.048	0.039	0.054
	PAS	0.053	**0.045**	**0.047**	**0.037**	**0.052**
	PAS-nA	0.084	0.050	0.138	0.053	0.144
	DLMCF	0.086	0.063	0.178	0.053	0.176
	DLMCF-nA	0.092	0.069	0.048	0.085	**0.052**
	DLMC	0.105	0.056	0.183	0.097	0.182
	DLMC-nA	0.113	0.048	0.082	0.091	0.086
Balanceness ↑	HB-10-1	0.999	0.999	0.999	0.999	0.999
	BG-2	0.999	0.997	0.999	0.999	0.999
	RMW	0.999	0.998	0.999	0.999	0.999
	GWG	0.999	0.997	0.999	0.999	0.999
	DMALA	0.999	0.998	0.999	0.999	0.999
	DMALA-nA	0.999	0.997	0.999	0.999	0.999
	PAS	0.999	0.997	0.999	1.000	0.999
	PAS-nA	0.999	0.998	0.999	0.999	0.999
	DLMCF	0.999	0.997	0.999	0.999	0.999
	DLMCF-nA	0.999	0.995	0.999	0.999	0.999
	DLMC	0.999	0.994	0.999	0.999	0.999
	DLMC-nA	0.999	0.993	0.999	0.999	0.999

Unlike the previous three models, where the parameters are hand designed, we train binary RBM on MNIST (LeCun, 1998) and categorical RBM on Fashion-MNIST (Xiao et al., 2017a) using contrastive divergence Hinton (2002). Across all settings, we have $D = 784$. For binary models, we use $M = 25$ for high temperature setting and $M = 200$ for low temperature setting. For categorical models, we use $M = 100$. We report the results in Figure 16. The experimental setup is similar to classical graphical models.

A.3.2 Deep residual network

In this experiment, we train a deep residual network on MNIST, Omniglot and Caltech dataset. The model paramters and experimental setup could be found at DISCS DATA. We then use all the samplers to sample from the trained energy models. We use the chain length of 10k and number of chains of 100. We randomly selected one chain from the 100 chains and save its sample after each 1k steps, giving us 10 images per each chain for each sampler 17. We can see that locally balanced samplers are able to generate higher quality images faster and visit more diverse modalities.

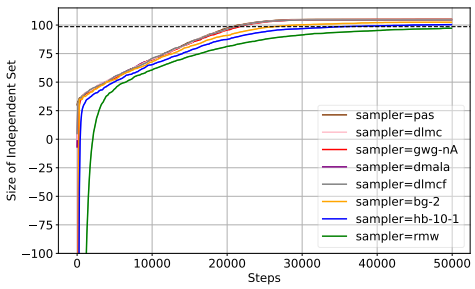
A.3.3 Text Infilling

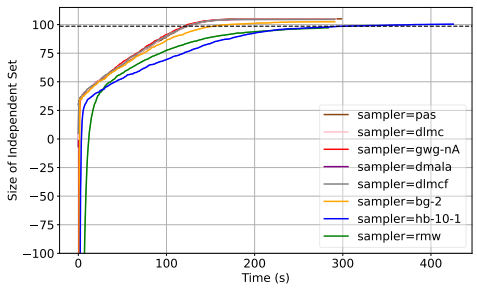
Here we additionally provide the performance of the locally balanced samplers in their non adaptive condition observed at 9. The data used for this experiment could be found at DISCS DATA.

Table 9: Quantative results on text infilling. The reference text for computing the Corpus BLEU is the combination of WT103 and TBC.

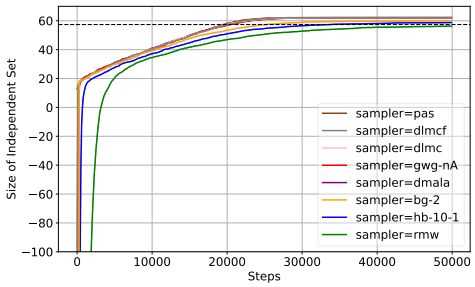
Methods	Self-BLEU (↓)	Unique n-grams (%) (↑)						Corpus BLEU (↑)
		Self		WT103		TBC		
		$n=2$	$n=3$	$n=2$	$n=3$	$n=2$	$n=3$	
RMW	92.41	6.26	9.10	18.97	26.73	19.33	26.67	16.24
GWG$\sqrt{t}$	85.93	11.22	17.14	23.16	35.56	23.58	35.56	16.75
GWG$\frac{t}{t+1}$	81.15	15.47	22.70	25.62	38.91	25.62	38.58	16.68
DMALA-nA$\sqrt{t}$	83.99	13.26	19.52	24.33	36.40	25.30	36.40	16.37
DMALA-nA$\frac{t}{t+1}$	80.44	15.86	23.58	25.79	39.88	26.57	40.20	16.64
DMALA$\sqrt{t}$	85.88	11.58	17.14	22.07	34.08	23.22	34.15	17.06
DMALA$\frac{t}{t+1}$	80.21	16.36	23.71	25.60	39.39	26.75	39.72	16.53
PAS$\sqrt{t}$	85.39	11.37	17.60	22.61	35.53	23.65	35.47	16.57
PAS$\frac{t}{t+1}$	81.02	15.62	22.65	25.59	39.28	26.08	39.48	16.69
DLMCf-nA$\sqrt{t}$	91.57	7.25	10.42	19.53	28.31	20.13	28.18	16.56
DLMCf-nA$\frac{t}{t+1}$	81.66	15.31	21.78	26.39	39.56	27.60	39.69	16.31
DLMCf$\sqrt{t}$	88.39	9.53	14.06	21.00	31.85	22.27	31.98	16.70
DLMCf$\frac{t}{t+1}$	80.12	16.25	23.76	25.41	39.31	26.86	39.57	16.73
DLMC-nA$\sqrt{t}$	83.74	12.74	19.64	24.27	37.27	24.94	37.34	16.73
DLMC-nA$\frac{t}{t+1}$	82.26	14.18	21.41	25.51	39.10	26.18	39.29	16.55
DLMC$\sqrt{t}$	85.28	12.05	17.65	24.03	36.34	24.51	36.27	16.45
DLMC$\frac{t}{t+1}$	84.55	12.62	18.47	24.27	37.28	24.94	37.14	16.69

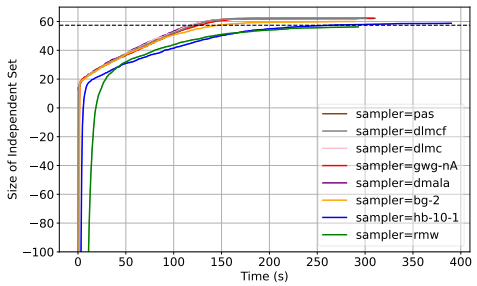
ER[800-800-0.05]



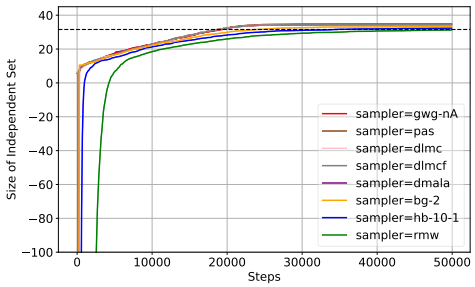


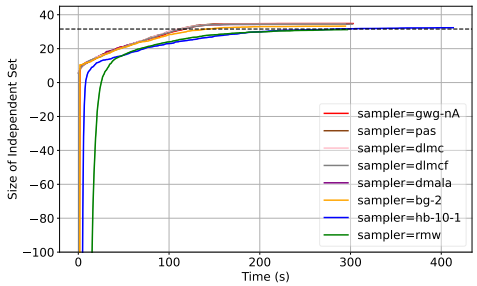
ER[800-800-0.10]



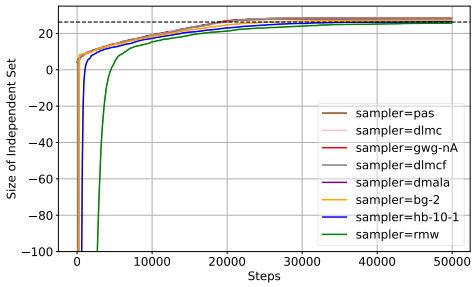


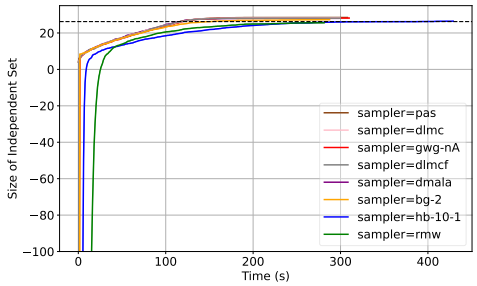
ER[800-800-0.20]



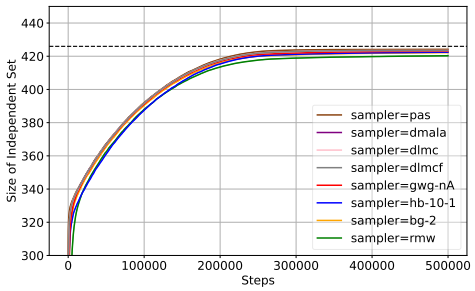


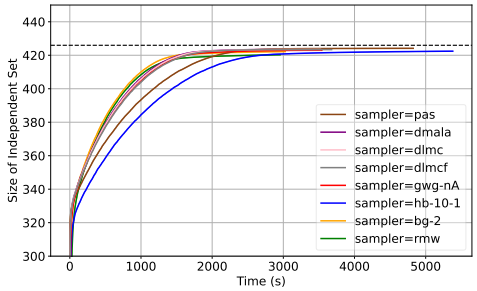
ER[800-800-0.25]





SATLIB





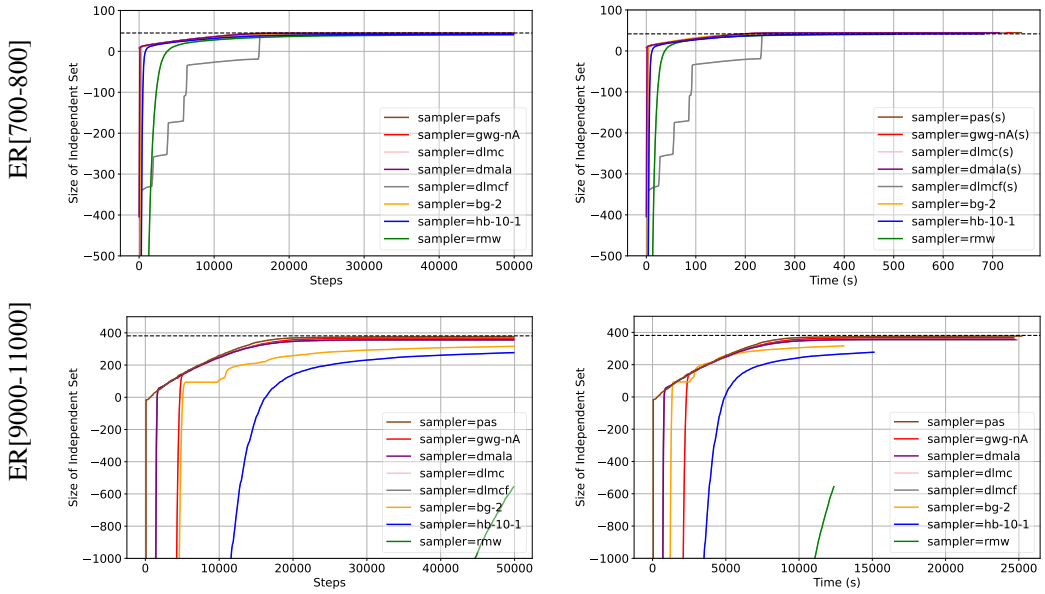


Figure 12: Solving progress on MIS

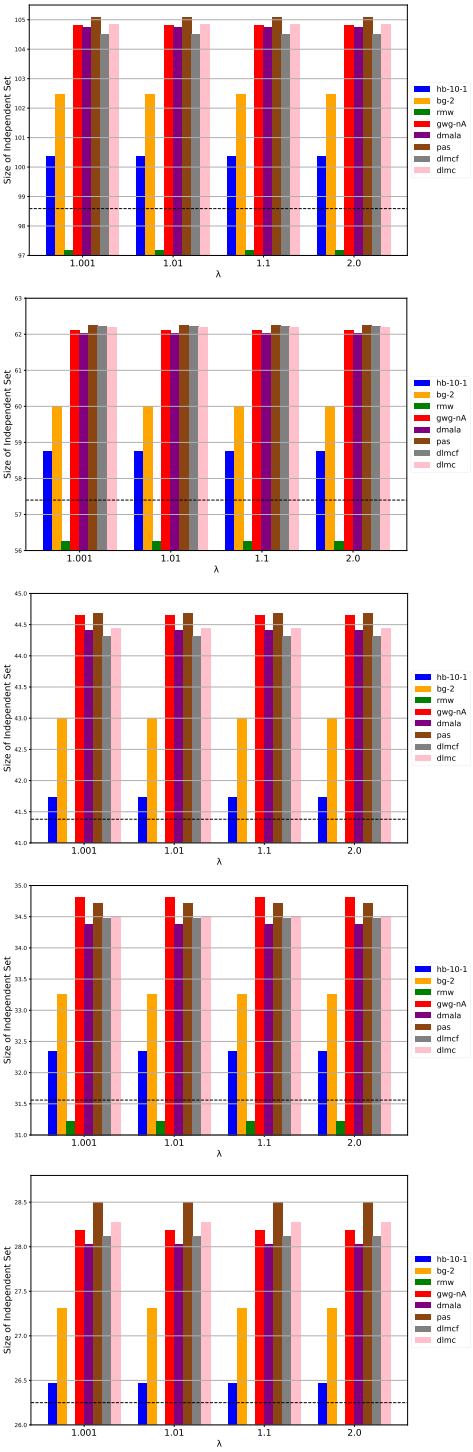


Figure 13: Reuslts on MIS: effect of penalty coefficient. (top)-(bottom) ER-[700-800] with density {0.05, 0.10, 0.15, 0.20, 0.25}. The dashed line represents the best result obtained by running Gurobi for 1 hour.

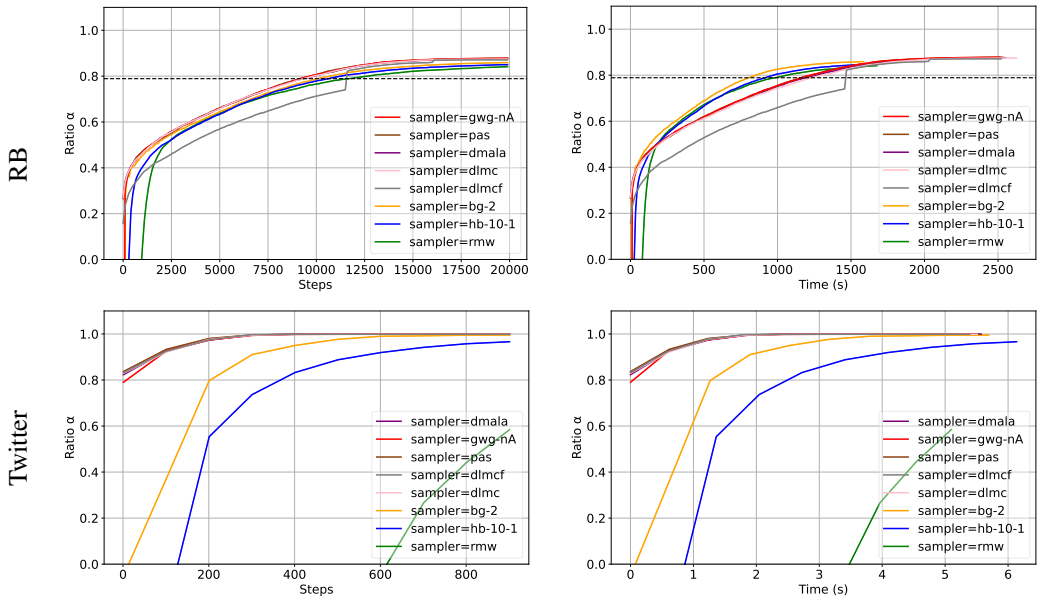
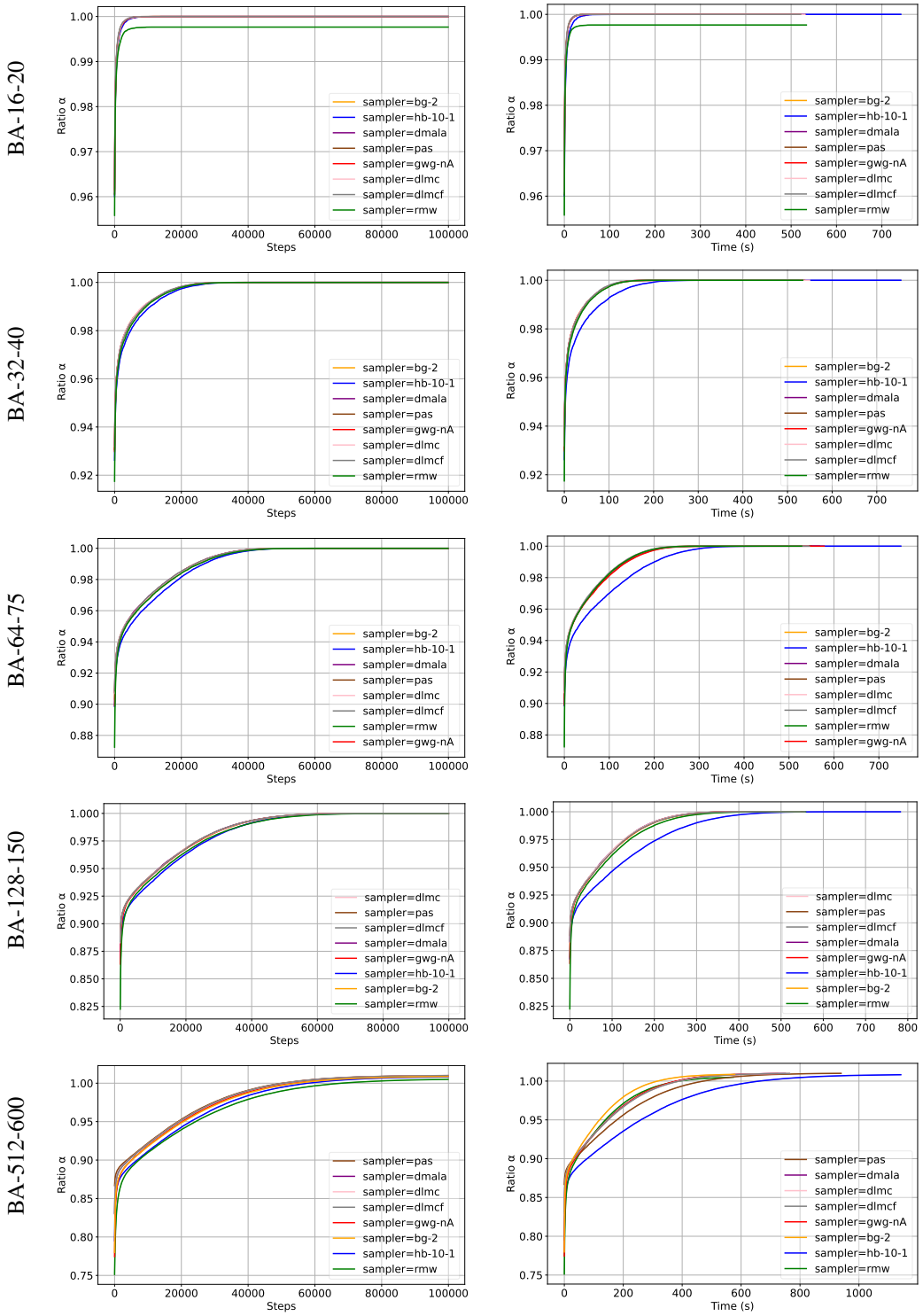


Figure 14: Solving progress on Max Clique


BA-16-20
BA-32-40
BA-64-75
BA-128-150
BA-512-600
Ratio α
Steps
Time (s)
sampler=bg-2
sampler=hb-10-1
sampler=dmala
sampler=pas
sampler=gwg-nA
sampler=dlmc
sampler=dlmcf
sampler=rmw

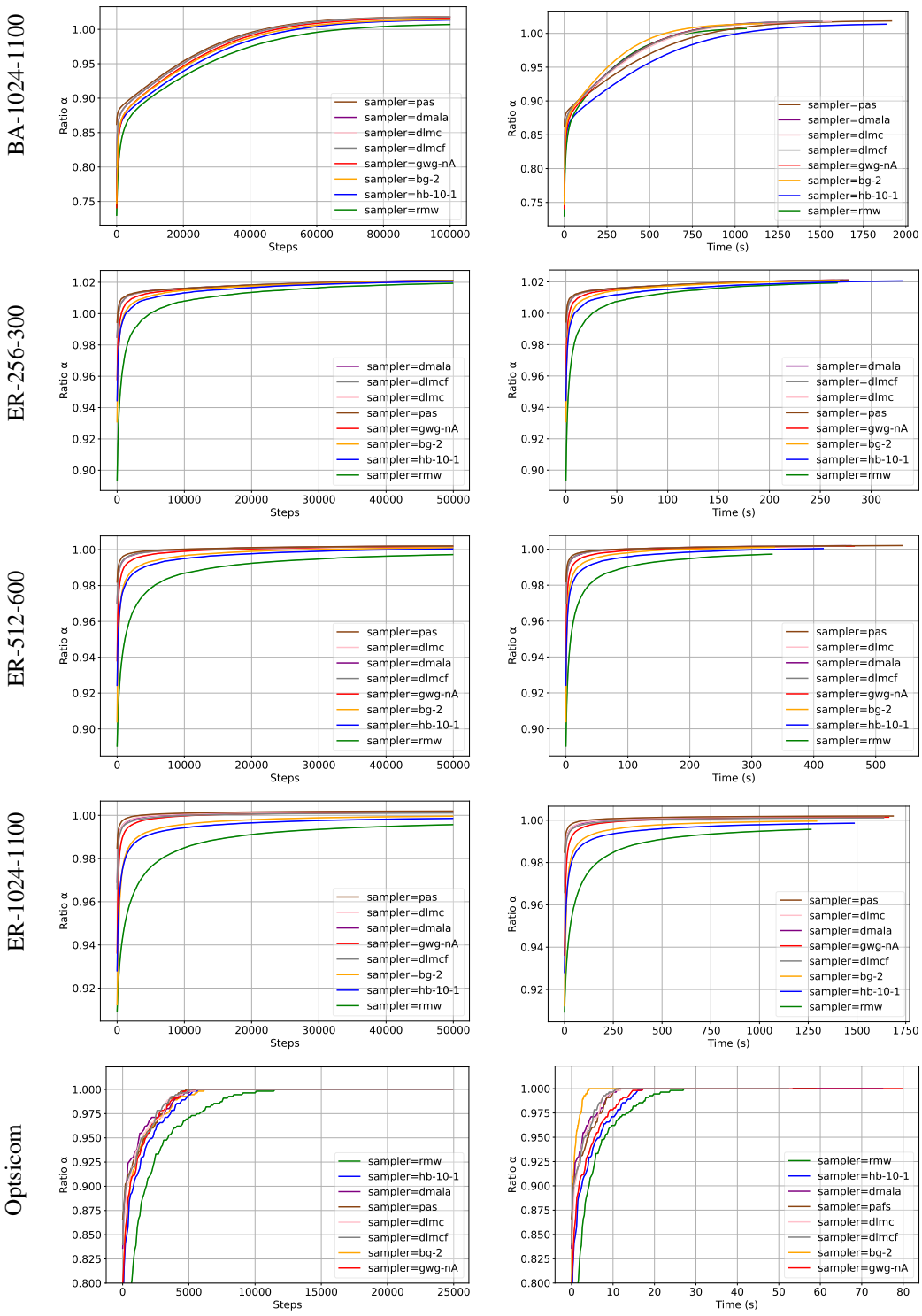


Figure 15: Maxcut

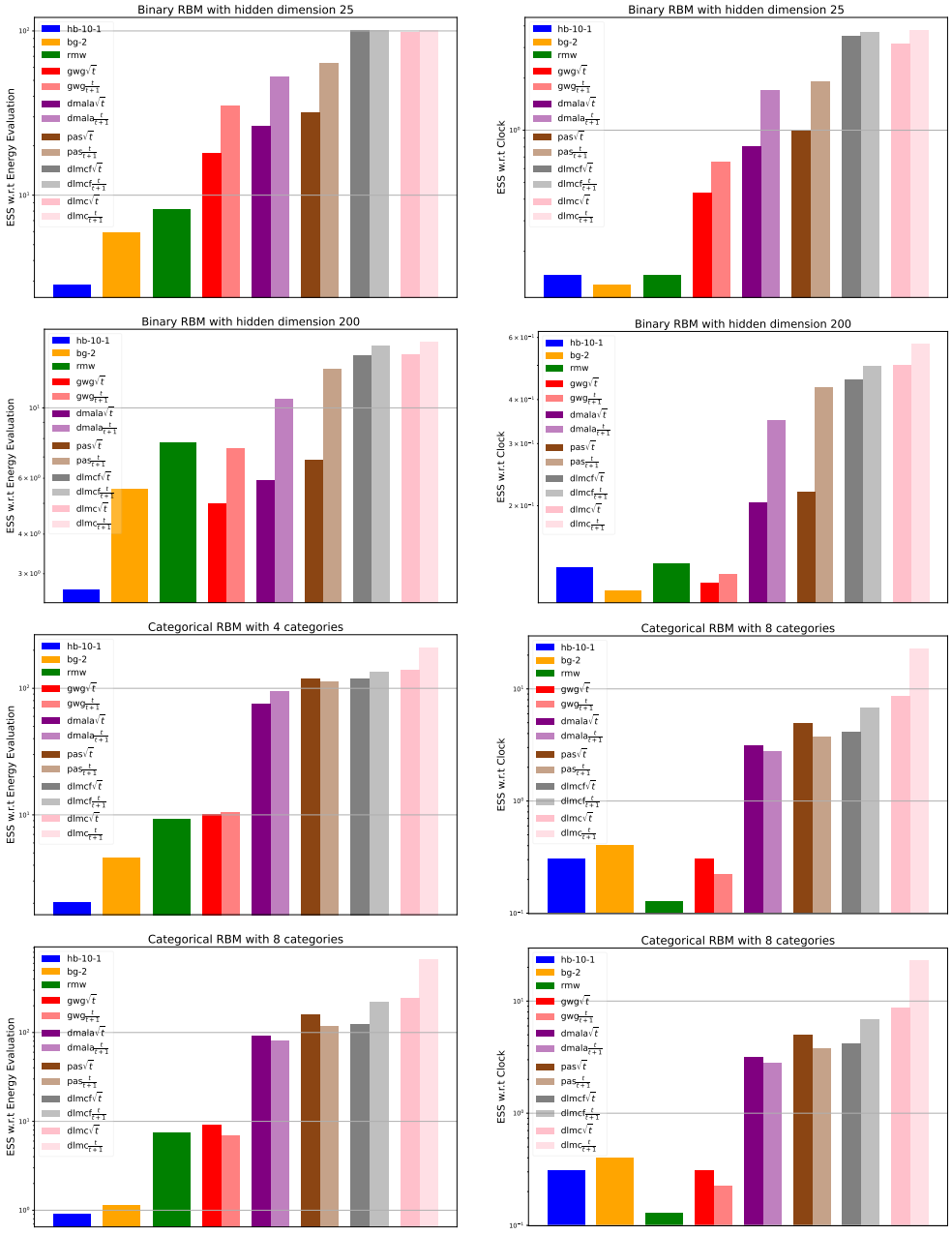
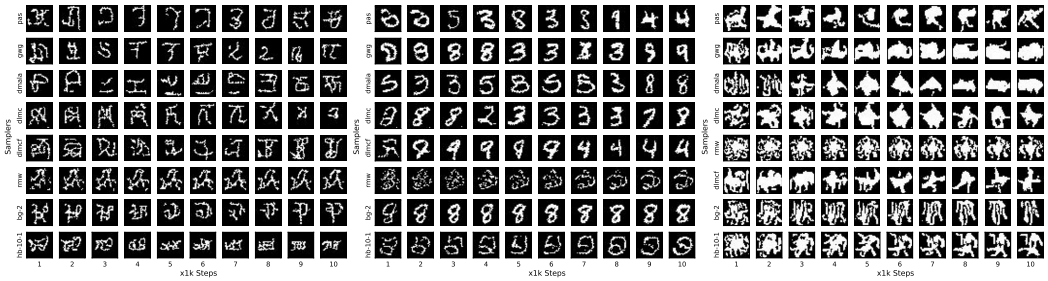


Figure 16: Results on RBMs



Omniglot MNIST Caltech

Figure 17: Resnet EBM