DISCS: A Benchmark for Discrete Sampling

Anonymous Author(s)
Affiliation
Address
email

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 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.

Submitted to ICML 2023 Workshop: Sampling and Optimization in Discrete Space. Do not distribute.

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

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

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 $x^{(0)}, x^{(1)}, \ldots$ 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 $\sigma_i \sim \mathcal{N}(0, 2.25)$ 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.

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

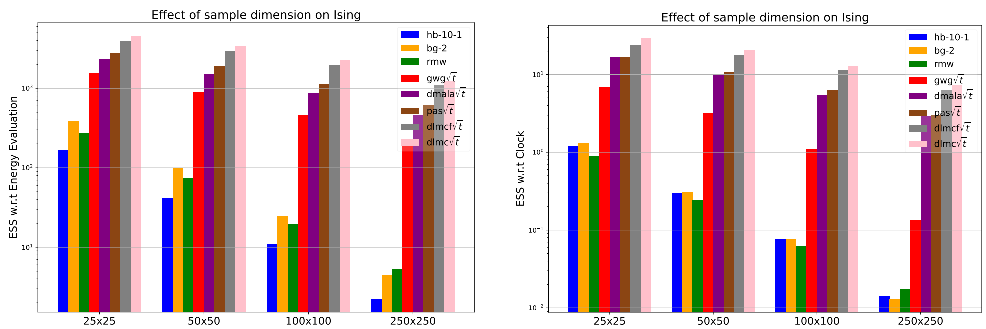


Figure 1: Results on Ising model with different dimensions

- 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.

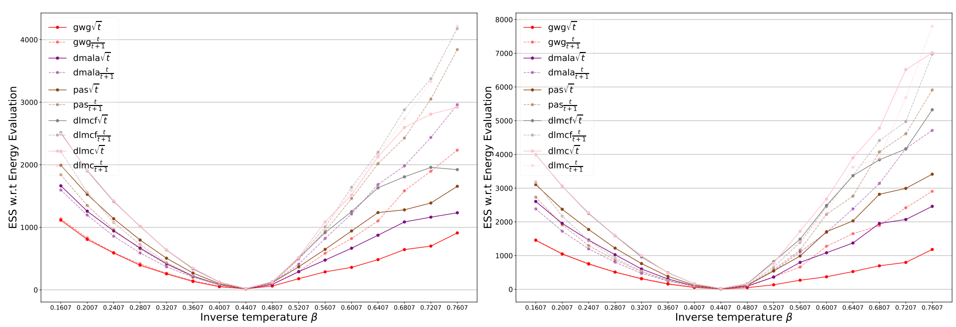


Figure 2: Performance of locally balanced samplers with different types of weight functions v.s temperature on: (left) 50×50 Ising model, (right) 100×100 Ising model

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.

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:

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

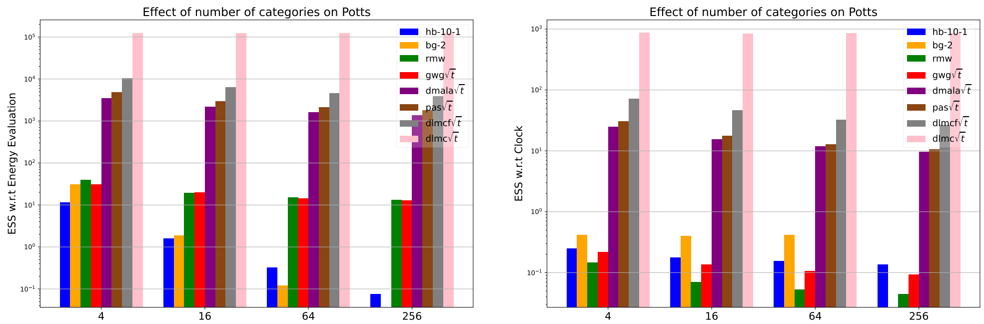


Figure 3: Results of Potts models with different number of categories

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.

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.

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.

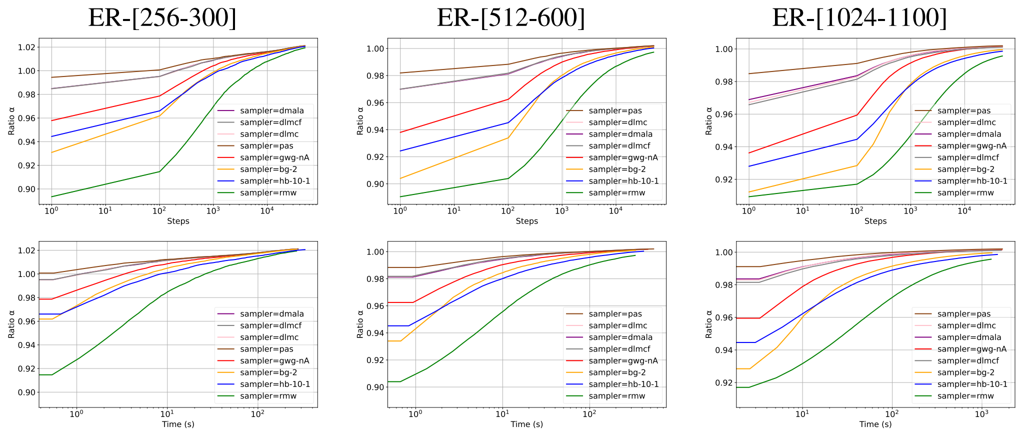


Figure 4: Results for MAXCUT on ER graphs. The ratio is computed by dividing the optimal cut size obtained from running Gurobi for 1 hour. (top) ratio with respect to number of M-H steps, (bottom) ratio with respect to running time.

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

For the models that are relatively simple, for example, Restricted Boltzmann Machine (RBM) trained on MNIST (LeCun, 1998) and fashion-MNIST (Xiao et al., 2017), 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.

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.

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

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

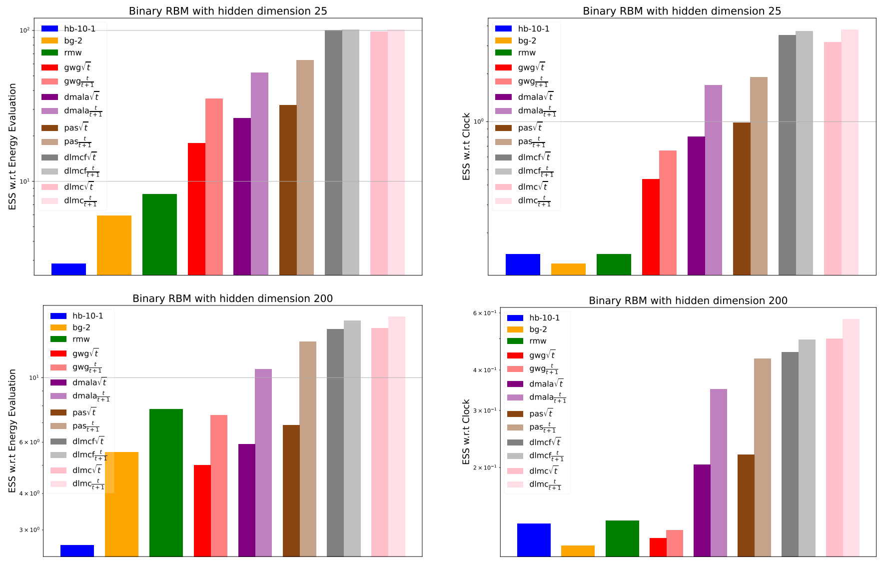
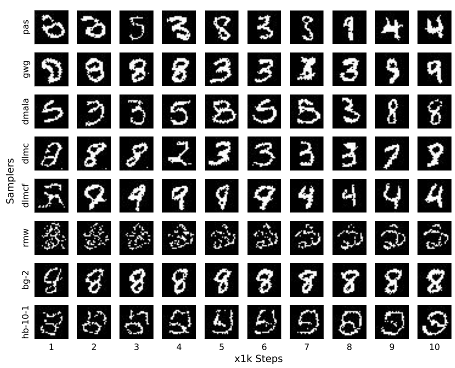


Figure 5: Results on RBMs trained on MNIST dataset. (top) RBM with 25 binary hidden variables, (bottom) RBM with 200 binary hidden variables



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

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.

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

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

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.

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.

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.

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. *arXiv preprint arXiv:1708.07747*, 2017.

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.

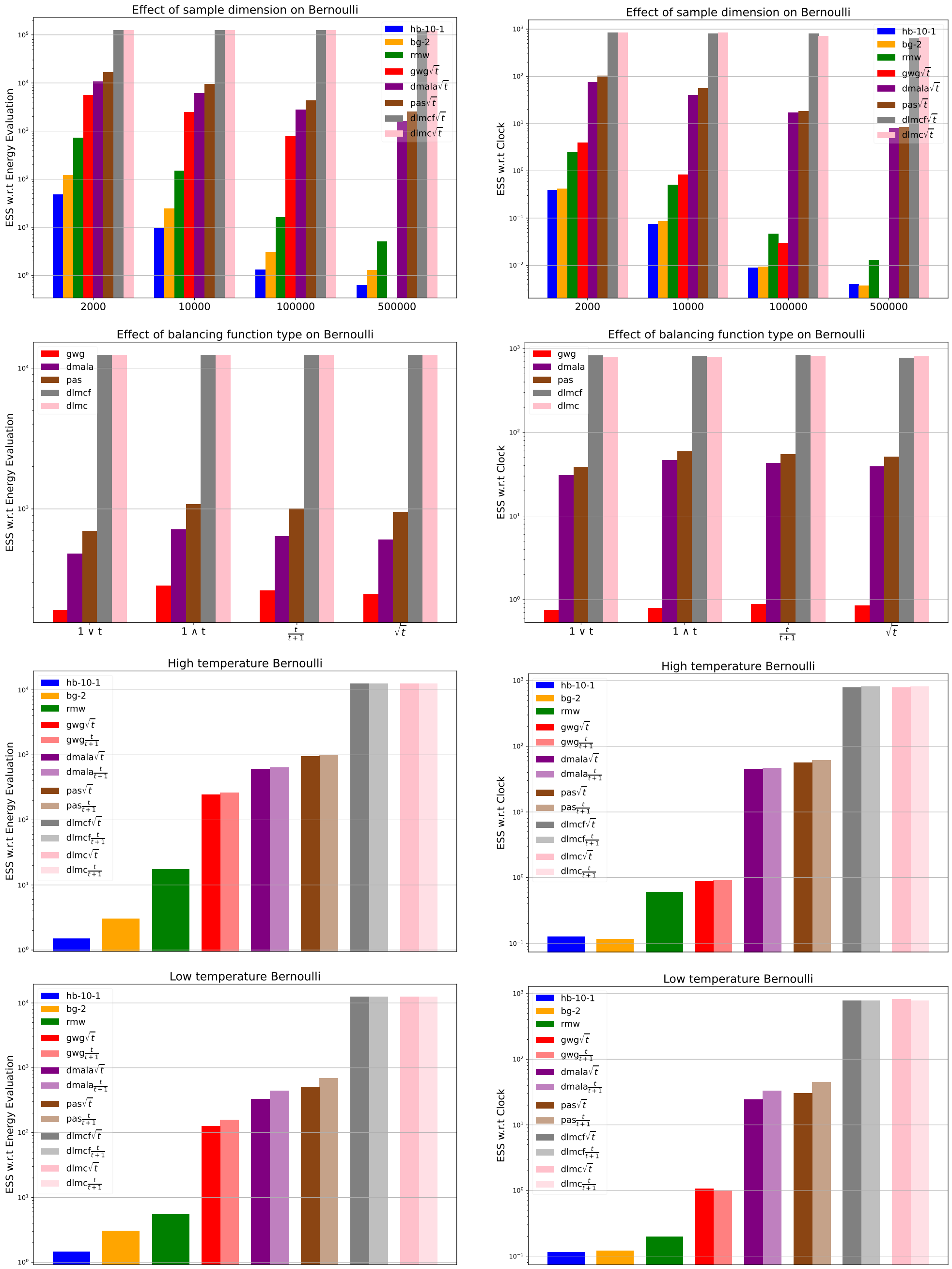


Figure 7: Bernoulli

A Appendix

A.1 Put to Appendix

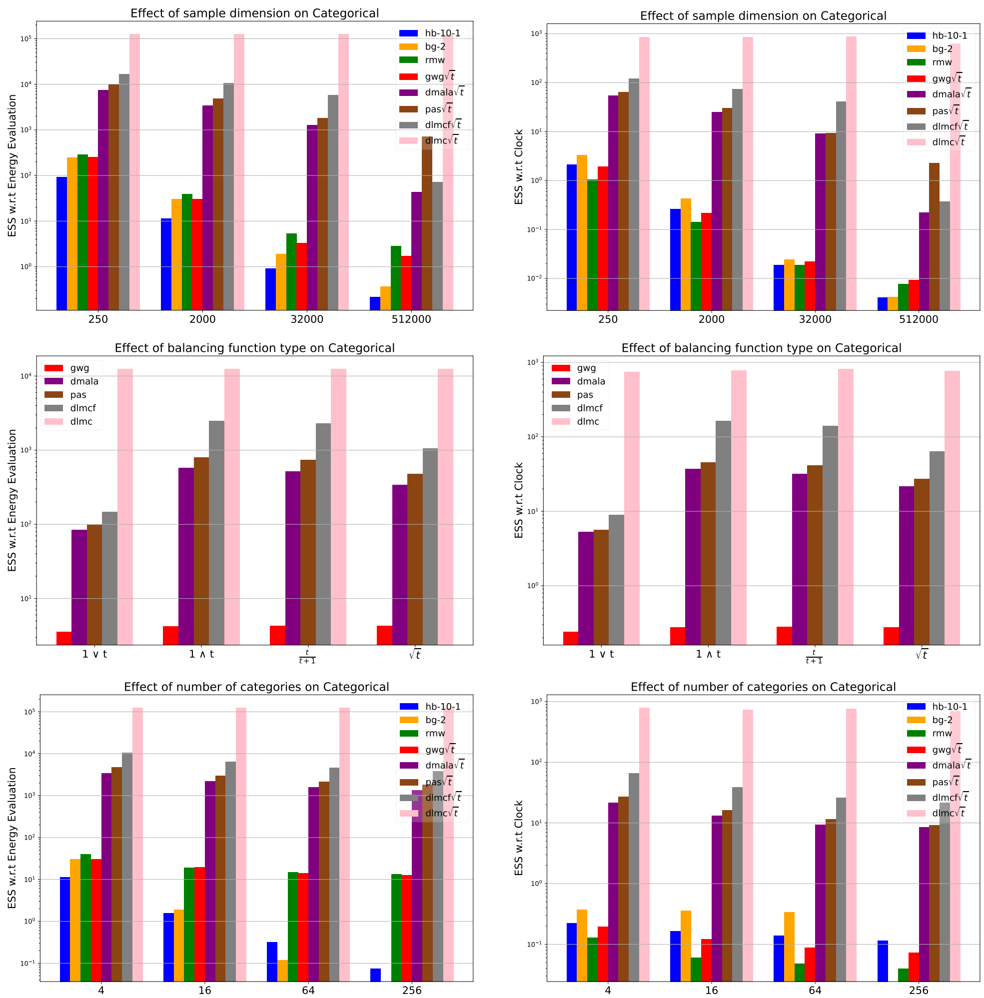


Figure 8: Categorical

Table 3: 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)	371.284	377.306	374.813	391.639	396.169	571.651	945.267	165.510	208.001	744.191	37.673
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)	258.592	269.129	275.041	276.931	265.860	289.496	578.785	134.558	168.507	647.610	8.525
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)	267.107	267.307	264.320	279.304	270.651	287.389	532.926	133.536	166.701	633.315	29.480
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)	261.047	265.713	289.458	275.961	272.817	362.360	713.788	132.10	233.100	833.010	40.062
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)	265.716	269.469	284.112	274.513	272.284	375.455	745.436	138.927	230.589	821.567	26.754
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)	259.921	269.407	275.017	275.289	290.025	470.204	958.977	146.716	465.481	3400.855	29.607
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)	260.800	263.145	272.938	278.782	266.559	382.859	755.190	136.420	226.126	819.769	26.276
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)	265.501	275.059	271.643	272.305	271.338	382.552	782.099	135.631	225.540	821.111	26.684

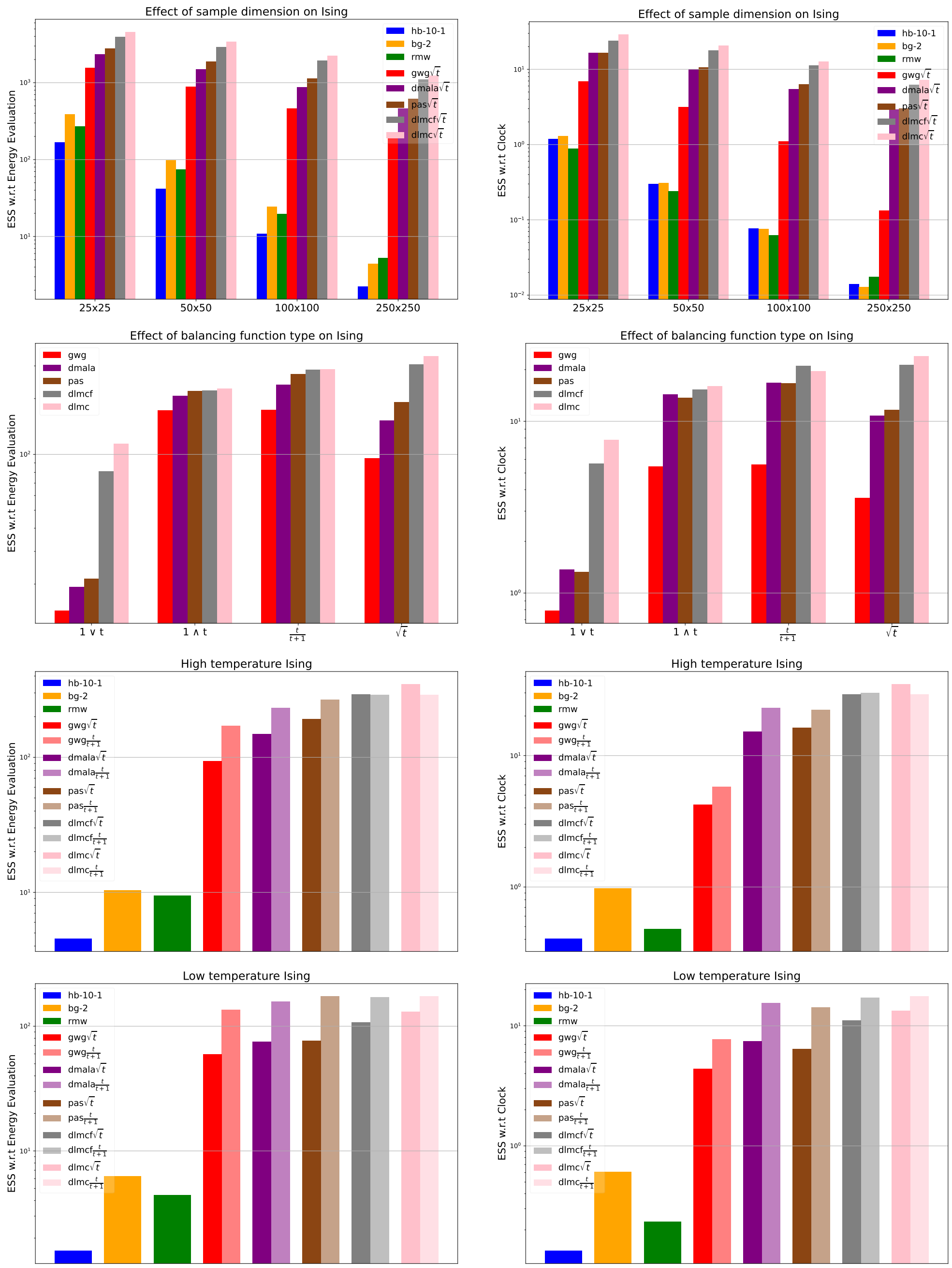
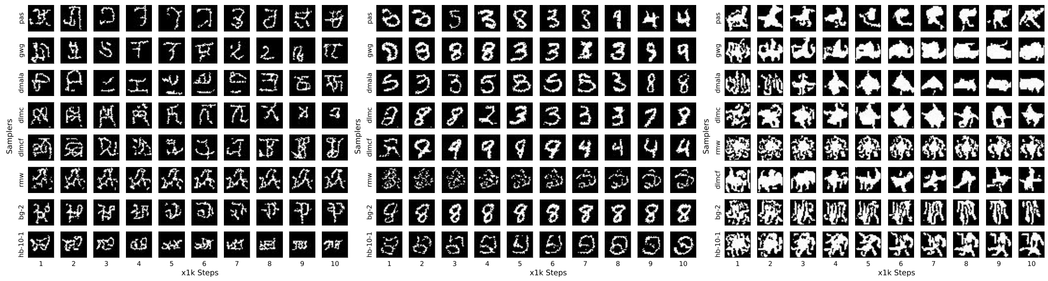


Figure 9: Ising



Omniglot MNIST Caltech

Figure 10: EBM

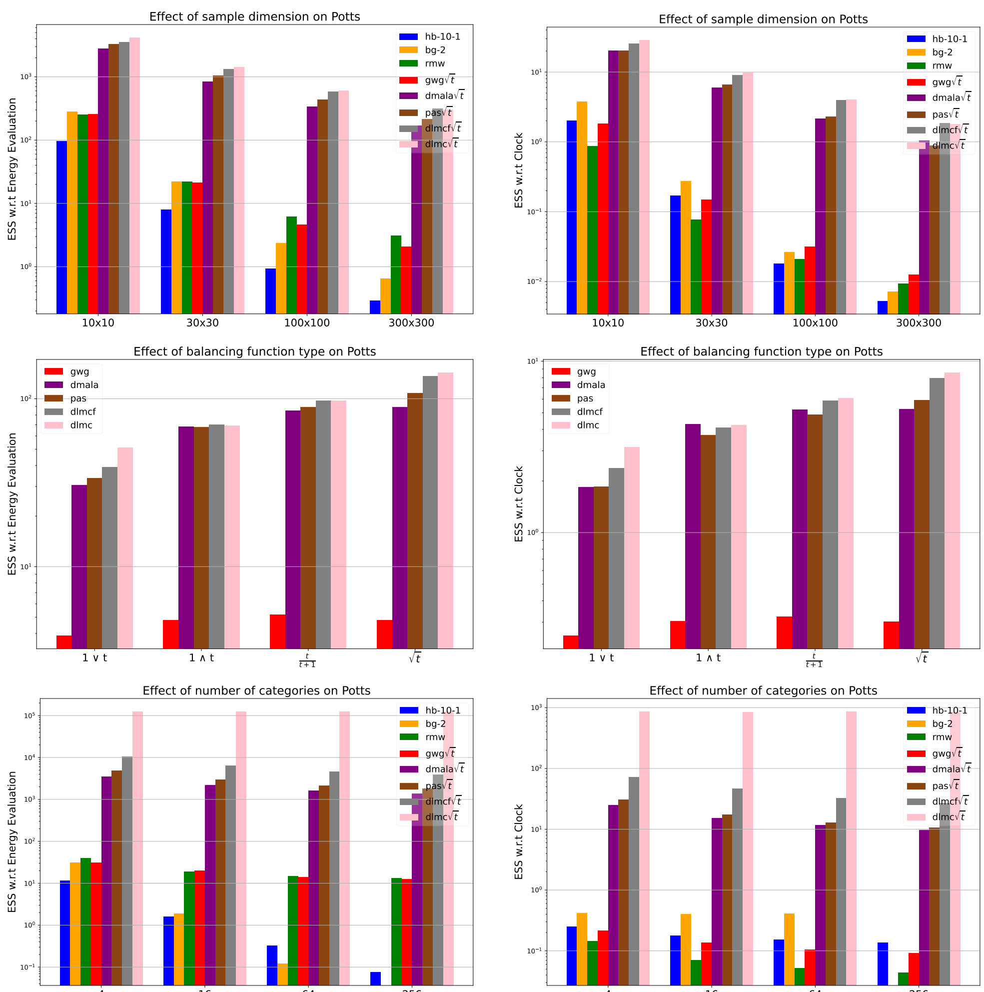


Figure 11: Potts

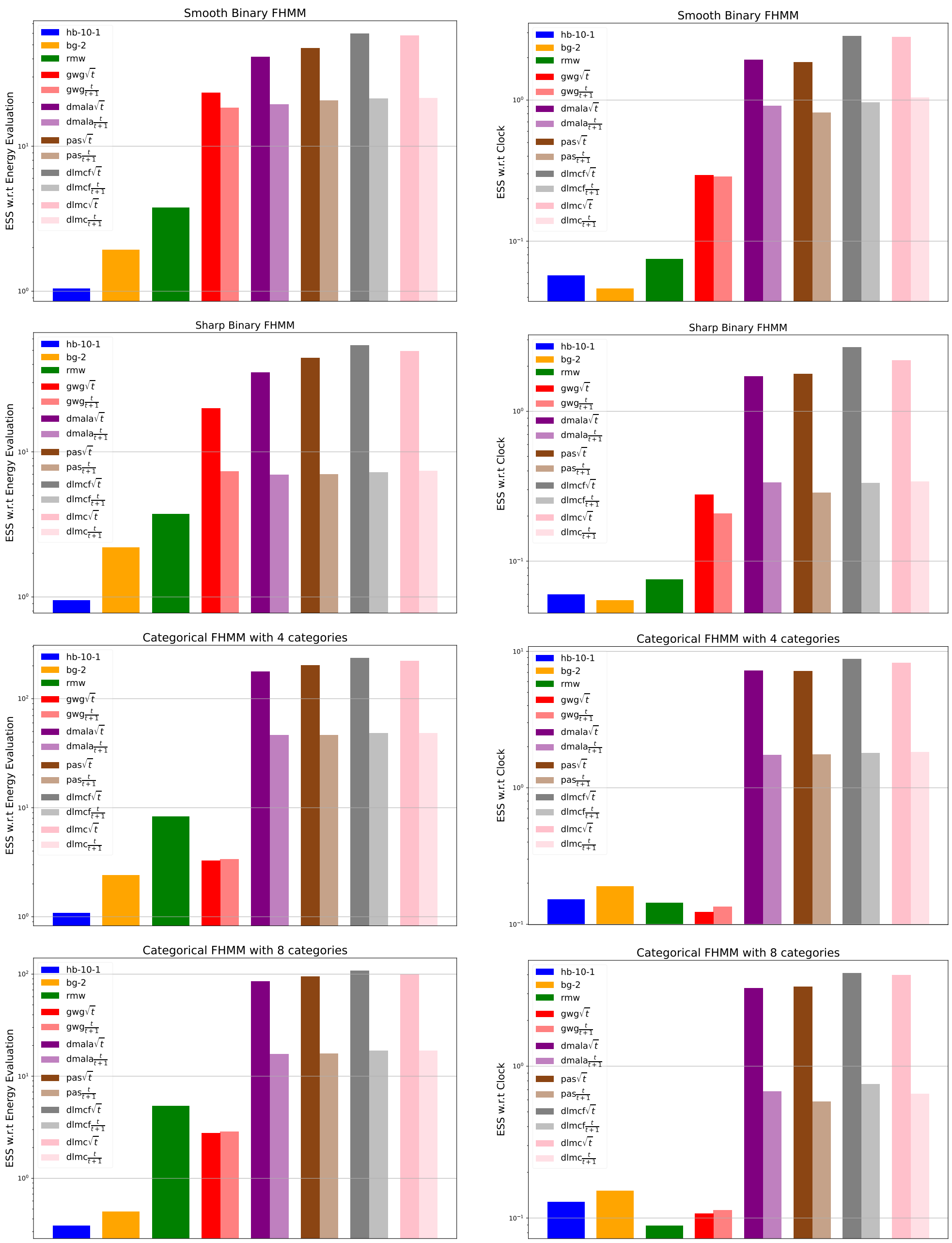


Figure 12: FHMM

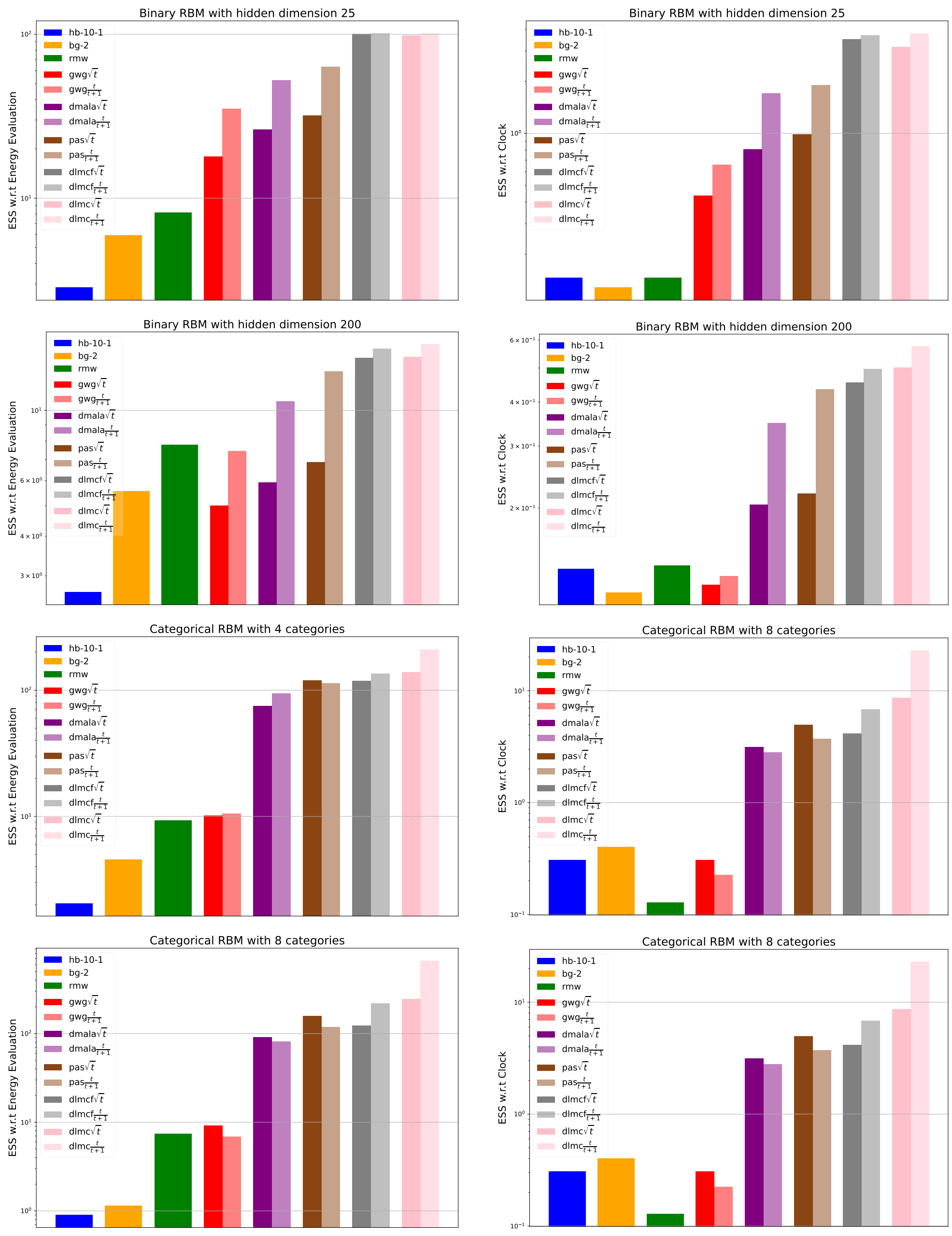
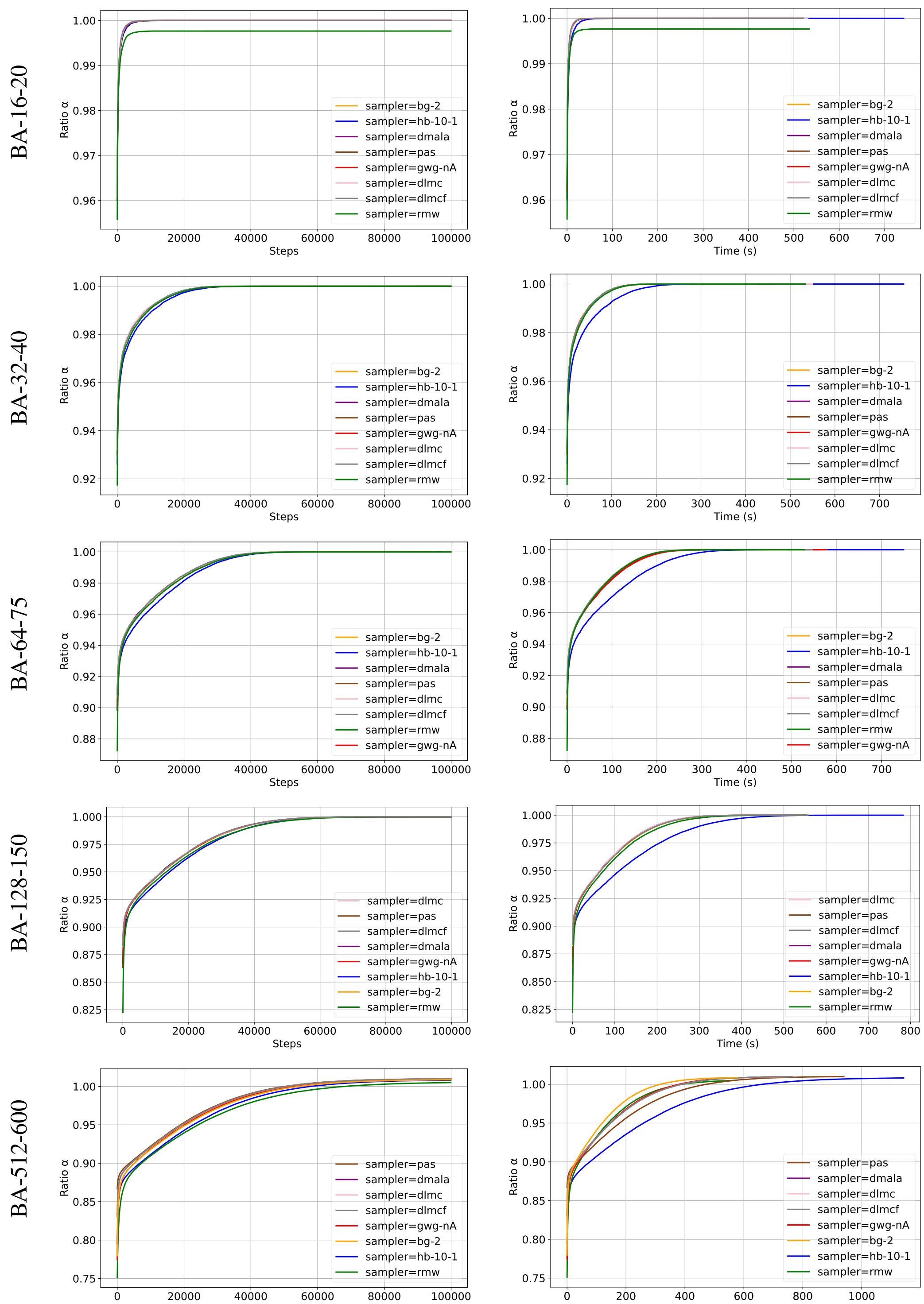


Figure 13: rbm


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

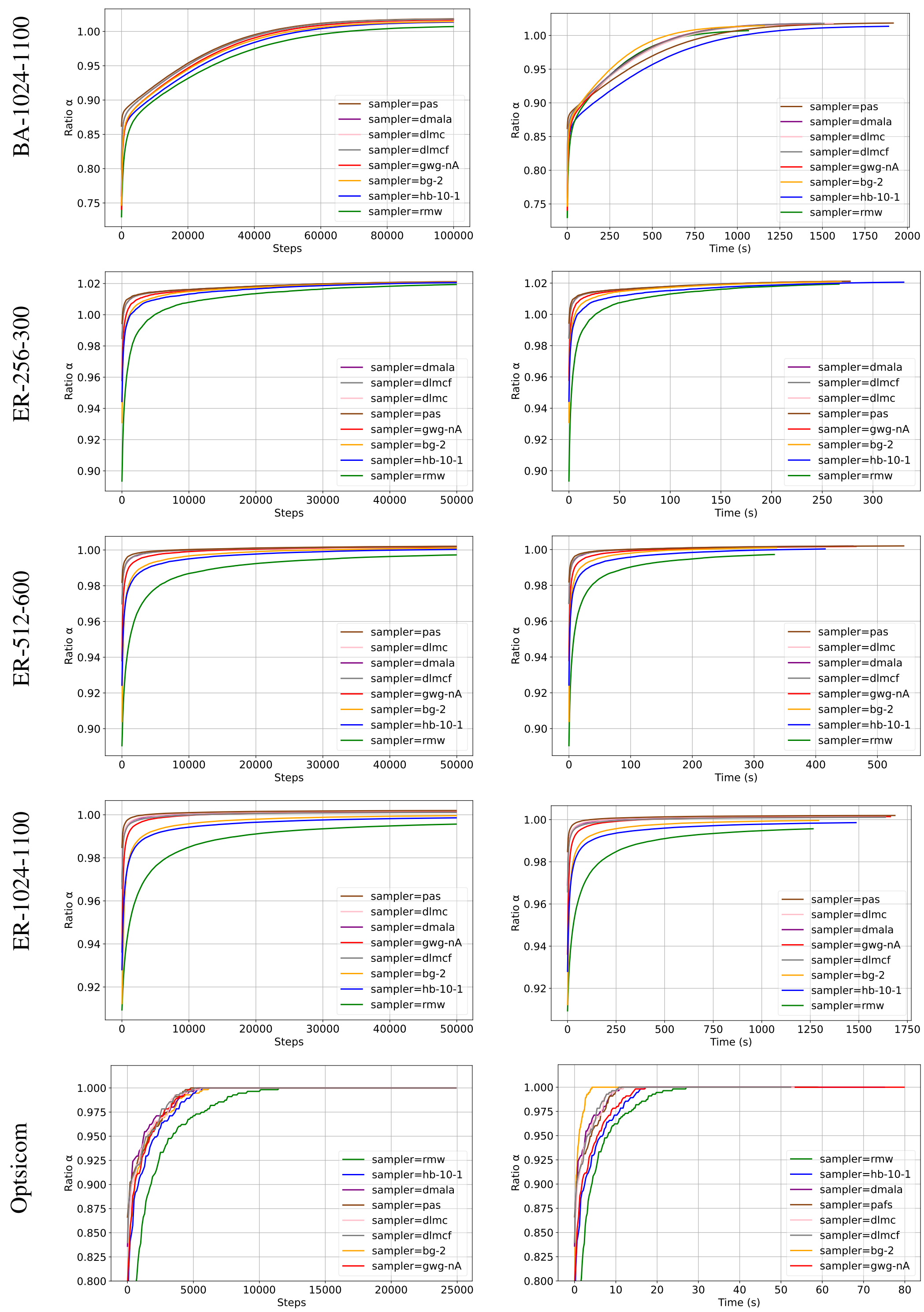


Figure 14: MAXCUT

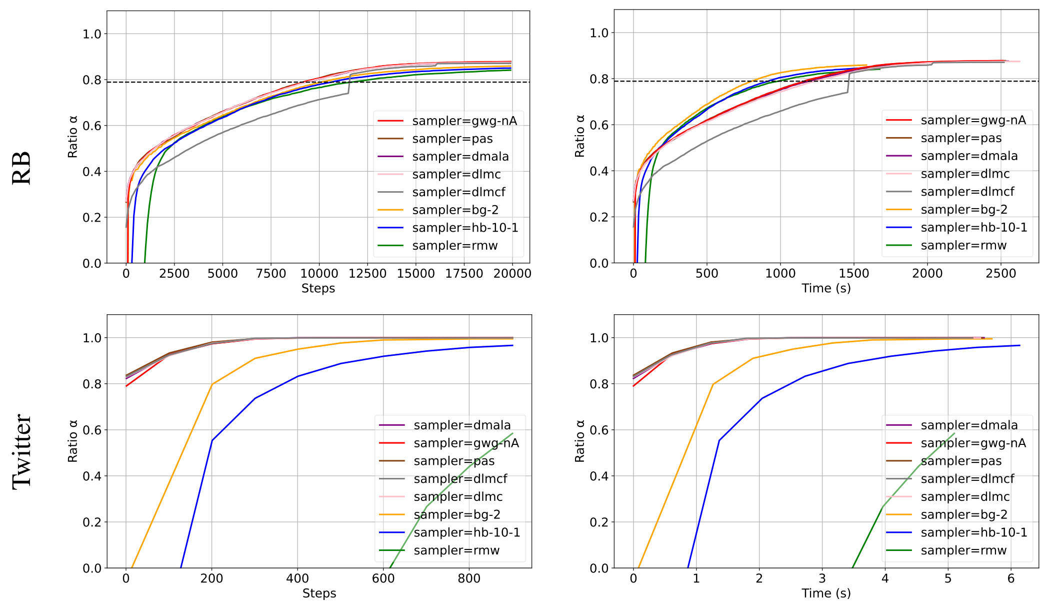
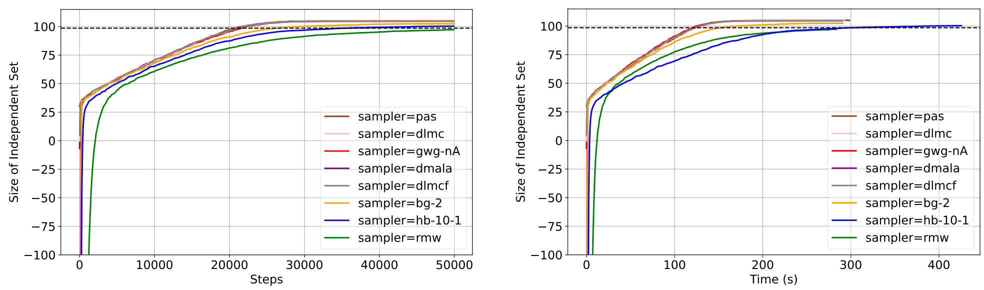


Figure 15: maxclique

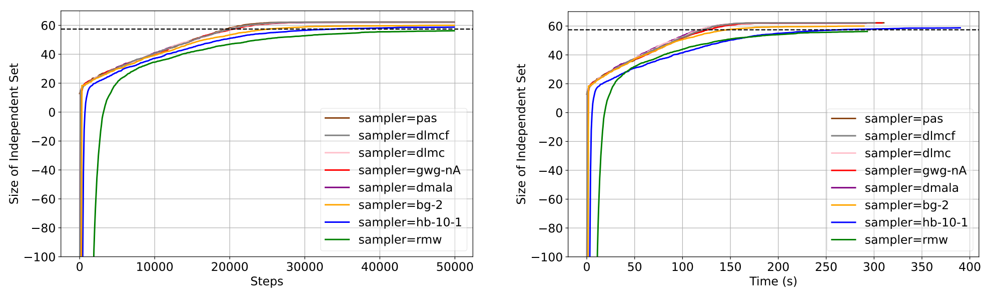
Table 4: MIS.

Sampler	Graphs Density	ER[700-800] 0.05	0.10	0.15	0.20	0.25	ER[9000-11000] 0.15	SATLIB
HB-10-1	Size	100.374	58.750	41.812	32.344	26.469	277.149	434.804
	Time(s)	213.092	377.306	342.295	207.034	214.940	7569.712	2063.689
BG-2	Size	102.468	60.000	42.820	32.250	27.312	316.170	434.545
	Time(s)	145.713	195.405	281.493	147.512	144.054	6539.562	1477.161
RMW	Size	97.186	56.249	40.429	31.219	25.594	-555.674	432.746
	Time(s)	142.046	145.021	249.789	148.570	140.886	6200.869	1468.328
GWG-nA	Size	104.812	62.125	44.383	34.812	28.187	367.310	435.419
	Time(s)	139.442	146.758	368.836	151.717	155.275	12349.148	1488.152
DMALA	Size	104.750	62.031	44.195	34.375	28.031	357.058	436.152
	Time(s)	145.635	154.437	357.307	148.924	149.366	12384.69	1494.575
PAS	Size	105.062	62.250	44.570	34.719	28.500	377.123	436.644
	Time(s)	149.502	155.382	379.686	149.785	154.238	12621.083	1517.682
DLMCF	Size	104.450	62.219	44.078	34.469	28.125	354.121	435.894
	Time(s)	145.683	150.777	363.143	151.334	150.206	12446.108	1486.004
DLMC	Size	104.844	62.187	44.273	34.500	28.281	355.058	436.046
	Time(s)	146.617	147.487	362.663	147.344	149.942	12488.156	1428.965

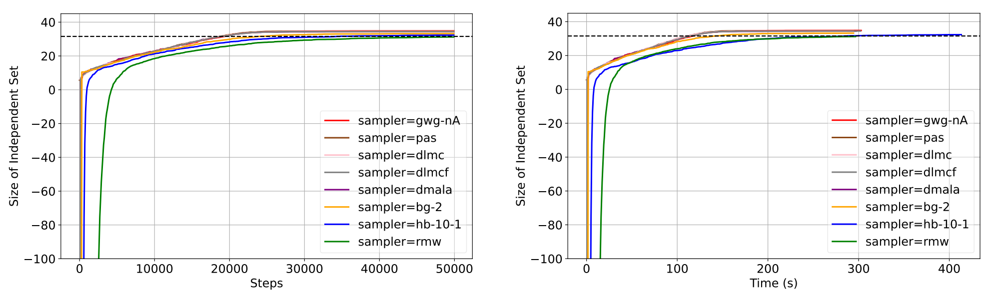
ER[800-800-0.05]



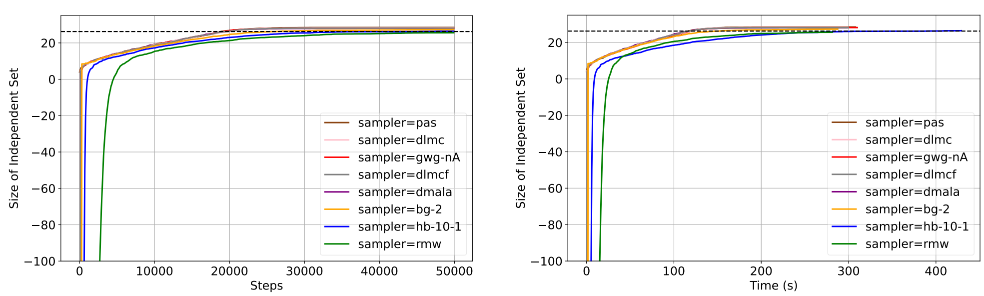
ER[800-800-0.10]



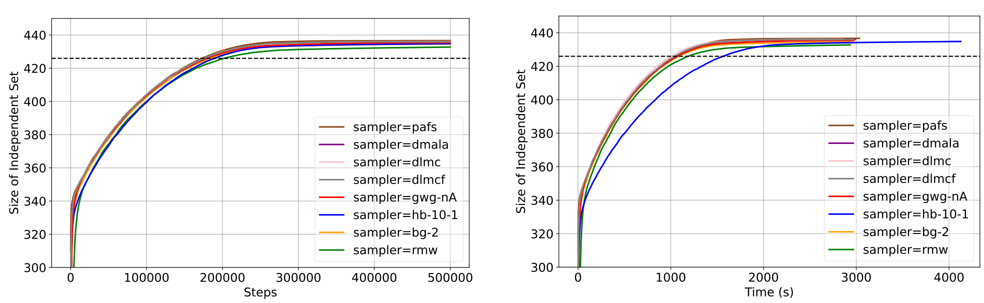
ER[800-800-0.20]



ER[800-800-0.25]



SATLIB



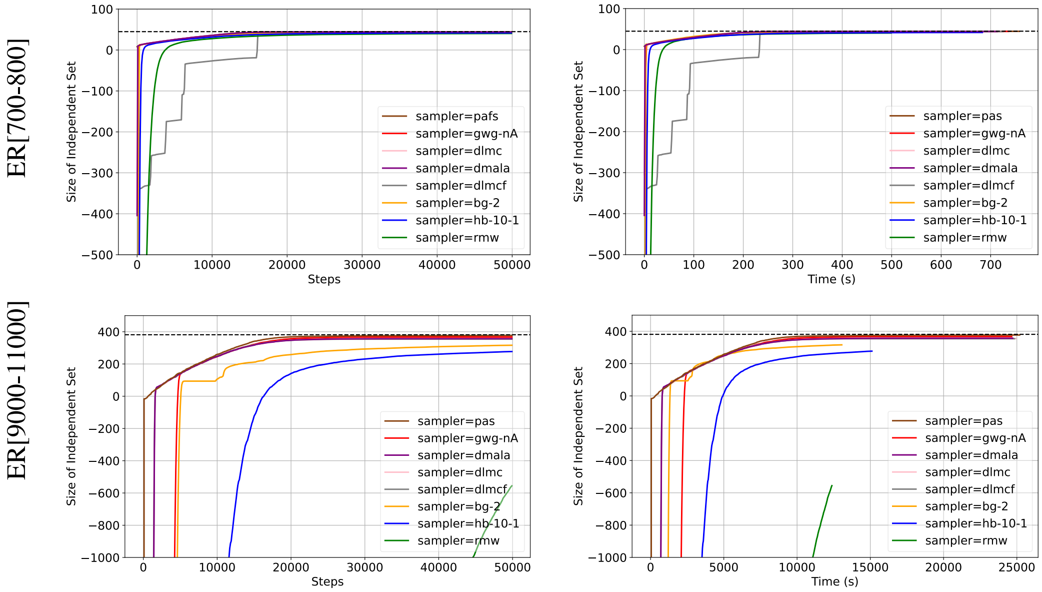


Figure 16: mis

Table 5: MAXCLIQUE.

Sampler	Results	RB	TWITTER
HB-10-1	Ratio α	0.850	0.966
	Time(s)	862.447	3.408
BG-2	Ratio α	0.859	0.995
	Time(s)	796.404	3.163
RMW	Ratio α	0.841	0.584
	Time(s)	841.698	2.832
GWG-nA	Ratio α	0.878	0.999
	Time(s)	1262.900	3.016
DMALA	Ratio α	0.876	0.999
	Time(s)	1280.807	3.095
PAS	Ratio α	0.878	0.999
	Time(s)	1271.269	3.090
DLMCF	Ratio α	0.871	0.999
	Time(s)	1266.417	2.994
DLMC	Ratio α	0.875	0.999
	Time(s)	1319.794	3.062

Table 6: 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

Table 7: 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