Oops I Took A Gradient: Scalable Sampling for Discrete Distributions

Will Grathwohl [1 2] **Kevin Swersky** [2] **Milad Hashemi** [2] **David Duvenaud** [1 2] **Chris J. Maddison** [1]

Abstract

We propose a general and scalable approximate sampling strategy for probabilistic models with discrete variables. Our approach uses gradients of the likelihood function with respect to its discrete inputs to propose updates in a Metropolis-Hastings sampler. We show empirically that this approach outperforms generic samplers in a number of difficult settings including Ising models, Potts models, restricted Boltzmann machines, and factorial hidden Markov models. We also demonstrate the use of our improved sampler for training deep energy-based models (EBM) on high dimensional discrete data. This approach outperforms variational auto-encoders and existing energy-based models. Finally, we give bounds showing that our approach is near-optimal in the class of samplers which propose local updates.

1. Introduction

Discrete structure is everywhere in the real world, from text to genome sequences. The scientific community is building increasingly complex models for this discrete data, increasing the need for methods to sample from discrete distributions. Sampling from a discrete distribution may seem like a simpler task than sampling from a continuous one: even a one-dimensional continuous distribution can have an uncountable infinity of outcomes, whereas a discrete distribution is at most countable. However, most common continuous distributions have some kind of simplifying structure, such as differentiable densities, which can be exploited to speed up sampling and inference.

Of course, many discrete distributions have structure as well. Notably, discrete distributions over combinatorial spaces often have some kind of block independence structure among their variables. Although this can be used to speed up sampling and inference, it may be difficult to detect such structure automatically. Typically, users need to know this structure a priori and must hard-code it into an algorithm to speed up sampling.

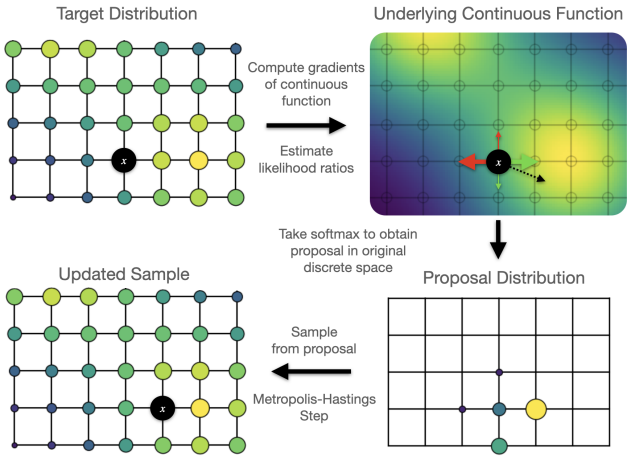

Figure 1. Our approach visualized. Often discrete distributions are defined by continuous functions whose input is restricted to a discrete set; here R^2 restricted to $\mathcal{Z}^2$. We use a Taylor series computed on the underlying continuous function to estimate likelihood ratios of making discrete moves; here ± 1 in either direction. These estimated likelihood ratios are used to inform a proposal distribution over moves in the original discrete space.

In search for a unifying structure, we notice that many discrete distributions are written (and implemented) as differentiable functions of real-valued inputs. The discrete structure is created by restricting the continuous inputs to a discrete subset of their domain. In this paper, we use the gradients of these underlying continuous functions to inform proposal distributions for MCMC sampling in large discrete probabilistic models. This new family of gradient-informed proposals for MCMC may be seen as a class of adaptive Gibbs samplers or a fast approximation to locally-informed proposals (Umrigar, 1993; Liu, 1996; Zanella, 2020).

As we show, this gradient information is cheaply available for many discrete distributions and can lead to orders of magnitude improvements in sampling efficiency. In some cases, it even outperforms samplers that exploit hard-coded independence structure.

[1]University of Toronto and Vector Institute [2]Google Research, Brain Team. Correspondence to: Will Grathwohl <wgrathwohl@cs.toronto.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

We apply these sampling techniques to improve parameter inference in discrete energy-based models such as Ising and Potts models. We also find that our method is particularly well-suited to sample from unnormalized discrete distributions parameterized by deep neural networks. The improved efficiency of our sampler allows us to apply many techniques from large-scale continuous EBMs to successfully train deep EBMs for discrete data. These models are simple to define, flexible, and outperform baseline variational auto-encoders (Kingma & Welling, 2013) and existing energy-based models.

2. Background

In this work we are concerned with sampling from unnormalized distributions over discrete data

$$\log p(x) = f(x) - \log Z$$

where $f(x)$ is the unnormalized log-probability of x and $Z = \sum_x e^{f(x)}$ is the normalizing constant which we assume is unknown. We restrict our study to D-dimensional binary $x \in \{0, 1\}^D$ and categorical $x \in \{0, 1, \dots, K\}^D$ data as all finite-dimensional discrete distributions can be embedded in this way. Throughout the paper we assume all categorical variables are one-hot encoded.

2.1. Gibbs Sampling

Gibbs sampling is perhaps the simplest and most generic method for sampling from discrete distributions. At each step, the sampler partitions the dimensions of an input x into two groups x_u and x_{-u} where u is a subset of the D dimensions of x and $-u$ is its complement. The next sample in the chain is created by updating x_u to be a sample from $p(x_u|x_{-u})$, the conditional distribution of the chosen dimensions, given all others. In some distributions, block-independence structure exists, making certain partitions easy to sample from and update in parallel.

In the worst case, if such structure does not exist, we can let $x_u = x_i$, simply the i-th dimension of x. In this setting $p(x_i|x_{-i})$ is simply a one-dimensional categorical distribution over K possible outcomes, which requires K evaluations of the unnormalized log-density function. We typically fix an ordering of the dimensions and iterate through this ordering to ensure that each dimension is updated.

While simple, updating one dimension at a time can be problematic for high-dimensional data. Consider for example a binary model of the MNIST dataset. Almost all dimensions will represent the background and if chosen for a Gibbs update, they will almost certainly not change. Similarly, the pixels on the interior of a digit will also not change. This amounts to wasted computation every time we propose a dimension, which does not change. Ideally, if we could bias our partition choice to dimensions which are likely to change, we could build a more efficient Gibbs sampler.

Consider a similar sampler for the binary case; Metropolis-Hastings with the proposal distribution $q(x'|x) = \sum_i q(x'|x, i)q(i)$ where $q(i)$ is a distribution over indices $i \in \{1, \dots, D\}$ and $q(x'|x, i) = \delta(x' = x_{-i})$, where $\delta(\cdot)$ is the Dirac-delta distribution. With this sampler we will first sample an index $i \sim q(i)$, then flip the i-th bit of x to obtain x' and accept this proposed update with probability

$$\min\left\{\exp(f(x') - f(x))\frac{q(x|x')}{q(x'|x)}, 1\right\}. \tag{1}$$

This approach can lead to considerable performance improvements if $q(i)$ is biased towards dimensions which are more likely to flip. To this end, considerable work has been done to use prior sampling data to adaptively update the $q(i)$ while maintaining the validity of MCMC sampling (Łatuszyński et al., 2013; Richardson et al., 2010).

Revisiting our MNIST example, we can see the pixels most likely to change are those at the edge of a digit. Where this edge is varies considerably and depends on the entire input x. Thus, we can imagine an input dependent proposal $q(i|x)$ should be able to outperform an unconditional proposal $q(i)$. Our proposed approach does exactly that, using gradient information to inform a proposal over dimensions which leads to more efficient sampling.

2.2. Locally-Informed Proposals

A good Metropolis-Hastings proposal needs to balance the increase in the likelihood of the proposed point x' with the decrease in the probability of the reverse proposal $q(x|x')$. A natural strategy for increasing the likelihood is to use locally-informed proposals:

$$q_\tau(x'|x) \propto e^{\frac{1}{\tau}(f(x') - f(x))}\mathbf{1}(x' \in H(x)). \tag{2}$$

where $H(x)$ is the Hamming ball of some size around x and $\tau > 0$ is a temperature parameter. In this case, the proposal is simply a tempered Softmax over $f(x') - f(x)$ for all $x' \in H(x)$. If the temperature τ is too low, this proposal will aggressively optimize the local likelihood increase from $x \to x'$, but possibly collapse the reverse proposal probability $q_\tau(x|x')$. If the temperature τ is too high, this proposal may increase the reverse proposal probability $q_\tau(x|x')$, but ignore the local likelihood increase.

The temperature that balances both of these terms is $\tau = 2$ (Umrigar, 1993; Zanella, 2020). We include a derivation of this fact in Appendix A. In fact, Zanella (2020) showed that $q_2(x'|x)$ is in the optimal subclass of locally-informed proposals. Zanella (2020) also demonstrated that this can lead to large improvements in empirical performance per sam-

pling step compared to other generic samplers like Gibbs and the Hamming-Ball sampler (Titsias & Yau, 2017).

Unfortunately, while powerful, these locally-informed proposals requires us to compute $f(x') - f(x)$ for every $x' \in H(x)$. For D-dimensional data and a Hamming window size of 1, this requires $\mathcal{O}(D)$ evaluations of f which can become prohibitive as D grows. Our proposed approach reduces this to $\mathcal{O}(1)$ evaluations while incurring a minimal decrease in the efficiency of each sampling step.

3. Searching for Structure

For some distributions, structure exists which enables the local differences $f(x') - f(x)$ to be computed efficiently. This is not true in general, but even in settings where it is, bespoke derivations and implementations are required. Ideally, we could identify a structure that is ubiquitous across many interesting discrete distributions, can be exploited in a generic and generalizable way, and can allow us to accurately estimate the local differences.

To find such structure, we examine the functional form of the unnormalized log-probability for some common, diverse families of discrete distributions in Table 1.

Distribution	$\log p(x) + \log Z$
Categorical	$x^T\theta$
Poisson[1]	$x \log \lambda - \log \Gamma(x+1)$
HMM	$\sum_{t=1}^{T} x_{t+1}^T A x_t - \frac{(w^T x - y)^2}{2\sigma^2}$
RBM	$\sum_i \text{softplus}(Wx+b)_i + c^T x$
Ising	$x^T W x + b^T x$
Potts	$\sum_{i=1}^{L} h_i^T x_i + \sum_{i,j=1}^{L} x_i^T J_{ij} x_j$
Deep EBM	$f_\theta(x)$

Table 1. Unnormalized log-likelihoods of common discrete distributions. All are differentiable with respect to x.

The formulas in Table 1 are not only the standard way these distributions are written down, but they are also the standard way these distributions are implemented in probabilisitic programming frameworks.

The key insight here is that these are all continuous, differentiable functions accepting real-valued inputs, even though they are evaluated only on a discrete subset of their domain. We propose to exploit this structure and that gradients, in the form of Taylor-series approximations, can be used to efficiently estimate likelihood ratios between a given input x and other discrete states x'.

When we are dealing with D-dimensional binary data, we can estimate the likelihood ratios of flipping each bit with

$$\tilde{d}(x) = -(2x-1) \odot \nabla_x f(x) \tag{3}$$

where $\tilde{d}(x)_i \approx f(x_{-i}) - f(x)$ and x_{-i} is x with the i-th bit flipped. If we are dealing with D-dimensional categorical data we can estimate a similar quantity

$$\tilde{d}(x)_{ij} = \nabla_x f(x)_{ij} - x_i^T \nabla_x f(x)_i \tag{4}$$

where $\tilde{d}(x)_{ij}$ approximates the log-likelihood ratio of flipping the i-th dimension of x from its current value to the value j. Similar first-order approximations can easily be derived for larger window sizes as well with linear operators applied to the gradient of the log-probability function.

We note, there may be many different underlying continuous functions which correspond to a given discrete distribution. We later present theory (Theorem 1) that shows that we should choose the smoothest possible function.

4. Gibbs With Gradients

We now present our main algorithm. We use a Taylor-series (Equations 3, 4) to approximate the likelihood ratios within a local window of a point x. We use these estimated likelihood ratios to produce an approximation

$$q^\nabla(x'|x) \propto e^{\frac{\tilde{d}(x)}{2}} \mathbf{1}(x' \in H(x)) \tag{5}$$

to $q_2(x'|x)$ of Equation 2, which we use in the standard Metropolis-Hastings algorithm.

Our experiments focus on a simple and fast instance of this approach which only considers local moves inside of a Hamming window of size 1. For binary data, these window-1 proposals have an even simpler form since all $x' \in H(x)$ differ by only dimension. Proposing a move from x to x' is equivalent to choosing which dimension i to change. We can sample this from a categorical distribution over D choices:

$$q(i|x) = \text{Categorical}\left(\text{Softmax}\left(\frac{\tilde{d}(x)}{2}\right)\right) \tag{6}$$

Thus when x binary, to sample from $q^\nabla(x'|x)$, we simply sample which dimension to change $i \sim q(i|x)$, and then deterministically set $x' = \texttt{flipdim}(x, i)$. In this case, when x and x' differ only in dimension i, we have $q^\nabla(x'|x) = q(i|x)$ and $q^\nabla(x|x') = q(i|x')$.

Because of the relationship to Adaptive Gibbs, we call our sampler Gibbs-With-Gradients (GWG). Pseudo-code describing our sampler can be found in Algorithm 1. In the

[1]While we only present results in this paper for finite-dimensional discrete distributions, Gibbs-With-Gradients can be easily applied to (infinite) integer-valued distributions using a proposal window of $\{x-1, x+1\}$.

Algorithm 1 Gibbs With Gradients

Input: unnormalized log-prob $f(\cdot)$, current sample x
Compute $\tilde{d}(x)$ {Eq. 3 if binary, Eq. 4 if categorical.}
Compute $q(i|x) = \text{Categorical}\left(\text{Softmax}\left(\frac{\tilde{d}(x)}{2}\right)\right)$
Sample $i \sim q(i|x)$
$x' = \texttt{flipdim}(x, i)$
Compute $q(i|x') = \text{Categorical}\left(\text{Softmax}\left(\frac{\tilde{d}(x')}{2}\right)\right)$
Accept with probability:

$$\min\left(\exp(f(x') - f(x))\frac{q(i|x')}{q(i|x)}, 1\right)$$

categorical data setting, the proposal must choose not only which dimension to change, but also to what value. Thus, $q(i|x)$ in this setting is a $D(K-1)$-way Softmax.

We describe some simple extensions in Appendix D and code to replicate our experiments is available here.

4.1. Analyzing Approximations

Zanella (2020) proved that "locally-balanced" proposals, like $q_2(x'|x)$ in Equation 2, are the optimal locally-informed proposals for Metropolis-Hastings. In this section we show that, under smoothness assumptions on f, our methods are within a constant factor of $q_2(x'|x)$ in terms of asymptotic efficiency.

To understand the asymptotic efficiency of MCMC transition kernels, we can study the asymptotic variance and spectral gap of the kernel. The asymptotic variance is defined as

$$\text{var}_p(h, Q) = \lim_{T\to\infty} \frac{1}{T}\text{var}\left(\sum_{t=1}^{T} h(x_t)\right) \tag{7}$$

where $h : \mathcal{X} \to R$ is a scalar-valued function, Q is a p-stationary Markov transition kernel, and $X_1 \sim p(x)$. The spectral gap is defined as

$$\text{Gap}(Q) = 1 - \lambda_2 \tag{8}$$

where λ_2 is the second largest eigenvalue of the transition probability matrix of Q. Both of these quantities measure the asymptotic efficiency of Q. The asymptotic variances measures the additional variance incurred when using sequential samples from Q to estimate $E_p[h(x)]$. For transition probability matrices with non-negative eigenvalues, the spectral gap is related to the mixing time, with larger values corresponding to faster mixing chains (Levin & Peres, 2017).

Since our method approximates $q_2(x'|x)$, we should expect some decrease in efficiency. We characterize this decrease in terms of the asymptotic variance and spectral gap, under the assumption of Lipschitz continuity of $\nabla_x f(x)$. In particular, we show that the decrease is a constant factor that depends on the Lipschitz constant of $\nabla_x f(x)$ and the window size of our proposal.

Theorem 1 *Let $Q(x', x)$ and $Q^\nabla(x', x)$ be the Markov transition kernels given by the Metropolis-Hastings algorithm using the locally balanced proposal $q_2(x'|x)$ and our approximation $q^\nabla(x'|x)$. Let f be an L-smooth log-probability function and $p(x) = \frac{\exp(f(x))}{Z}$. Then it holds*

(a) $\textit{var}_p(h, Q^\nabla) \leq \frac{\textit{var}_p(h,Q)}{c} + \frac{1-c}{c} \cdot \textit{var}_p(h)$

(b) $\textit{Gap}(Q^\nabla) \geq c \cdot \textit{Gap}\,(Q)$

where $c = e^{-\frac{1}{2}LD_H^2}$ and $D_H = \sup_{x'\in H(x)} ||x - x'||$.

A proof can be found in Appendix B. This roughly states that $Q^\nabla(x', x)$ is no less than c-times as efficient than $Q(x', x)$ per step for estimating expectations. As expected, our approach matches the efficiency of the target proposal when the Taylor-series approximation is accurate.

We can derive from Theorem 1 that when considering which functional representation of our distribution to choose, we should choose the representation whose gradient has the smallest Lipschitz constant.

An example: Consider an Ising model on a cyclic 2D lattice. This model has log-probability function $f(x) = \theta \cdot x^T Jx - \log Z$ where J is the binary adjacency matrix of the lattice, θ is the connectivity strength and Z is the unknown normalizing constant. We can see the gradient is $\nabla_x f(x) = 2\theta \cdot Jx$ and can bound $L \leq 2\sigma(J)\theta = 8\theta$.

For $\theta = .1$ and a Hamming window of size 1, this gives $c = .67$, regardless of D. Since a single evaluation of f has an $\mathcal{O}(D^2)$ cost, it costs $\mathcal{O}(D^3)$ to compute[2] $q_2(x'|x)$. Compared to the exact local-differences proposal, the Gibbs-With-Gradients proposal $q^\nabla(x'|x)$ incurs at most a constant loss in sampling efficiency *per-iteration* but gives a $\mathcal{O}(D)$ increase in speed.

5. Relationship to Continuous Relaxations

Why hasn't this relatively simple approach been proposed before? A number of prior works have used gradient information for discrete sampling. Instead of using gradients to inform discrete updates directly, these methods transport the problem into a continuous relaxation, perform updates

[2]The local difference function of an Ising model can be computed more efficiently. However, this requires a bespoke derivation and implementation, and is not possible for general pmfs, such as those parameterized by neural networks.

there, and transform back after sampling. This approach incurs many of the pitfalls of continuous sampling without providing the scalability. We find these methods are not competitive with Gibbs-With-Gradients in high dimensions.

In more detail, these methods use the discrete target distribution to create a related continuous distribution (relaxation) whose samples can be transformed to samples from the target distribution. They then apply gradient-based sampling methods such as Stein Variational Gradient Descent (Liu & Wang, 2016) or Hamiltonian Monte-Carlo (HMC) (Neal et al., 2011) to the new distribution. Examples of such methods are the recently proposed Discrete-SVGD (D-SVGD) (Han et al., 2020) and Discontinuous HMC (Nishimura et al., 2017).

A key challenge of these approaches is that the relaxed distribution can be arbitrarily difficult to sample from, highly multi-modal and require small step-sizes. Further, metrics of MCMC performance and mixing in the relaxed space may not indicate performance and mixing in the discrete space. These methods also require the tuning of many additional hyper-parameters such as step-size, momentum, and the temperature of the relaxation. In contrast, our approach operates directly in discrete space, and has no hyper-parameters.

Figure 2 compares these approaches on the task of sampling from restricted Boltzmann machines (RBMs) of up to 1000 dimensions. We compare to D-SVGD and two relaxation-based baselines derived from the Metropolis-Adjusted Langevin Algorithm (Besag, 1994) and HMC. We compare the log-MMD between generated samples and "ground-truth" samples generated with Block-Gibbs. We also display samples from an MNIST-trained model. In contrast to all three baselines, our approach does not degrade with dimension. Additional results, details, and discussion can be found in Appendix C. These relaxation-based approaches do not scale beyond 200 dimensions, so we do not compare to them in our main experimental results section

6. Sampling From EBMs

To demonstrate the benefits and generality of our proposed approach to sampling, we present results sampling from 3 distinct and challenging distributions; Restricted Boltzmann Machines, Lattice Ising models, and Factorial Hidden Markov Models. Each is evaluated differently based on the properties of the distribution. We compare our sampler, Gibbs-With-Gradients, against standard Gibbs sampling and the Hamming Ball Sampler (Titsias & Yau, 2017) – two generic approaches for discrete sampling. When available, we also compare with samplers which exploit known structure in the distribution of interest.

In the following, Gibbs-X refers to Gibbs sampling with a block-size of X, and HB-X-Y refers to the Hamming Ball sampler with a block size of X and a hamming ball size of Y, and GWG refers to Gibbs-With-Gradients. Gibbs-1 is the fastest sampler tested. In our current implementation, we find Gibbs-2, HB-10-1, and GWG have approximately 1.6, 6.6, 2.1 times the cost of Gibbs-1 per step, respectively. Thus the run-time of GWG is most comparable to Gibbs-2.

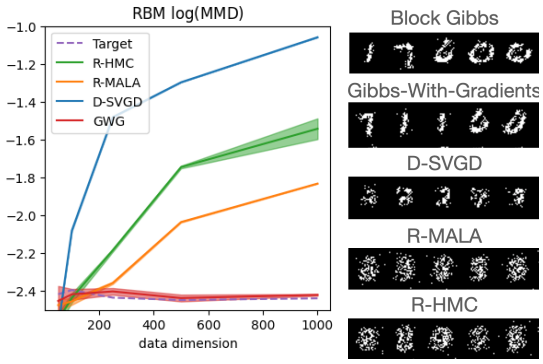

Figure 2. Comparison to gradient-based samplers with continuous relaxations. GWG, D-SVGD, R-HMC, and R-MALA refer to Gibbs-With-Gradients, Discrete SVGD, Relaxed HMC and Relaxed MALA, respectively. Left: Log-MMD (lower is better) between true samples and generated samples for RBMs of increasing dimension (over 3 runs). "Target" is log-MMD between two sets of Block-Gibbs samples. Right: Visualized samples from an RBM trained on MNIST.

Restricted Boltzmann Machines are unnormalized latent-variable models defined as:

$$\log p(x) = \log(1 + \exp(Wx + c)) + b^T x - \log Z \quad (9)$$

where $\{W, b, c\}$ define its parameters and $x \in \{0, 1\}^D$. We train an RBM with 500 hidden units on the MNIST dataset using contrastive divergence (Hinton, 2002). We generate samples with various MCMC samplers and compare them in two ways. First, using the Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) between a set of samples from each sampler and a set of "ground-truth" samples generated using the structured Block-Gibbs sampler available to RBMs (see Appendix E for details). Next, we report the Effective Sample Size (ESS) of a summary statistic over sampling trajectories. Results can be seen in Figure 3.

We see on the left that GWG matches the structured Block-Gibbs sampler in MMD ("Target" in the Figure), while the other samplers do not. On the right we see that the effective sample size of GWG is notably above the baselines and is approximately halfway between the baselines and the Block-Gibbs sampler (in log-space). We note, Block-Gibbs can update all 784 dimensions in each iterations. GWG and Gibbs-1 can update 1 and Gibbs-2 and Hamming Ball can update 2 dimensions per iteration.

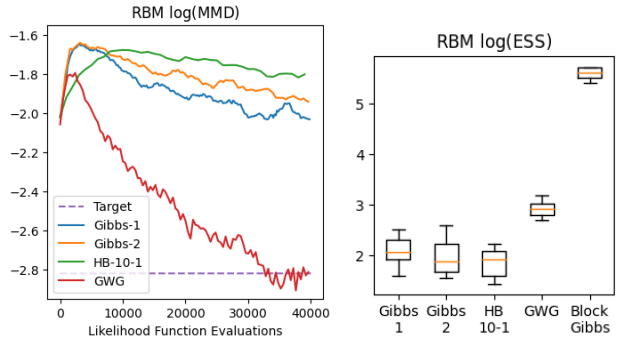

Figure 3. RBM sampling results. Left: Log-MMD of samples over steps (lower is better). "Target" is Log-MMD between two sets of Block-Gibbs samples. Right: Log-ESS of various samplers after 10,000 steps. As each sampler evaluates the likelihood a different number of times per sampling step, we plot by likelihood evaluations. Gibbs-With-Gradients matches Block-Gibbs in MMD and outperforms unstructured baselines in ESS.

Lattice Ising Models are models for binary data defined by

$$\log p(x) = \theta \cdot x^T Jx - \log Z \tag{10}$$

where θ is the connectivity strength and J is the binary adjacency matrix, here restricted to be a 2D cyclic lattice. This model was originally proposed to model the spin magnetic particles (Ising, 1924). We sample from models with increasing dimension and connectivity strength. We evaluate using Effective Sample Size (ESS)[3] of a summary statistic (full details in Appendix F). Results can be seen in Figure 4.

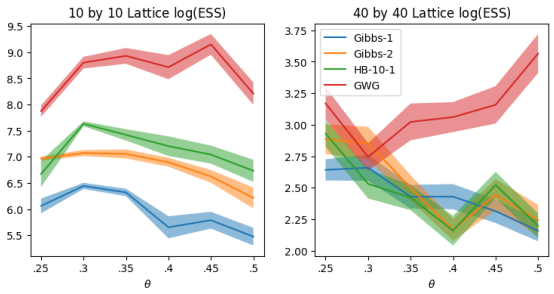

Figure 4. Ising model sampling results. The y-axis shows log-ESS over 100,000 samples steps. Left: 10x10 lattice, right: 40x40 lattice. We see Gibbs-With-Gradients (GWG) outperforms in most settings.

We see Gibbs-With-Gradients provides a notable increase in ESS for Ising models with higher connectivity. These models are harder to sample from as their dimensions are more correlated.

Factorial Hidden Markov Models (FHMM) are latent-variable time-series models, similar to HMMs but their hidden state consists of distinct, independent factors. The continuous data $y \in R^L$ of length L is generated by the binary hidden state $x \in \{0,1\}^{L \times K}$ with K factors as

$$\begin{aligned} p(x,y) &= p(y|x)p(x) \\ p(y|x) &= \prod_{t=1}^{L} \mathcal{N}(y_t; Wx_t + b, \sigma^2) \\ p(x) &= p(x_1) \prod_{t=2}^{L} p(x_t|x_{t-1}) \end{aligned} \tag{11}$$

We create a random FHMM with 1000 time-steps and a 10-dimensional hidden state and then draw samples y. We generate posterior samples $p(x|y)$ and evaluate our samplers using reconstruction error and joint likelihood. Full model description and experimental details can be found in Appendix G and results can be seen in Figure 5.

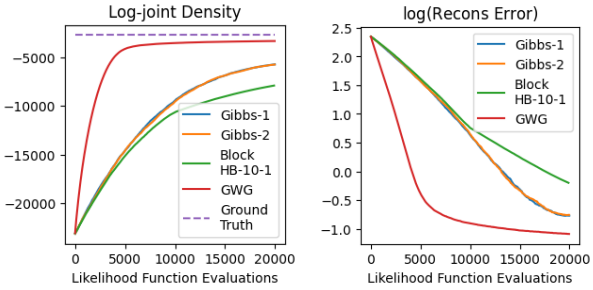

Figure 5. Factorial Hidden Markov Model results. "Block-HB" refers to the block-structured hamming ball sampler. Left, log-joint density and right, mean log-reconstruction error. As each sampler evaluates the likelihood a different number of times per sampling step, we plot by likelihood evaluations. GWG performs best in both evaluations, outperforming the Hamming Ball sampler which exploits model structure.

In this setting, the Hamming-Ball sampler exploits known structure in the problem. Each block chosen by the sampler consists of the 10-dimensional latent state x_t, as opposed to 10 random dimensions. Thus, the Hamming-Ball sampler in this setting is a stronger baseline. Despite this, we find Gibbs-With-Gradients notably outperforms the baseline samplers.

7. Training EBMs

Training EBMs is a challenging task. Computing the likelihood for Maximum Likelihood inference requires computation of the normalizing constant $Z = \sum_x e^{f_\theta(x)}$ which is

[3] Computed using Tensorflow Probability

typically intractable. Thankfully, the gradient of the log-likelihood can be more easily expressed as:

$$\nabla_\theta \log p(x) = \nabla_\theta f_\theta(x) - E_{p(x)}[\nabla_\theta f_\theta(x)] \tag{12}$$

therefore, if samples can be drawn from $p(x)$, then an unbiased gradient estimator can be derived. We can approximate this estimator using MCMC. When a slow-mixing MCMC sampler is used to draw these samples, we obtain biased gradient estimates and this leads to sub-par learning. Improvements in MCMC can then lead to improvements in parameter inference for unnormalized models. We explore how Gibbs-With-Gradients can be applied to parameter inference for some classical discrete EBMs.

7.1. Training Ising models on generated data

We generate Ising models with different sparse graph structures; a 2D cyclic lattice and a random Erdos-Renyi graph. We generate training data with a long-run Gibbs chain and train models using Persistent Contrastive Divergence (Tieleman, 2008) with an $\ell 1$ penalty to encourage sparsity. We evaluate our models using the Root-Mean-Squared-Error (RMSE) between the inferred connectivity matrix $\hat{J}$ and the true matrix J.

Full experimental details and additional results can be found in Appendix F.2, F.3 and results can be seen in Figure 6. In all settings, GWG greatly outperforms Gibbs sampling. This allows for much faster training than standard Gibbs while recovering higher-quality solutions.

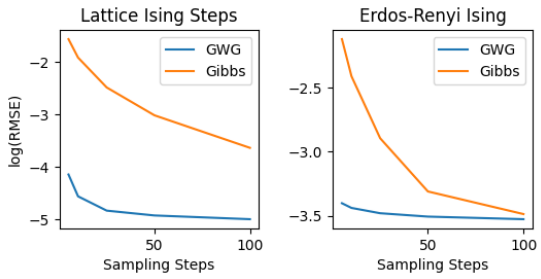

Figure 6. Training Ising models with increasing MCMC steps. Left: Lattice Ising (dim = 625, $\theta = .25$). Right: Erdos-Renyi Ising. Values are log(RMSE) between the learned and true J. GWG leads to better solutions with lower computational cost.

7.2. Protein Coupling Prediction with Potts Models

Proteins are defined by a discrete sequence of 20 amino acids $x \in \{1, \dots, 20\}^D$ where D is the length of the protein. The Potts model has long been a popular approach for modelling the evolutionary distribution of protein sequences (Lapedes et al., 1999). The model takes the form

$$\log p(x) = \sum_{i=1}^{D} h_i^T x_i + \sum_{i,j=1}^{D} x_i^T J_{ij} x_j - \log Z \tag{13}$$

where x_i is a one-hot encoding of the i-th amino acid in x, $J \in R^{\{D \times D \times 20 \times 20\}}$ and $h \in R^{\{D \times 20\}}$ are the model's parameters and Z is the model's normalizing constant. The Potts model's likelihood is the sum of pairwise interactions. Marks et al. (2011) demonstrated that the strength of these interactions can correspond to whether or not two amino acids touch when the protein folds. These inferred contacts can then be used to infer the 3D structure of the protein.

Since the Potts model is unnormalized, maximum likelihood learning is difficult, and $\ell 1$-regularized Pseudo-likelihood Maximization (PLM) (Besag, 1975) is used to train the model. Recently Ingraham & Marks (2017) found that improved contact prediction could be achieved with MCMC-based maximum likelihood learning. Unfortunately, due to the limitations of discrete MCMC samplers, their study was restricted to small proteins (less than 50 amino acids).

GWG allows these performance improvements to scale to large proteins as well. We train Potts models on 2 large proteins: OPSD_BOVIN, and CADH1_HUMAN. We train using PCD where samples are generated with GWG and Gibbs. We run PLM as a baseline. These proteins are much larger than those studied in Ingraham & Marks (2017) with OPSD_BOVIN, and CADH1_HUMAN having 276, and 526 amino acids, respectively[4].

We predict couplings using the J parameter of our models. We compute a "coupling-strength" for each pair of amino-acids as $||J_{ij}||_2$ which gives a measure of how much indices i and j interact to determine the fitness of a protein. We sort index pairs by their coupling strength and compare the highest scoring pairs with known contacts in the proteins.

Full experimental details and additional results can be found in Appendix H and results can be seen in Figure 7. For the smaller protein, Gibbs sampling outperforms PLM but for the larger protein, the slow-mixing of the sampler causes the performance to drop below that of PLM. Despite the increased size, Gibbs-With-Gradients performs the best.

8. Deep EBMs for Discrete Data

Deep Energy-Based Models have rapidly gained popularity for generative modeling. These models take the form

$$\log p(x) = f_\theta(x) - \log Z \tag{14}$$

where $f_\theta : R^D \to R$ is a deep neural network. The recent success of these models can be attributed to a few

[4] After standard data pre-processing as in Ingraham & Marks (2017)

Data Type	Dataset	VAE (MLP)	VAE (Conv)	EBM (GWG)	EBM (Gibbs)	RBM	DBN
Binary (log-likelihood ↑)	Static MNIST	-86.05	-82.41	**-80.01**	-117.17	-86.39	-85.67
	Dynamic MNIST	-82.42	**-80.40**	-80.51	-121.19	—	—
	Omniglot	-103.52	-97.65	**-94.72**	-142.06	-100.47	-100.78
	Caltech Silhouettes	-112.08	-106.35	**-96.20**	-163.50	—	—
Categorical (bits/dim ↓)	Frey Faces	4.61	**4.49**	4.65	—	—	—
	Histopathology	5.82	5.59	**5.08**	—	—	—

Table 2. Test-set log-likelihoods for models trained on discrete image datasets. RBM and DBN results are taken from Burda et al. (2015), VAE results taken from Tomczak & Welling (2018).

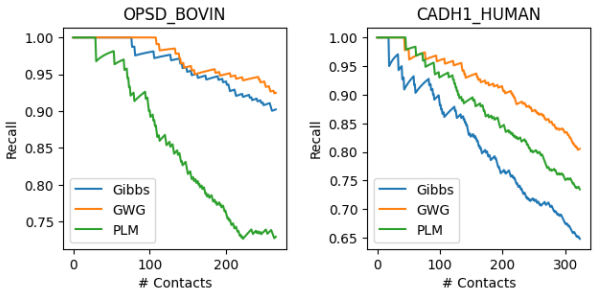

Figure 7. Recall Curves for contact prediction with Potts models. Gibbs-With-Gradients leads to higher recall.

advancements including; the use of tempered Langevin samplers (Nijkamp et al., 2020) and large persistent chains (Du & Mordatch, 2019). This has enabled EBMs to become a competitive approach for image-generation (Song & Ermon, 2019), adversarial robustness (Grathwohl et al., 2019; Hill et al., 2020), semi-supervised learning (Song & Ou, 2018; Grathwohl et al., 2020a) and many other problems.

These advances rely on gradient-based sampling which requires continuous data. Thus, these scalable methods cannot be applied towards training deep EBMs on discrete data. We explore how Gibbs-With-Gradients can enable the training of deep EBMs on high-dimensional binary and categorical data. To our knowledge, models of this form have not be successfully trained on such data in the past. We train deep EBMs parameterized by Residual Networks (He et al., 2016) on small binary and continuous image datasets using PCD (Tieleman, 2008) with a replay buffer as in Du & Mordatch (2019); Grathwohl et al. (2019). The continuous images were treated as 1-of-256 categorical data.

PCD training is very sensitive to the choice of MCMC sampler. As an initial experiment, we attempted to train these models using standard Gibbs but found that the sampler was too slow to enable stable training within a reasonable compute budget. On the binary data we needed to train with **800** Gibbs sampling steps per training iteration. All models we trained with fewer steps quickly diverged. GWG required only 40. This made training with Gibbs 9.52x slower than GWG. For a fair comparison, the Gibbs results in Table 2 were trained for an equal amount of wall-clock time as the GWG models.

For the categorical data, we could not train models with Gibbs sampling. Each step of Gibbs requires us to evaluate the energy function 256 (for each possible pixel value) times. GWG requires 2 function evaluations. Thus the amount of compute per iteration for Gibbs is 128x greater than GWG. Further, to make Gibbs train stably, we would need to use many more steps, as with the binary data. This would give roughly a **870x** increase in run-time. Therefore, training a model of this form with Gibbs is simply not feasible.

Full experimental details can be found in Appendix I. We present long-run samples from our trained models in Figure 8 and test-set likelihoods in Table 2. Likelihoods are estimated using Annealed Importance Sampling (Neal, 2001). We compare the performance of our models to Variational Autoencoders (Kingma & Welling, 2013) and two other EBMs; an RBM and a Deep Belief Network (DBN) (Hinton, 2009). On most datasets, our Resnet EBM outperforms the other two EBMs and the VAEs. Our improved sampler enables deep EBMs to become a competitive approach to generative modeling on high-dimensional discrete data.

We include some preliminary results using Gibbs-With-Gradients to train EBMs for text data in Appendix J.

9. Future Directions and Conclusion

In this work we have presented Gibbs-With-Gradients, a new approach to MCMC sampling for discrete distributions. Our approach exploits a powerful structure, gradient information, which is available to a very large class of important discrete distributions. We then use this gradient information to construct proposal distributions for Metropolis-Hastings.

We have demonstrated on a diverse set of distributions that this approach to sampling considerably outperforms base-

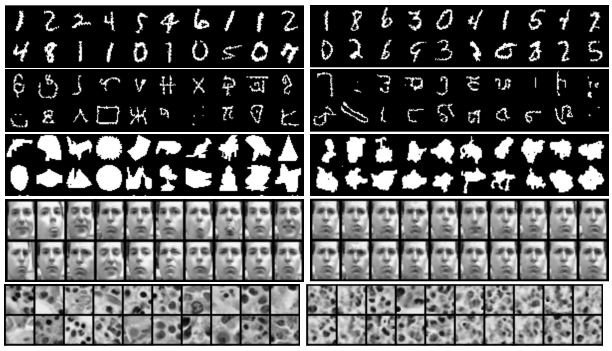

Figure 8. Left: data. Right: Samples from ResNet EBM. Samples generated with annealed Markov chain using 300,000 Gibbs-With-Gradients steps. Top to bottom: MNIST, omniglot, Caltech Silhouettes, Frey Faces, Histopathology.

line samplers which do not exploit known structure in the target distribution as well as many that do. Further, we find our approach outperforms prior discrete samplers which use gradient information with continuous relaxations.

We find Gibbs-With-Gradients performs very well at sampling from deep energy-based models and allows, for the first time, unconstrained deep EBMs to be trained on discrete data and outperform other deep generative models.

We believe there is considerable room for future work building on top of our method. We only explored samplers which modify 1 variable per proposed update. We believe considerable improvements could be made if the window size of the sampler was expanded but this would require more efficient algorithms to sample from the larger proposal.

Next, we have shown that gradient-based approximations to the local difference function can be accurate and useful for complex discrete distributions. Local difference functions have been used in the past to generalize Score Matching (Lyu, 2012), and Stein Discrepancies (Han & Liu, 2018). We believe there is great potential to explore how gradient-based approximations could enable the generalization of recent deep EBM training methods based on Score Matching and Stein Discrepancies (Song & Ermon, 2019; Grathwohl et al., 2020b) to models of discrete data.

10. Acknowledgements

We would like to thank Eli Weinstein for helping us properly present our protein model results and we would like to thank Kelly Brock for help and feedback for working with the protein data. We thank Jesse Bettencourt, James Lucas, and Jacob Kelly for helpful feedback on our draft. Last, we would like to thank Jeffrey Rosenthal for providing information on related methods and prior work.

References

Besag, J. Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 24(3):179–195, 1975.

Besag, J. Comments on "representations of knowledge in complex systems" by u. grenander and mi miller. *J. Roy. Statist. Soc. Ser. B*, 56:591–592, 1994.

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

Deng, Y., Bakhtin, A., Ott, M., Szlam, A., and Ranzato, M. Residual energy-based models for text generation. *arXiv preprint arXiv:2004.11714*, 2020.

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

Gers, F. A., Schmidhuber, J., and Cummins, F. Learning to forget: Continual prediction with lstm. 1999.

Grathwohl, W., Wang, K.-C., Jacobsen, J.-H., Duvenaud, D., Norouzi, M., and Swersky, K. Your classifier is secretly an energy based model and you should treat it like one. *arXiv preprint arXiv:1912.03263*, 2019.

Grathwohl, W., Kelly, J., Hashemi, M., Norouzi, M., Swersky, K., and Duvenaud, D. No mcmc for me: Amortized sampling for fast and stable training of energy-based models. *arXiv preprint arXiv:2010.04230*, 2020a.

Grathwohl, W., Wang, K.-C., Jacobsen, J.-H., Duvenaud, D., and Zemel, R. Learning the stein discrepancy for training and evaluating energy-based models without sampling. In *International Conference on Machine Learning*, pp. 3732–3747. PMLR, 2020b.

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

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

Han, J. and Liu, Q. Stein variational gradient descent without gradient. *arXiv preprint arXiv:1806.02775*, 2018.

Han, J., Ding, F., Liu, X., Torresani, L., Peng, J., and Liu, Q. Stein variational inference for discrete distributions. *arXiv preprint arXiv:2003.00605*, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, T., McCann, B., Xiong, C., and Hosseini-Asl, E. Joint energy-based model training for better calibrated natural language understanding models. *arXiv preprint arXiv:2101.06829*, 2021.

Hill, M., Mitchell, J., and Zhu, S.-C. Stochastic security: Adversarial defense using long-run dynamics of energy-based models. *arXiv preprint arXiv:2005.13525*, 2020.

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

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

Ingraham, J. and Marks, D. Variational inference for sparse and undirected models. In *International Conference on Machine Learning*, pp. 1607–1616. PMLR, 2017.

Ising, E. *Beitrag zur Theorie des Ferround Paramagnetismus*. PhD thesis, PhD thesis (Mathematisch-Naturwissenschaftliche Fakultät der Hamburgischen . . . , 1924.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

Lapedes, A. S., Giraud, B. G., Liu, L., and Stormo, G. D. Correlated mutations in models of protein sequences: phylogenetic and structural effects. *Lecture Notes-Monograph Series*, pp. 236–256, 1999.

Łatuszyński, K., Roberts, G. O., Rosenthal, J. S., et al. Adaptive gibbs samplers and related mcmc methods. *The Annals of Applied Probability*, 23(1):66–98, 2013.

Levin, D. A. and Peres, Y. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

Liu, J. S. Peskun's theorem and a modified discrete-state gibbs sampler. *Biometrika*, 83(3), 1996.

Liu, Q. and Wang, D. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, 29:2378–2386, 2016.

Lyu, S. Interpretation and generalization of score matching. *arXiv preprint arXiv:1205.2629*, 2012.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

Marks, D. S., Colwell, L. J., Sheridan, R., Hopf, T. A., Pagnani, A., Zecchina, R., and Sander, C. Protein 3d structure computed from evolutionary sequence variation. *PloS one*, 6(12):e28766, 2011.

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

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

Nijkamp, E., Hill, M., Han, T., Zhu, S.-C., and Wu, Y. N. On the anatomy of mcmc-based maximum likelihood learning of energy-based models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5272–5280, 2020.

Nishimura, A., Dunson, D., and Lu, J. Discontinuous hamiltonian monte carlo for sampling discrete parameters. *arXiv preprint arXiv:1705.08510*, 2, 2017.

Ramachandran, P., Zoph, B., and Le, Q. V. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

Richardson, S., Bottolo, L., and Rosenthal, J. S. Bayesian models for sparse regression analysis of high dimensional data. *Bayesian Statistics*, 9:539–569, 2010.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pp. 11918–11930, 2019.

Song, Y. and Ou, Z. Learning neural random fields with inclusive auxiliary generators. *arXiv preprint arXiv:1806.00271*, 2018.

Taylor, A., Marcus, M., and Santorini, B. The penn treebank: an overview. *Treebanks*, pp. 5–22, 2003.

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

Tieleman, T. and Hinton, G. Using fast weights to improve persistent contrastive divergence. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1033–1040, 2009.

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

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

Umrigar, C. J. Accelerated metropolis method. *Phys. Rev. Lett.*, 71:408–411, Jul 1993. doi: 10.1103/PhysRevLett.71.408. URL https://link.aps.org/doi/10.1103/PhysRevLett.71.408.

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