Oops I took a gradient!

Scalable sampling for discrete distributions

ICML 2021

Will Grathwohl
Kevin Swersky
Milad Hashemi
David Duvenaud
Chris J. Maddison

VECTOR INSTITUTE

Google AI

Energy-Based Models

- **An energy-based model (EBM) is a probability model in the following form:**

$$p_\theta(x) = \frac{e^{-E_\theta(x)}}{Z(\theta)} \qquad Z(\theta) = \int_x e^{-E_\theta(x)} dx$$

- **Where $E_\theta(x) : \chi \to R$ fully specifies the model so $Z(\theta)$ does not need to be modelled**

Training EBMs

- **To maximize likelihood we must compute**

$$\begin{aligned}\log p_\theta(x) &= -E_\theta(x) - \log Z(\theta) \\ &= -E_\theta(x) - \log \int e^{-E_\theta(x)} dx\end{aligned}$$

Training EBMs

- To maximize likelihood we must compute

$$\begin{aligned}\log p_\theta(x) &= -E_\theta(x) - \log Z(\theta) \\ &= -E_\theta(x) - \underline{\log \int e^{-E_\theta(x)} dx}\end{aligned}$$

- Which is intractable

Training EBMs

- **To maximize likelihood we must compute**

$$\begin{aligned}\log p_\theta(x) &= -E_\theta(x) - \log Z(\theta)\\ &= -E_\theta(x) - \log \int e^{-E_\theta(x)} dx\end{aligned}$$

- **Which is intractable**

- **The gradient however is simpler**

$$\nabla_\theta \log p_\theta(x) = -\nabla_\theta E_\theta(x) - \mathbf{E}_{p_\theta(x)}[\nabla_\theta E_\theta(x)]$$

Training EBMs

- To maximize likelihood we must compute

$$\begin{aligned}\log p_\theta(x) &= -E_\theta(x) - \log Z(\theta) \\ &= -E_\theta(x) - \log \int e^{-E_\theta(x)} dx\end{aligned}$$

- Which is intractable

- The gradient however is simpler

$$\nabla_\theta \log p_\theta(x) = -\nabla_\theta E_\theta(x) - \mathbf{E}_{p_\theta(x)}[\nabla_\theta E_\theta(x)]$$

- Draw samples to estimate gradient

- We can use this to train

Training EBMs

- To maximize likelihood we must compute

$$\begin{aligned}\log p_\theta(x) &= -E_\theta(x) - \log Z(\theta)\\ &= -E_\theta(x) - \log \int e^{-E_\theta(x)}dx\end{aligned}$$

- Which is intractable

- The gradient however is simpler

$$\nabla_\theta \log p_\theta(x) = -\nabla_\theta E_\theta(x) - \mathbf{E}_{\underline{p_\theta(x)}}[\nabla_\theta E_\theta(x)]$$

Use MCMC!

- Draw samples to estimate gradient

- We can use this to train

Recent Success!

- **Let** $E_\theta(x)$ **be a deep neural network** $E_\theta(x) = -f_\theta(x)$

- **How to sample?**

Recent Success!

- **Let** $E_\theta(x)$ **be a deep neural network** $E_\theta(x) = -f_\theta(x)$
- **How to sample?**
- **If data continuous, use gradient-based samplers!**

$$x_{t+t} = x_t + \frac{\epsilon}{2} \nabla_x f_\theta(x) + \epsilon \eta, \qquad \eta \sim N(0, I)$$

Recent Success!

- Let $E_\theta(x)$ be a deep neural network $E_\theta(x) = - f_\theta(x)$
- How to sample?
- If data continuous, use gradient-based samplers!

$$x_{t+t} = x_t + \frac{\epsilon}{2} \nabla_x f_\theta(x) + \epsilon\eta, \qquad \eta \sim N(0,I)$$

- High quality image generation
- Semi-supervised learning
- OOD
- Adversarial robustness

Du and Mordatch (2020)

Recent Success!

- **Let $E_\theta(x)$ be a deep neural network $E_\theta(x) = -f_\theta(x)$**
- **How to sample?**
- **If data discrete….?**

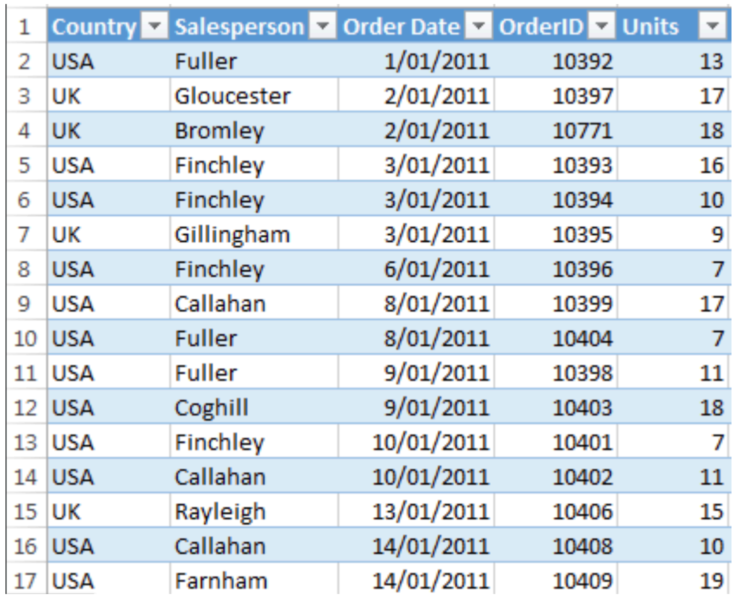
Recent Success!

- Let $E_\theta(x)$ be a deep neural network $E_\theta(x) = -f_\theta(x)$
- How to sample?
- If data discrete….?
- Many important data discrete…how to sample?

Text

["The", "cat", "sat"]
["The", "dog", "sat"]
["The", "dog", "ate"]
.
.
.

Tabular Data

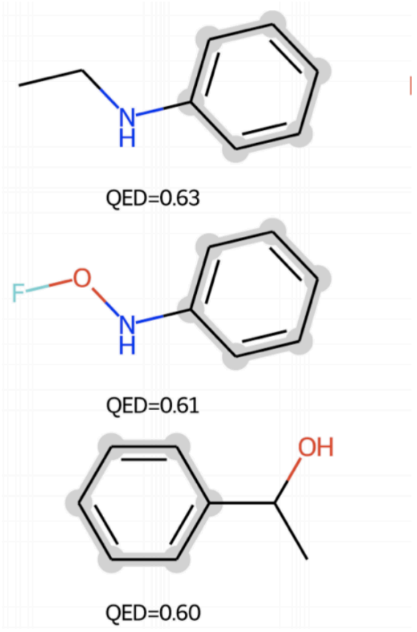


1	Country	Salesperson	Order Date	OrderID	Units
2	USA	Fuller	1/01/2011	10392	13
3	UK	Gloucester	2/01/2011	10397	17
4	UK	Bromley	2/01/2011	10771	18
5	USA	Finchley	3/01/2011	10393	16
6	USA	Finchley	3/01/2011	10394	10
7	UK	Gillingham	3/01/2011	10395	9
8	USA	Finchley	6/01/2011	10396	7
9	USA	Callahan	8/01/2011	10399	17
10	USA	Fuller	8/01/2011	10404	7
11	USA	Fuller	9/01/2011	10398	11
12	USA	Coghill	9/01/2011	10403	18
13	USA	Finchley	10/01/2011	10401	7
14	USA	Callahan	10/01/2011	10402	11
15	UK	Rayleigh	13/01/2011	10406	15
16	USA	Callahan	14/01/2011	10408	10
17	USA	Farnham	14/01/2011	10409	19

Proteins

Ingraham and Marks (2017)

Molecules



Hataya et al. (2021)

In this work…

- New MCMC sampler for discrete distributions

- Simple approach which exploits common structure (gradients!!!)

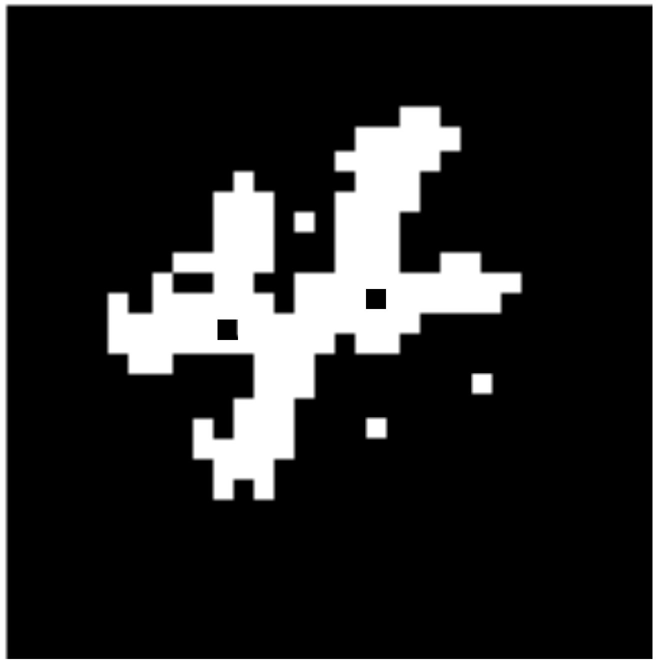
- Increases efficiency, enables the Deep EBMs on discrete data

Discrete Sampling

- **We focus on sampling from** $p(x) = \dfrac{e^{f(x)}}{Z}$ **where**
- $x \in \{0,1\}^D$ **or** $x \in \{0,\ldots,K\}^D$

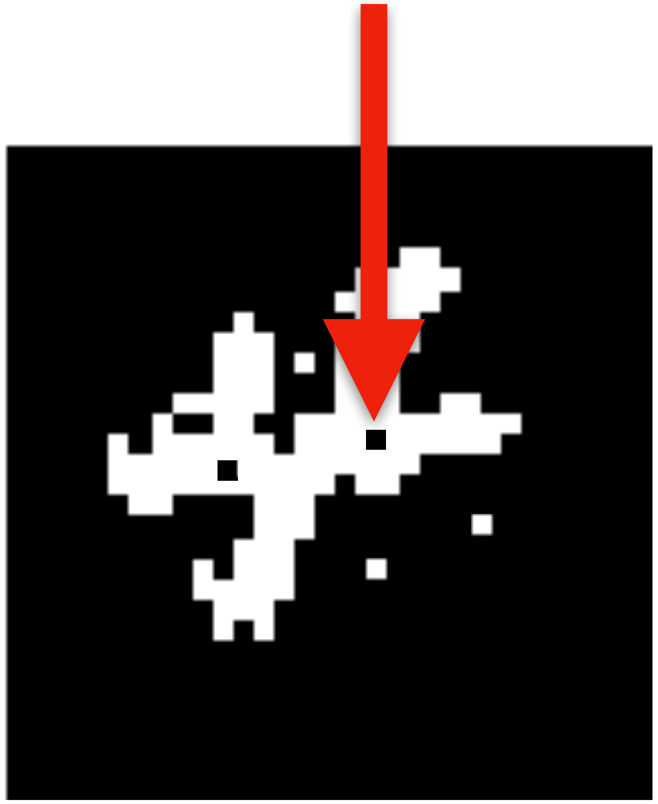
Gibbs Sampling

- **Pick dim i then re-sample $x[i]$ w/ all other dims fixed**



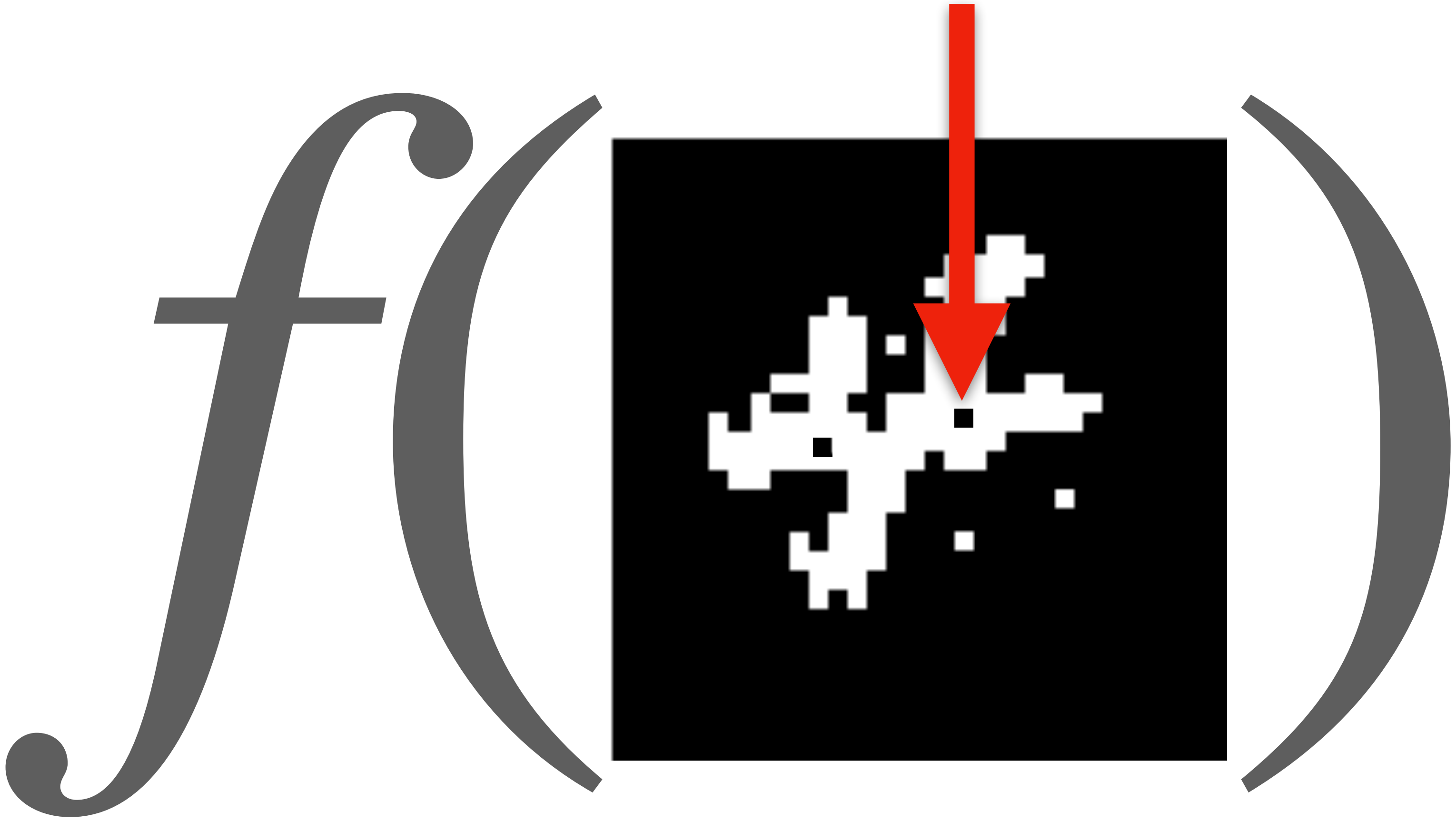
Gibbs Sampling

- Pick dim i then re-sample $x[i]$ w/ all other dims fixed
- Consider this dim



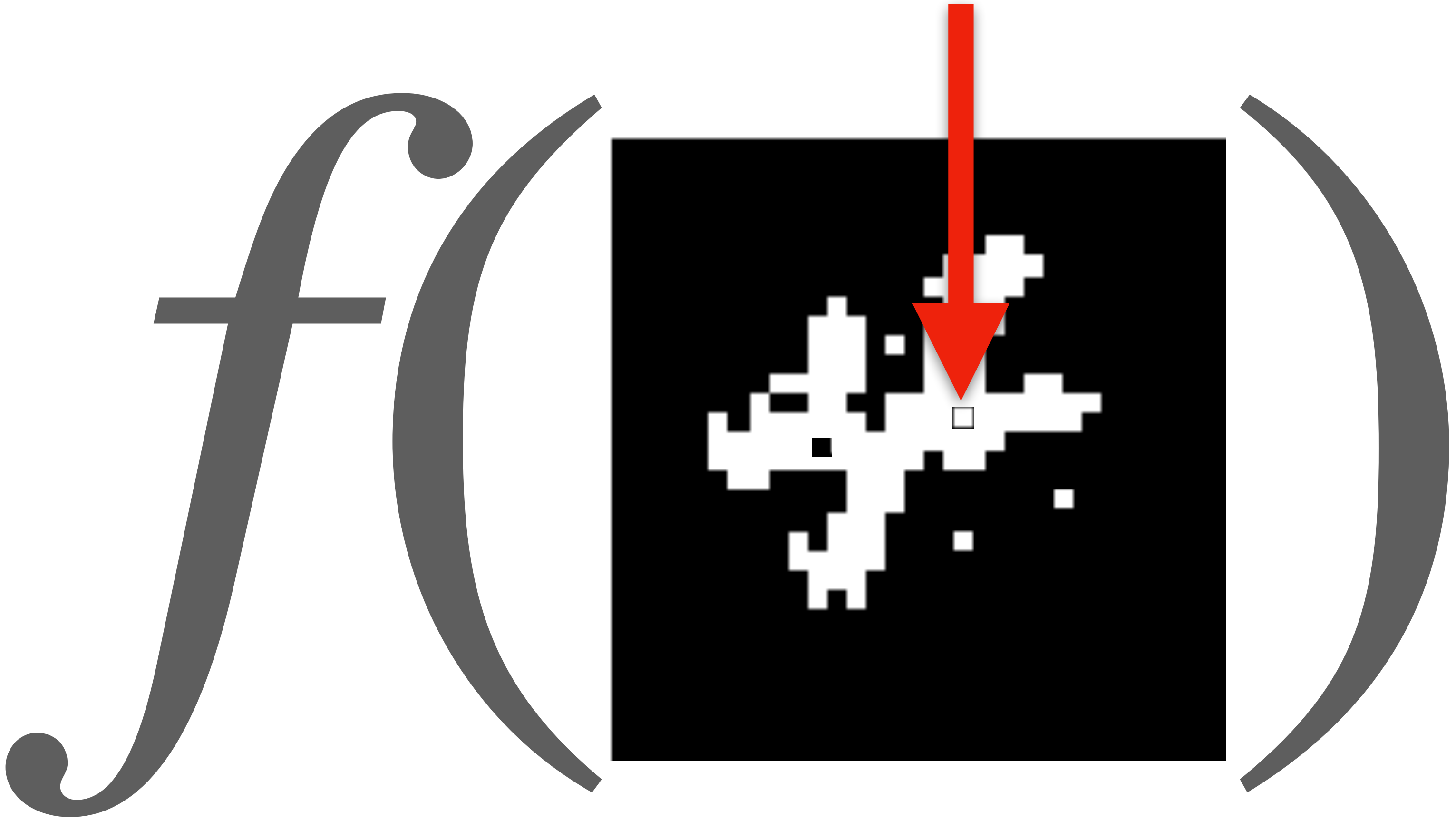
Gibbs Sampling

- Pick dim i then re-sample $x[i]$ w/ all other dims fixed
- Consider this dim
- We evaluate $f(x)$



Gibbs Sampling

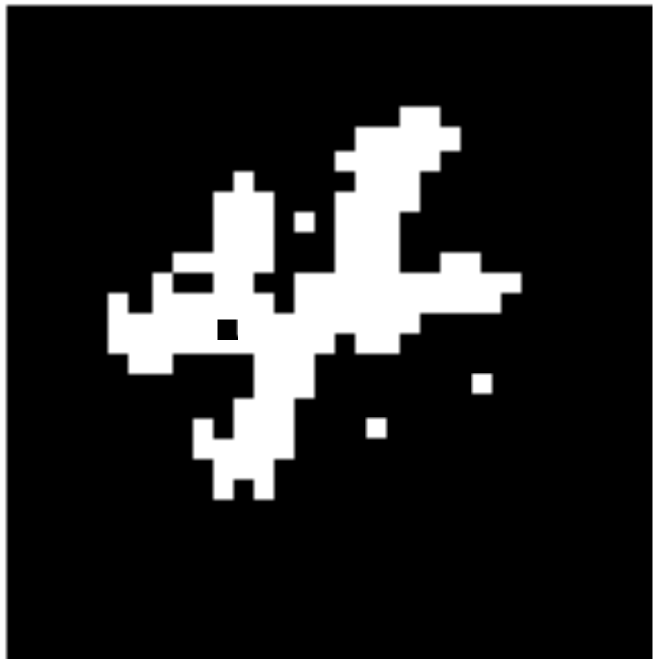
- Pick dim i then re-sample $x[i]$ w/ all other dims fixed
- Consider this dim
- We evaluate $f(x)$
- ...and $f(x_{-i})$ (flip i-th bit)



Gibbs Sampling

- **Pick dim i then re-sample $x[i]$ w/ all other dims fixed**
- **Consider this dim**
- **We evaluate $f(x)$**
- **…and $f(x_{-i})$ (flip i-th bit)**
- **Set $x \leftarrow x_{-i}$ with probability:**

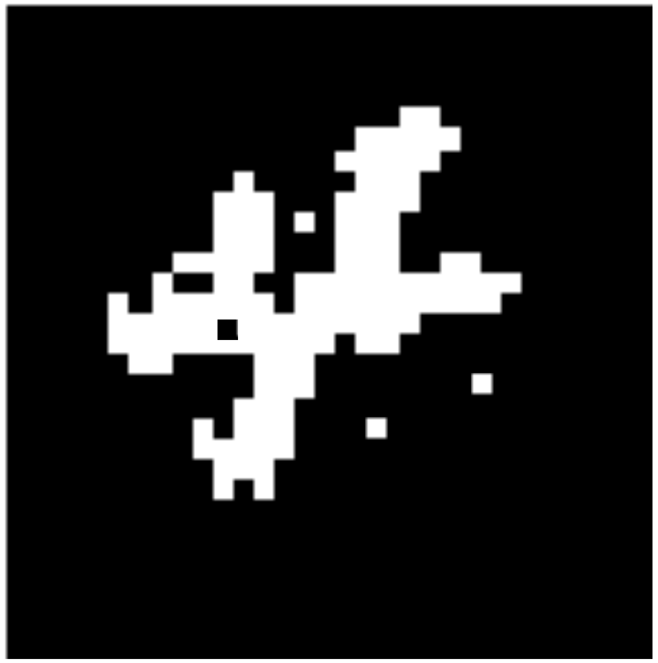
$$\sigma(f(x_{-i}) - f(x))$$



Gibbs Sampling

- Pick dim i then re-sample $x[i]$ w/ all other dims fixed

- Consider this dim



- We evaluate $f(x)$

- …and $f(x_{-i})$ (flip i-th bit)

- Set $x \leftarrow x_{-i}$ with probability:

$$\sigma(f(x_{-i}) - f(x))$$

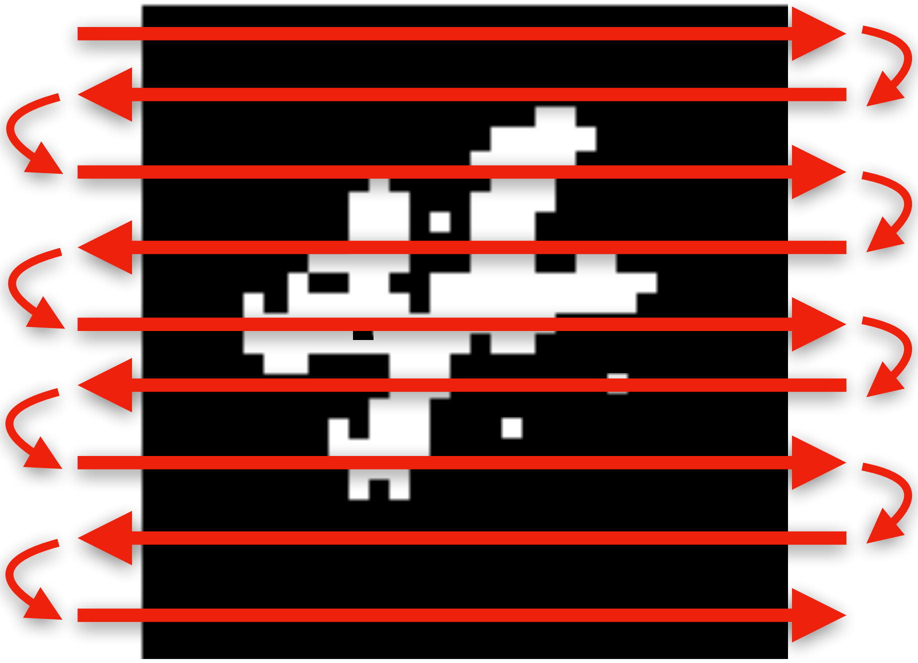
- Must resample all dims

Gibbs Sampling

- Pick dim i then re-sample $x[i]$ w/ all other dims fixed
- Consider this dim
- We evaluate $f(x)$
- ...and $f(x_{-i})$ (flip i-th bit)
- Set $x \leftarrow x_{-i}$ with probability:

$$\sigma(f(x_{-i}) - f(x))$$

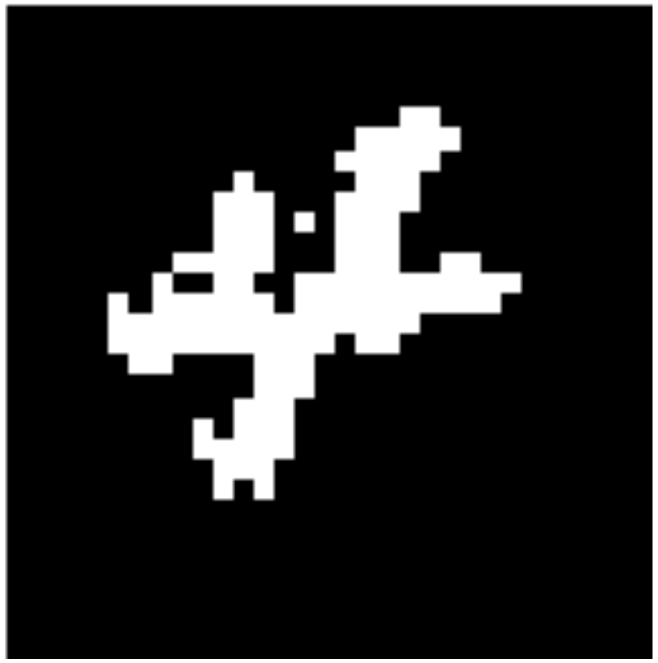
- Must resample all dims



Typically fix an ordering and iterate through

Some dims are better…

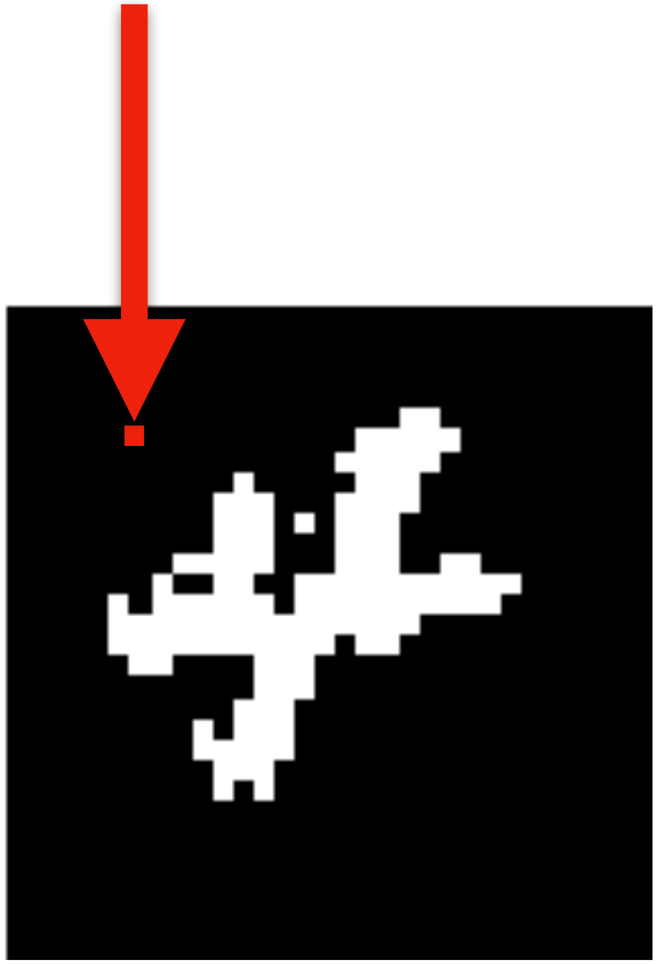
- **Most pixels are black**



Some dims are better…

- Most pixels are black

- If we propose dim in background
- Will not change → computation wasted

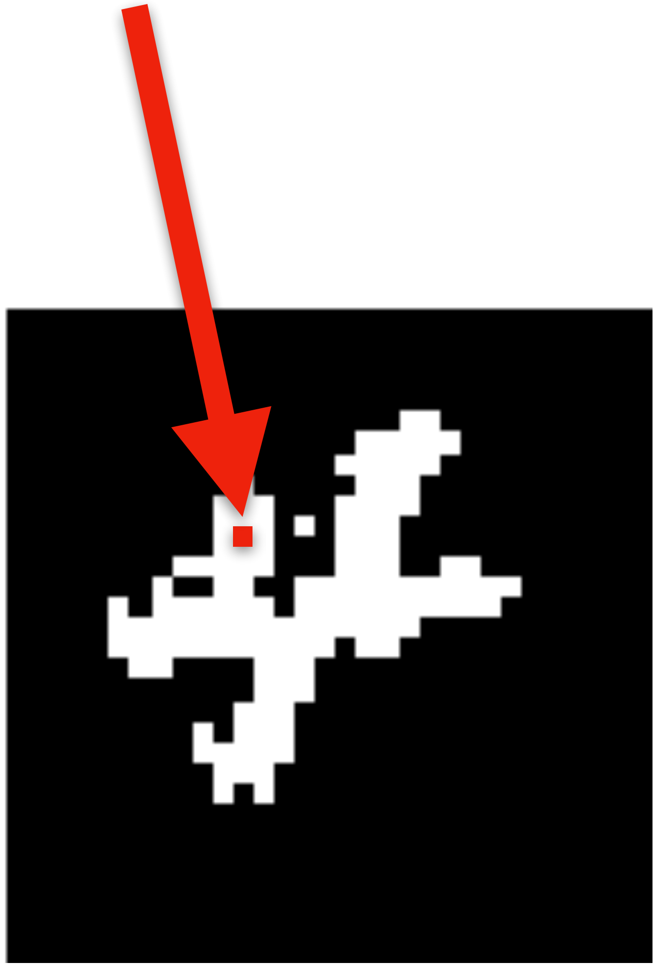


Some dims are better…

- Most pixels are black

- If we propose dim in background
- Will not change → computation wasted

- If we propose dim in middle of digit
- Will not change → computation wasted



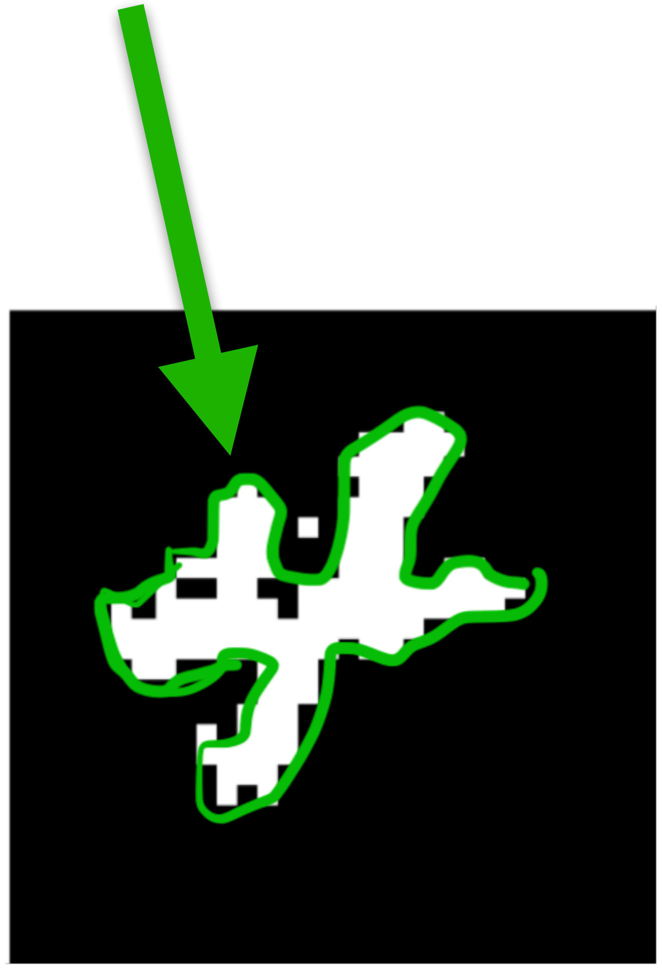
Some dims are better…

- Most pixels are black

- If we propose dim in background
- Will not change → computation wasted

- If we propose dim in middle of digit
- Will not change → computation wasted

- Dims on edge will change



Some dims are better…

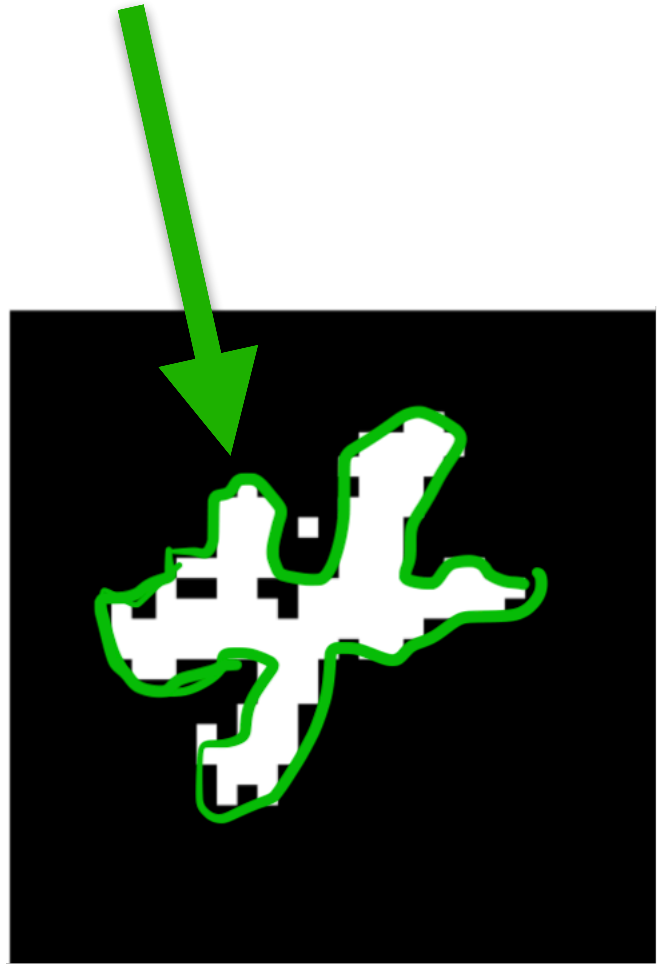
- Most pixels are black

- If we propose dim in background
- Will not change → computation wasted

- If we propose dim in middle of digit
- Will not change → computation wasted

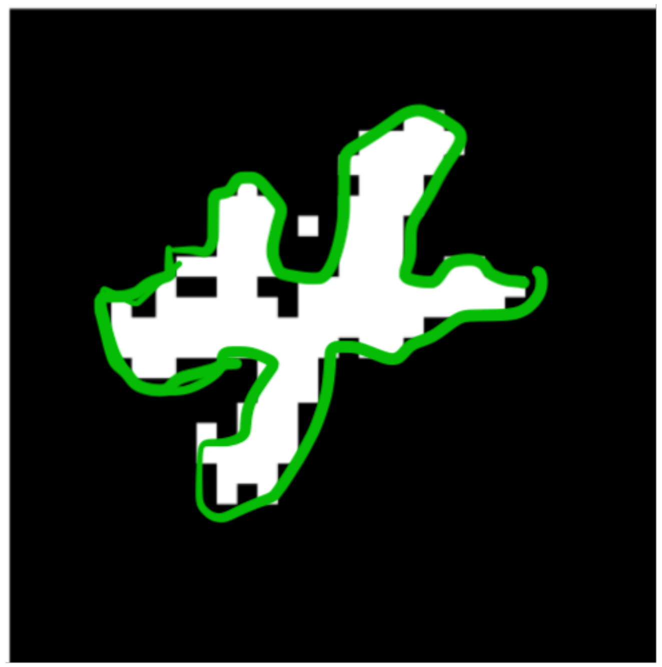
- Dims on edge will change

- Small subset of all variables! 2% on MNIST



Choosing dimensions

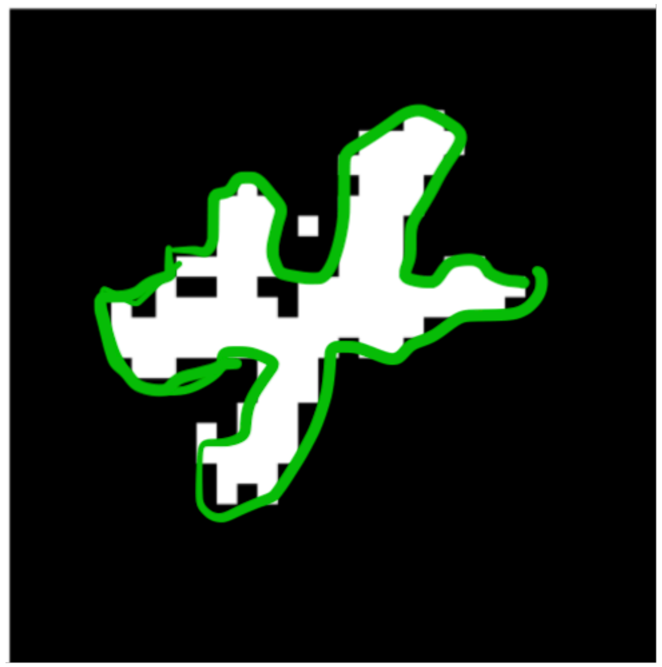
- Dims most likely to flip depend on input

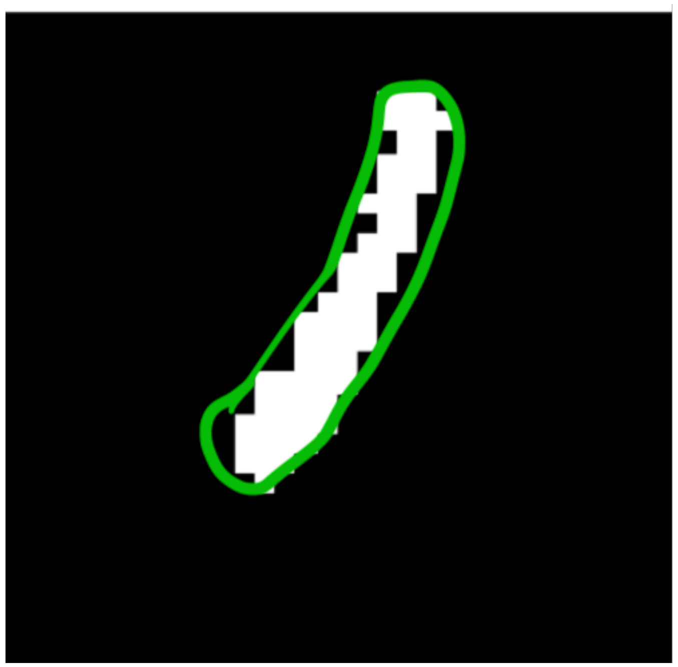


Choosing dimensions

- Dims most likely to flip depend on input
- Thus, sample dims from proposal $q(i \mid x)$
- To generate proposal, sample $i \sim q(i \mid x)$ and set $x_{-i} = \text{flip\_dim}(x, i)$
- Accept x_{-i} with probability

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





Proposals for Discrete Sampling

- **How to design $q(i \mid x)$? Acceptance prob:**

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

Proposals for Discrete Sampling

- **How to design $q(i \,|\, x)$? Acceptance prob:**

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

- **Want $\underline{f(x_{-i}) - f(x)}$ high to proposals have high likelihood**

Proposals for Discrete Sampling

- How to design $q(i \mid x)$? Acceptance prob:

$$\min \left\{ \exp(f(x_{-i}) - f(x)) \frac{q(i \mid x_{-i})}{\underline{q(i \mid x)}}, 1 \right\}$$

- Want $f(x_{-i}) - f(x)$ high to proposals have high likelihood

- Want $\underline{q(i \mid x)}$ to have high entropy

Proposals for Discrete Sampling

- **How to design $q(i|x)$? Acceptance prob:**

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

- **Want $f(x_{-i}) - f(x)$ high to proposals have high likelihood**

- **Want $q(i|x)$ to have high entropy**

- **Need $q(i|x)$ to balance these for good sampling**

Proposals for Discrete Sampling

- **How to design $q(i|x)$? Acceptance prob:**

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

- **Want $f(x_{-i}) - f(x)$ high to proposals have high likelihood**

- **Want $q(i|x)$ to have high entropy**

- **Need $q(i|x)$ to balance these for good sampling**

- **Idea: let $q_\tau(i|x) = \dfrac{\exp\left(\frac{f(x_{-i}) - f(x)}{\tau}\right)}{Z(x)} = \dfrac{\exp\left(\frac{f(x_{-i}) - f(x)}{\tau}\right)}{\sum_{j=1}^{D}\exp\left(\frac{f(x_{-j}) - f(x)}{\tau}\right)}$**

Proposals for Discrete Sampling

- How to design $q(i \mid x)$? Acceptance prob:

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

- Want $f(x_{-i}) - f(x)$ high to proposals have high likelihood
- Want $q(i \mid x)$ to have high entropy
- Need $q(i \mid x)$ to balance these for good sampling
- Idea: let $q_\tau(i \mid x) = \dfrac{\exp\left(\dfrac{f(x_{-i}) - f(x)}{\tau}\right)}{Z(x)} = \dfrac{\;}{\sum}$

Tempered softmax over

$$\frac{f(x_{-i}) - f(x)}{\tau}$$

For possible i

Choosing τ

- **Rewrite acceptance probability w.r.t** $q_\tau(i \mid x)$

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

$$= \min\left\{\exp\left(\left(1 - \frac{2}{\tau}\right)(f(x_{-i}) - f(x))\right)\frac{Z(x_{-i})}{Z(x)}, 1\right\}$$

Choosing τ

- **Rewrite acceptance probability w.r.t** $q_\tau(i \mid x)$

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

$$= \min\left\{\exp\left(\underbrace{\left(1 - \frac{2}{\tau}\right)(f(x_{-i}) - f(x))}_{\text{Set } \tau = 2 \text{ to cancel}}\right)\frac{Z(x_{-i})}{Z(x)}, 1\right\}$$

Choosing τ

- **Rewrite acceptance probability w.r.t** $q_2(i \mid x)$

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

$$= \min\left\{\frac{Z(x_{-i})}{Z(x)}, 1\right\}$$

Choosing τ

- **Rewrite acceptance probability w.r.t $q_2(i \mid x)$**

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

$$= \min\left\{\frac{Z(x_{-i})}{Z(x)}, 1\right\}$$

Should be near 1

Choosing τ

- **Rewrite acceptance probability w.r.t** $q_2(i \,|\, x)$

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

$$= \min\left\{\frac{Z(x_{-i})}{Z(x)}, 1\right\}$$

- **Shown to be near optimal proposal which makes local moves (Zanella (2020))**

Difference Functions

- **Optimal proposal**

$$q(i \mid x) = \frac{\exp\left(\frac{f(x_{-i}) - f(x)}{2}\right)}{Z(x)}$$

Difference Functions

- Optimal proposal

$$q(i \mid x) = \frac{\exp\left(\frac{f(x_{-i}) - f(x)}{2}\right)}{Z(x)}$$

- To sample we must compute $f(x_{-i}) - f(x)$ for all $i \in [1, \ldots, D]$

- This means $O(D)$ function evals

Difference Functions

- Optimal proposal

$$q(i \mid x) = \frac{\exp\left(\frac{f(x_{-i}) - f(x)}{2}\right)}{Z(x)}$$

- To sample we must compute $f(x_{-i}) - f(x)$ for all $i \in [1, \ldots, D]$

- This means $O(D)$ function evals

- Slow if D big…

A surprisingly common structure

Bernoulli:	$\log p(x) = \theta x - \log Z$
Categorical:	$\log p(x) = \theta^T x - \log Z$
Ising:	$\log p(x) = x^T W x + b^T x - \log Z$
Potts:	$\log p(x) = \sum_{i=1}^{D} h_i^T x_i + \sum_{ij} x_i^T J_{ij} x_j - \log Z$
RBM:	$\log p(x) = \sum_{i} \textbf{softplus}(Wx + b)_i + c^T x$
HMM:	$\log p(x \mid y) = \sum_{t=1}^{T} x_t A x_{t-1} + \frac{(w^T x_t - y_t)^2}{\sigma^2}$
Deep EBM:	$\log p(x) = f_\theta(x) - \log Z$

A surprisingly common structure

Bernoulli:	$\log p(x) = \theta x - \log Z$
Categorical:	$\log p(x) = \theta^T x - \log Z$
Ising:	$\log p(x) = x^T W x + b^T x - \log Z$
Potts:	$\log p(x) = \sum_{i=1}^{D} h_i^T x_i + \sum_{ij} x_i^T J_{ij} x_j - \log Z$
RBM:	$\log p(x) = \sum_{i} \textbf{softplus}(Wx + b)_i + c^T x$
HMM:	$\log p(x \mid y) = \sum_{t=1}^{T} x_t A x_{t-1} + \frac{(w^T x_t - y_t)^2}{\sigma^2}$
Deep EBM:	$\log p(x) = f_\theta(x) - \log Z$

These are all continuous, differentiable functions of real-valued inputs!

Discrete structure is created by restricting input to $\{0,1\} \subset R$

Exploiting a surprisingly common structure

- **We can use Taylor-series to estimate**

$$f(x_{-i}) \approx (x_{-i} - x)^T \nabla_x f(x)$$

Exploiting a surprisingly common structure

- **We can use Taylor-series to estimate**

$$f(x_{-i}) \approx (x_{-i} - x)^T \nabla_x f(x)$$

- **For binary data, we estimate** $f(x_{-i}) - f(x)$ **for all** i**:**

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

- **Where** $\tilde{d}(x)[i] = f(x_{-i}) - f(x)$

- **Similar expression for categorical data**

Gibbs With Gradients

- **We propose a new sampler for discrete distributions**
- **We do Metropolis-Hastings with a proposal** $q(i \mid x)$
- **The proposal approximates:**

$$q(i \mid x) = \frac{\exp\left(\frac{f(x_{-i}) - f(x)}{2}\right)}{Z(x)}$$

Gibbs With Gradients

- **We propose a new sampler for discrete distributions**

- **We do Metropolis-Hastings with a proposal** $q(i \,|\, x)$

- **The proposal approximates:**

$$q(i \,|\, x) = \frac{\exp\left(\frac{f(x_{-i}) - f(x)}{2}\right)}{Z(x)}$$

- **Using a Taylor-series**

$$q(i \,|\, x) = \frac{\exp\left(\frac{(x_{-i} - x)^T \nabla_x f(x)}{2}\right)}{\tilde{Z}(x)}$$

- **Using** $O(1)$ **function evaluations!**

Gibbs With Gradients

- We propose a new sampler for discrete distributions
- We do Metropolis-Hastings with a proposal $q(i|x)$
- The proposal approximates:

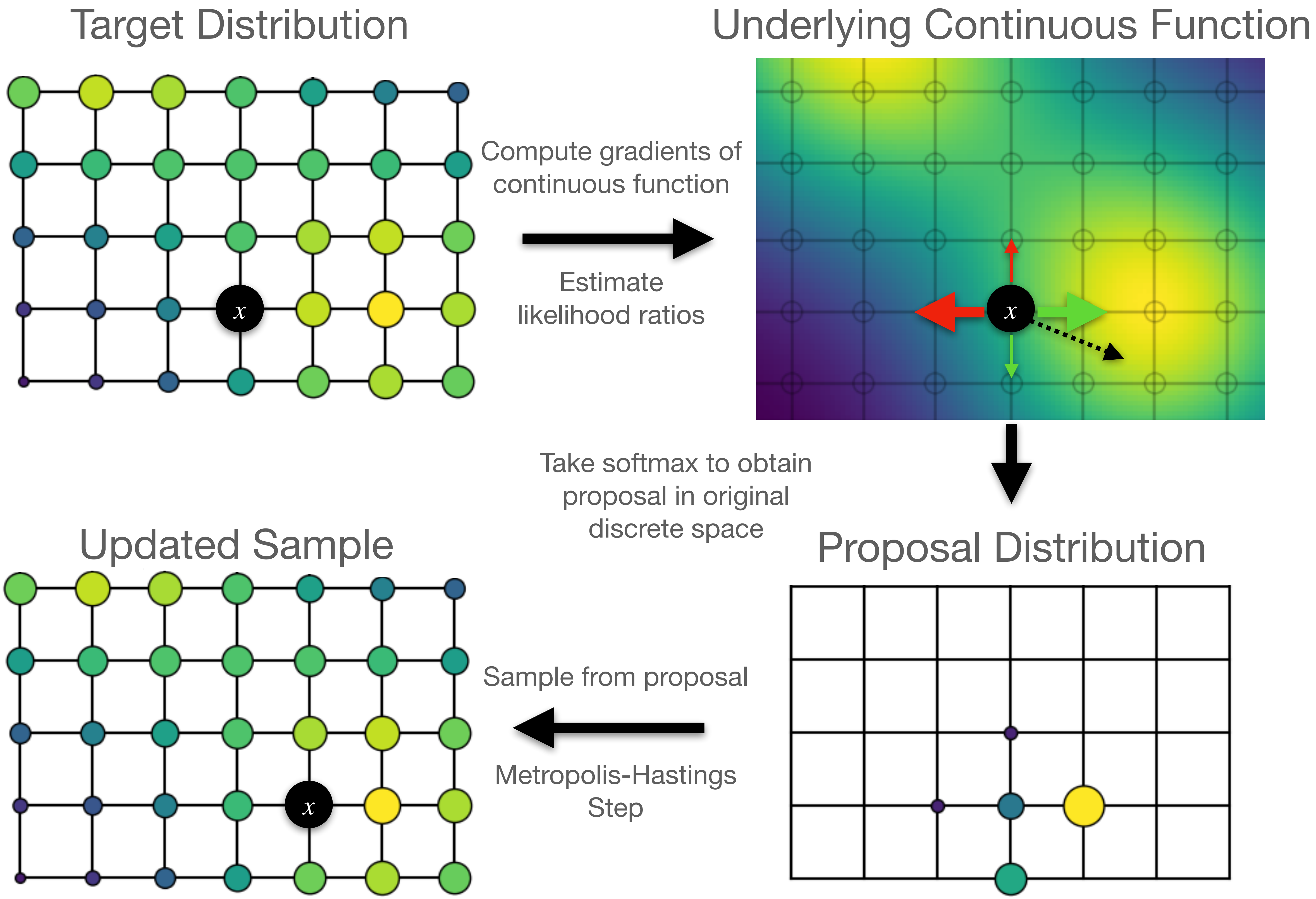
$$q(i|x) = \frac{\exp\left(\frac{f(x_{-i}) - f(x)}{2}\right)}{Z(x)}$$

- Using a Taylor-series

$$q(i|x) = \frac{\exp\left(\frac{(x_{-i} - x)^T \nabla_x f(x)}{2}\right)}{\tilde{Z}(x)}$$

- Using $O(1)$ function evaluations!
- Simple, efficient, no hyper-parameters(!!!!!)

Gibbs With Gradients (visually)



Target Distribution

Underlying Continuous Function

Compute gradients of continuous function

Estimate likelihood ratios

Take softmax to obtain proposal in original discrete space

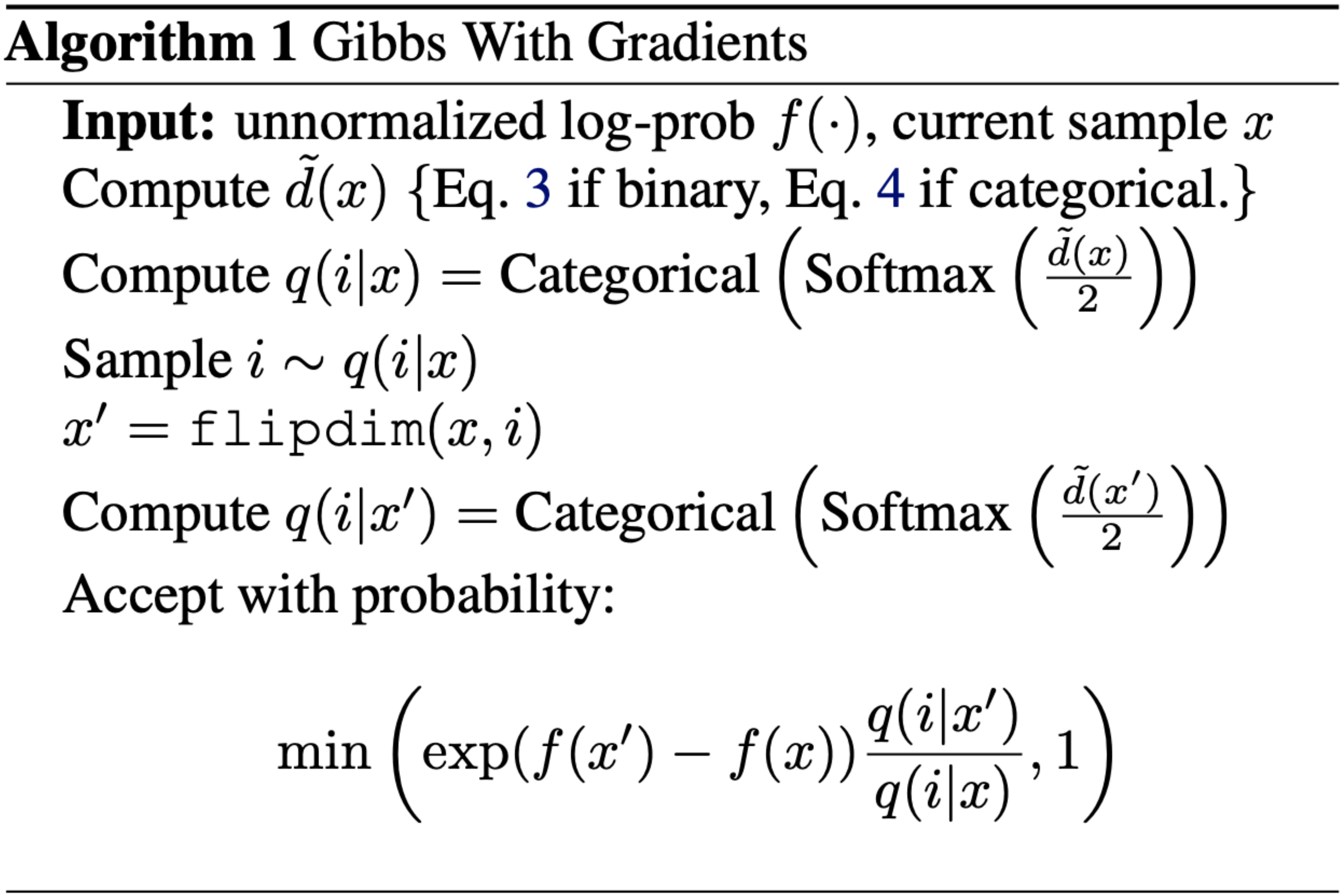
Proposal Distribution

Sample from proposal

Metropolis-Hastings Step

Updated Sample

Gibbs With Gradients (pseudo-code)

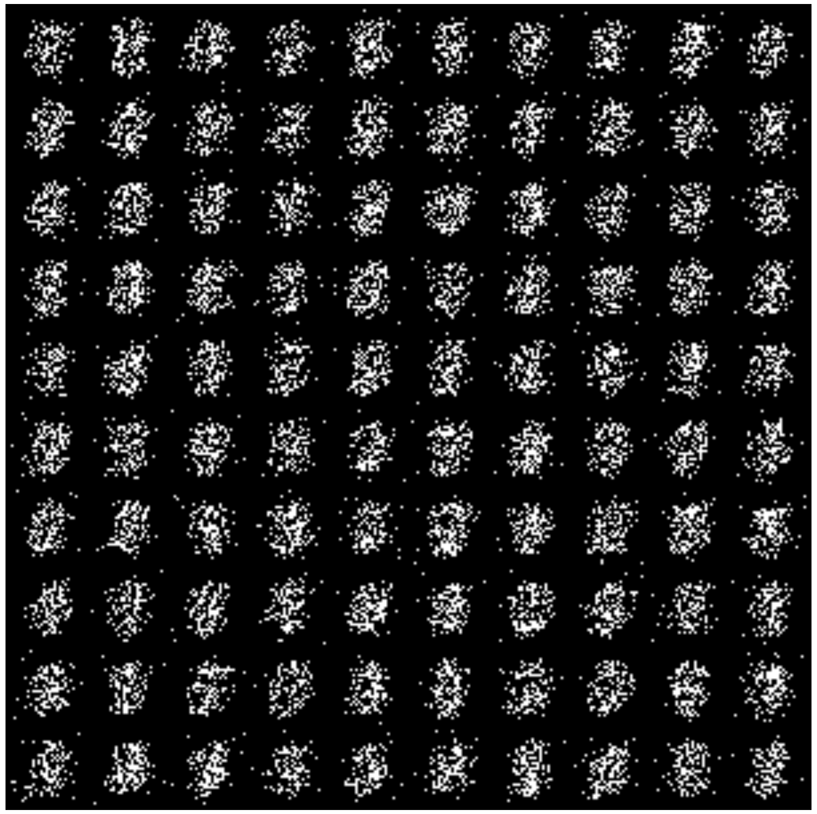


Algorithm 1 Gibbs With Gradients

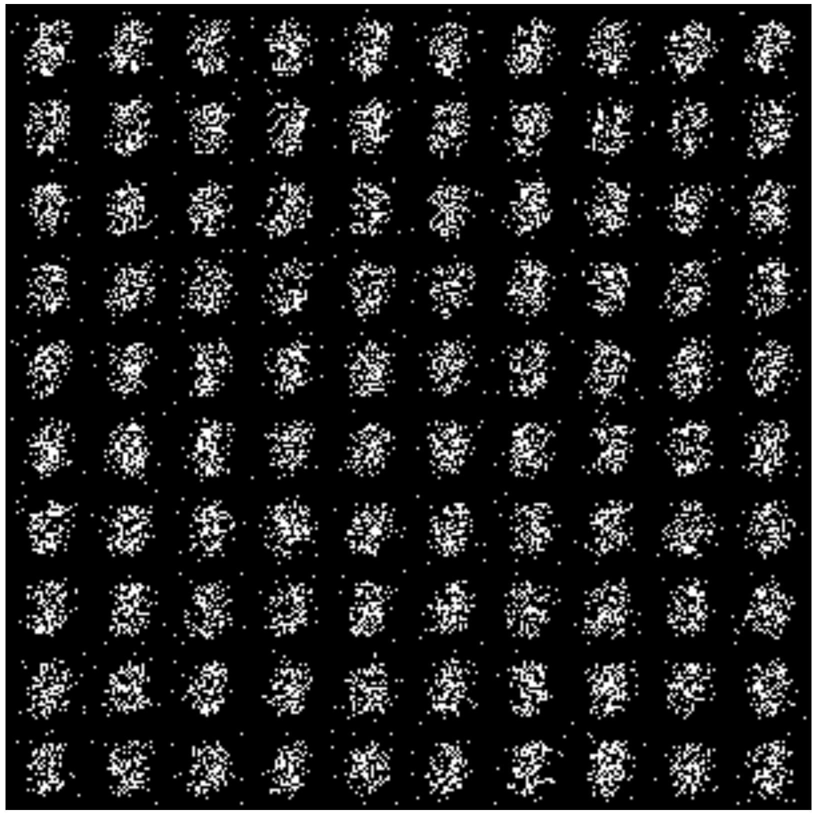
```
**Input:** unnormalized log-prob $f(\cdot)$, current sample $x$
Compute $\tilde{d}(x)$ {Eq. 3 if binary, Eq. 4 if categorical.}
Compute $q(i|x) = \text{Categorical}\left(\text{Softmax}\left(\frac{\tilde{d}(x)}{2}\right)\right)$
Sample $i \sim q(i|x)$
$x' = \texttt{flipdim}(x, i)$
Compute $q(i|x') = \text{Categorical}\left(\text{Softmax}\left(\frac{\tilde{d}(x')}{2}\right)\right)$
Accept with probability:
    $$\min\left(\exp(f(x') - f(x))\frac{q(i|x')}{q(i|x)}, 1\right)$$
```

RBM Sampling

GWG



Gibbs



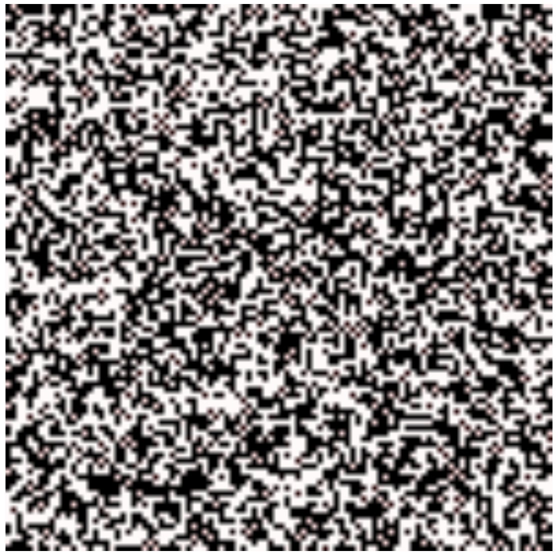
Ising Denoising

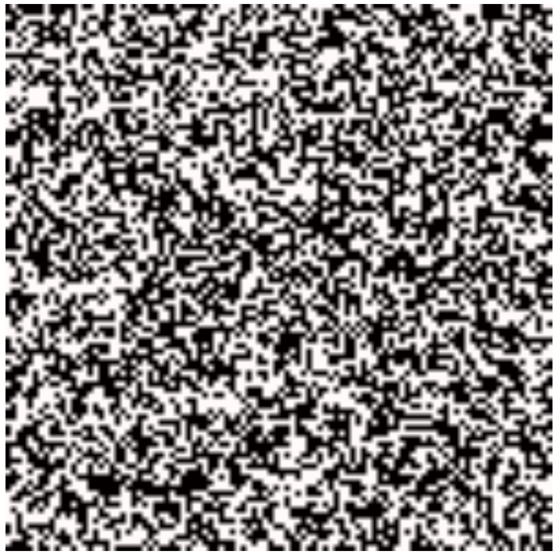
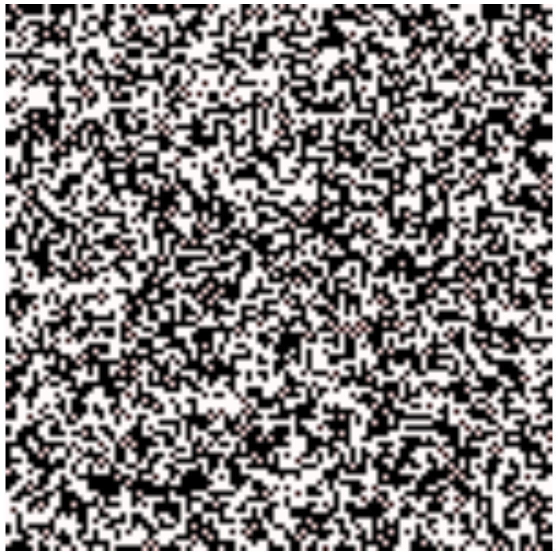
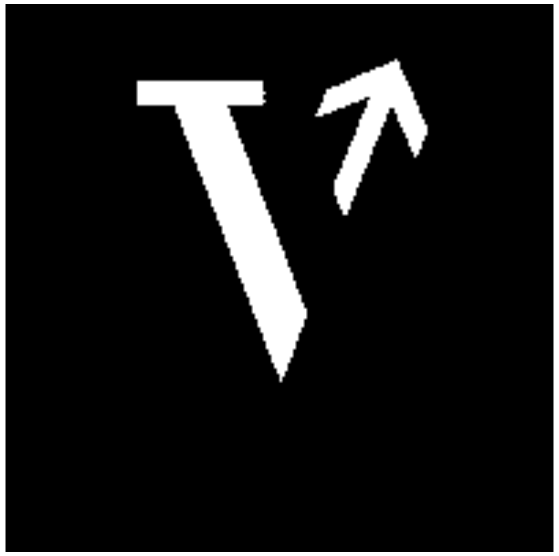
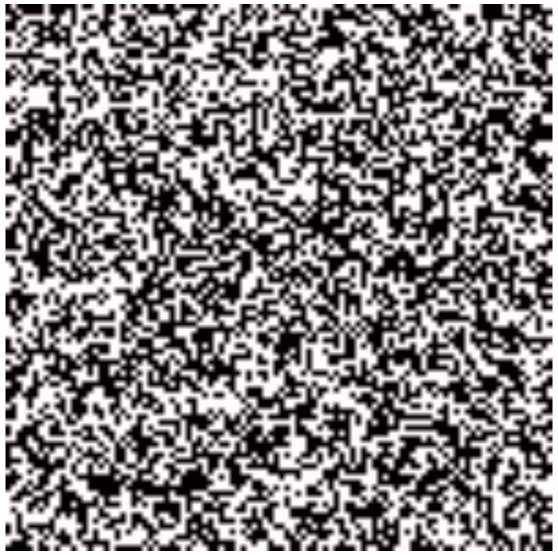
100x100 = 10,000 Variables!

GWG

Ground Truth

Gibbs



Training EBMs

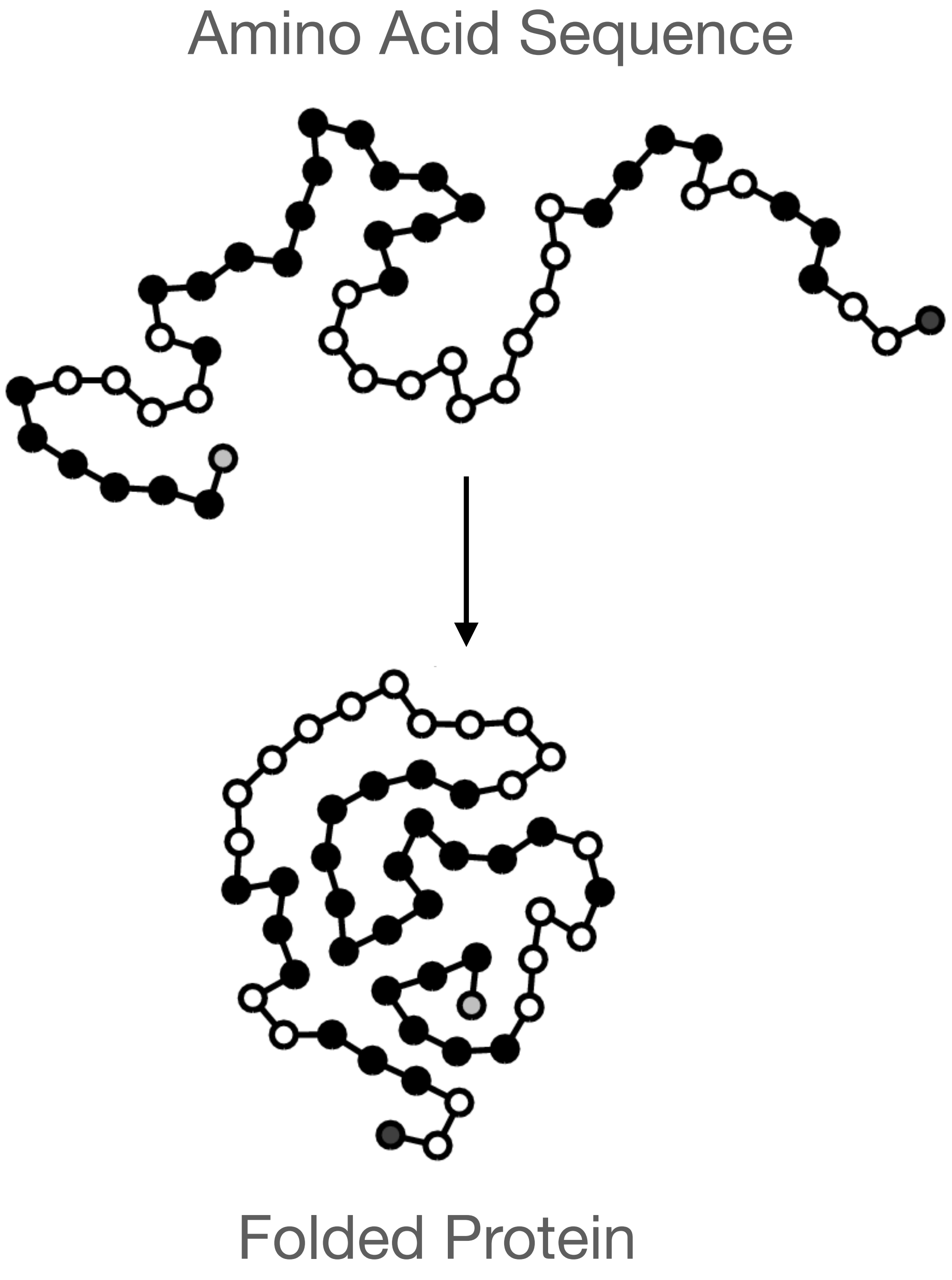
- **Recall** $\nabla_\theta \log p(x) = -\nabla_\theta E_\theta(x) + \mathbf{E}_{p_\theta(x)}[\nabla_\theta E_\theta(x)]$
- **So MCMC sampling can enable parameter inference for EBMs**

- **Protein Contact Prediction with Potts models**
- **Deep EBMs for discrete images**

Protein Contact Prediction

- **A protein x is a sequences of D amino acids** $x_i \in \{1,\ldots,20\}$

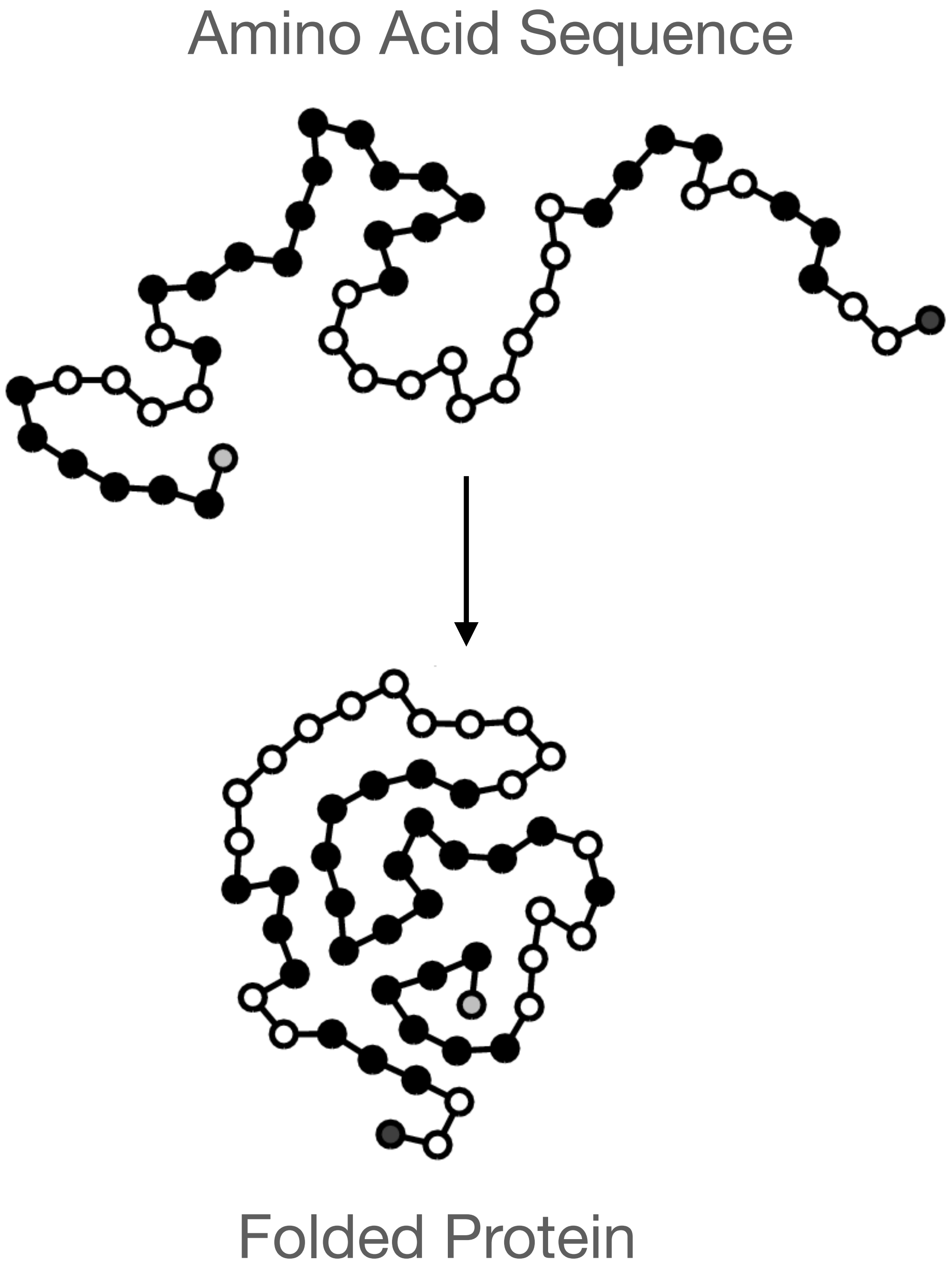
- **Want to know which x_i and x_j contact when folded**



Protein Contact Prediction

- **A protein x is a sequences of D amino acids** $x_i \in \{1,\ldots,20\}$
- **Want to know which x_i and x_j contact when folded**
- **Train Potts model:**

$$E_\theta(x) = \sum_{i=1}^{D} h_i^T x_i + \sum_{ij} x_j^T J_{ij} x_j$$



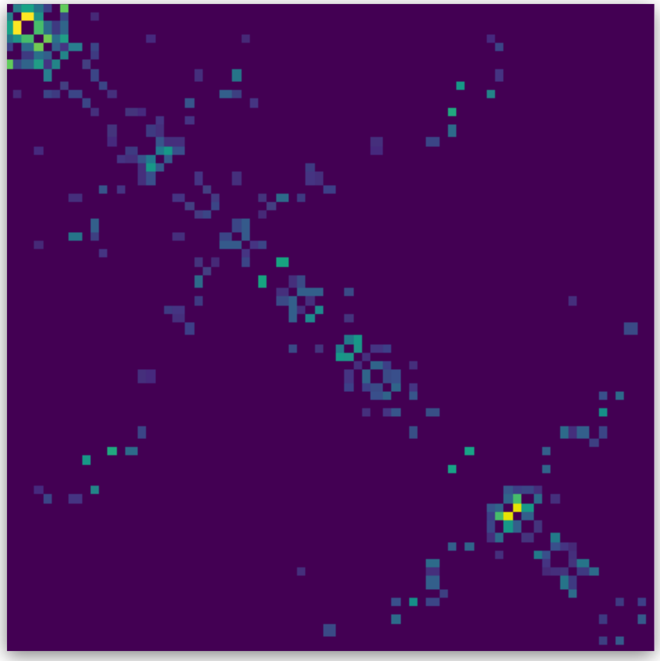
Protein Contact Prediction

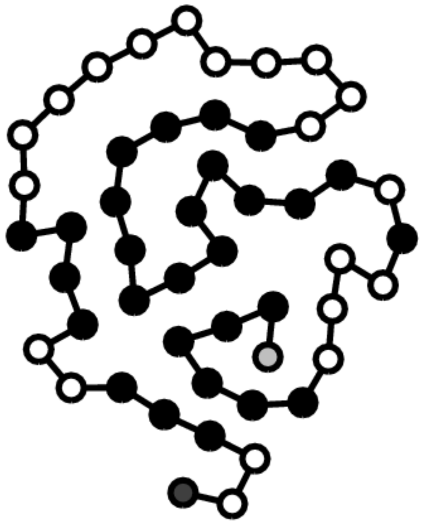
- **A protein x is a sequences of D amino acids** $x_i \in \{1, \ldots, 20\}$
- **Want to know which x_i and x_j contact when folded**
- **Train Potts model:**

$$E_\theta(x) = \sum_{i=1}^{D} h_i^T x_i + \sum_{ij} x_j^T J_{ij} x_j$$

- **Model J matrix learns interactions**

J





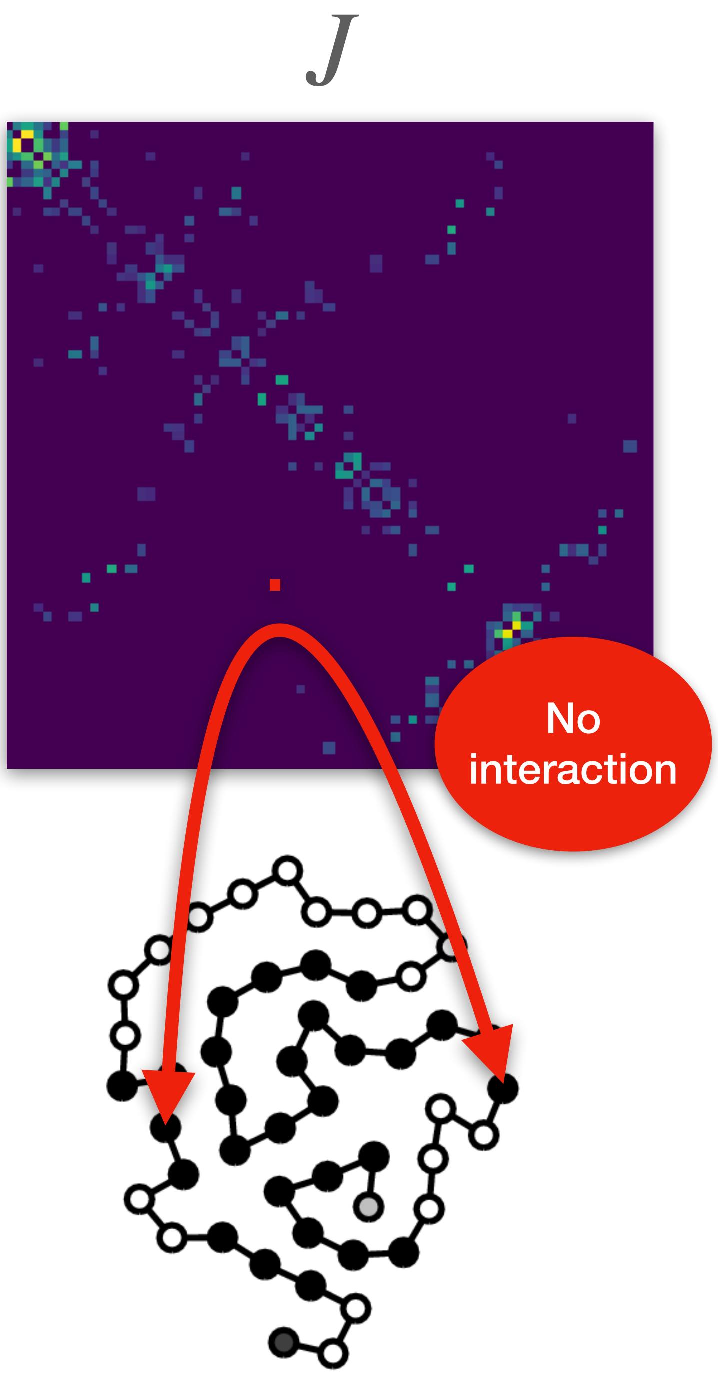
Folded Protein

Protein Contact Prediction

- **A protein x is a sequences of D amino acids** $x_i \in \{1,\ldots,20\}$
- **Want to know which x_i and x_j contact when folded**
- **Train Potts model:**

$$E_\theta(x) = \sum_{i=1}^{D} h_i^T x_i + \sum_{ij} x_j^T J_{ij} x_j$$

- **Model J matrix learns interactions**
- **Make predictions with interaction strength**



Folded Protein

Protein Contact Prediction

- A protein x is a sequences of D amino acids $x_i \in \{1,\ldots,20\}$

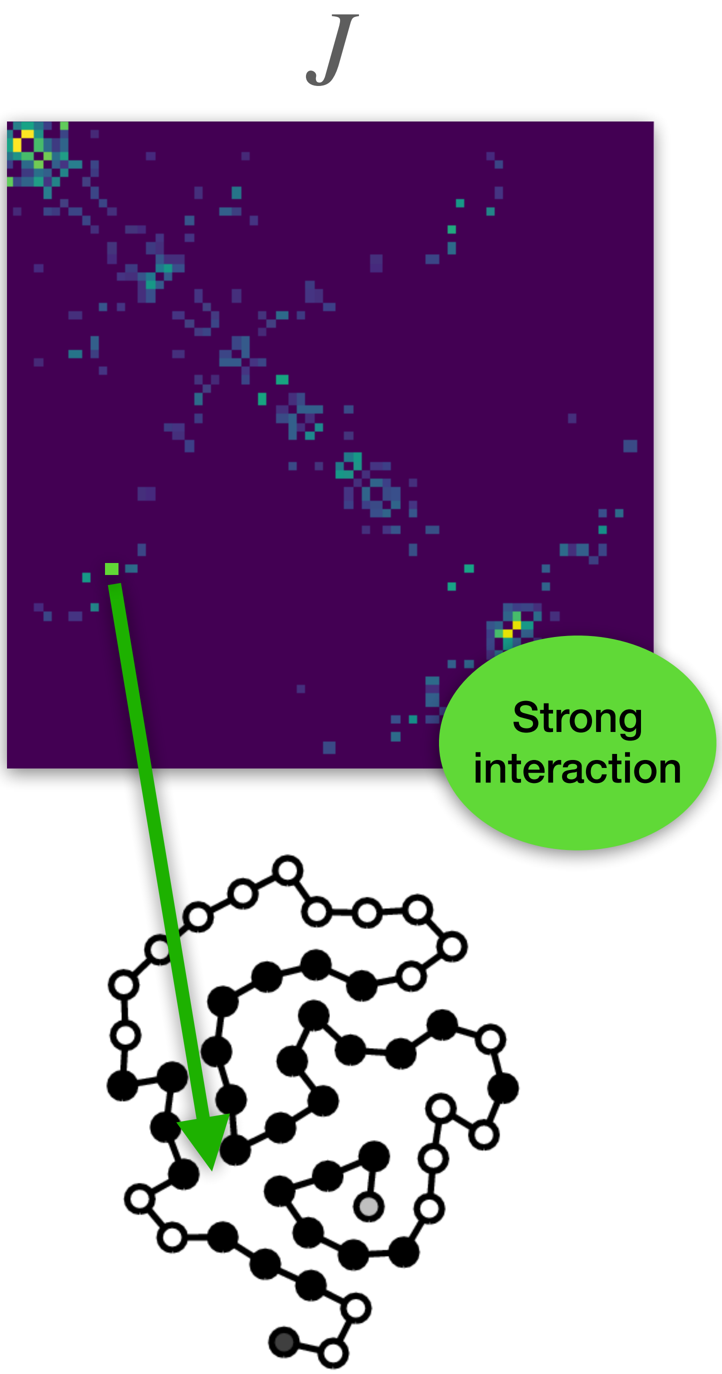
- Want to know which x_i and x_j contact when folded

- Train Potts model:

$$E_\theta(x) = \sum_{i=1}^{D} h_i^T x_i + \sum_{ij} x_j^T J_{ij} x_j$$

- Model J matrix learns interactions

- Make predictions with interaction strength

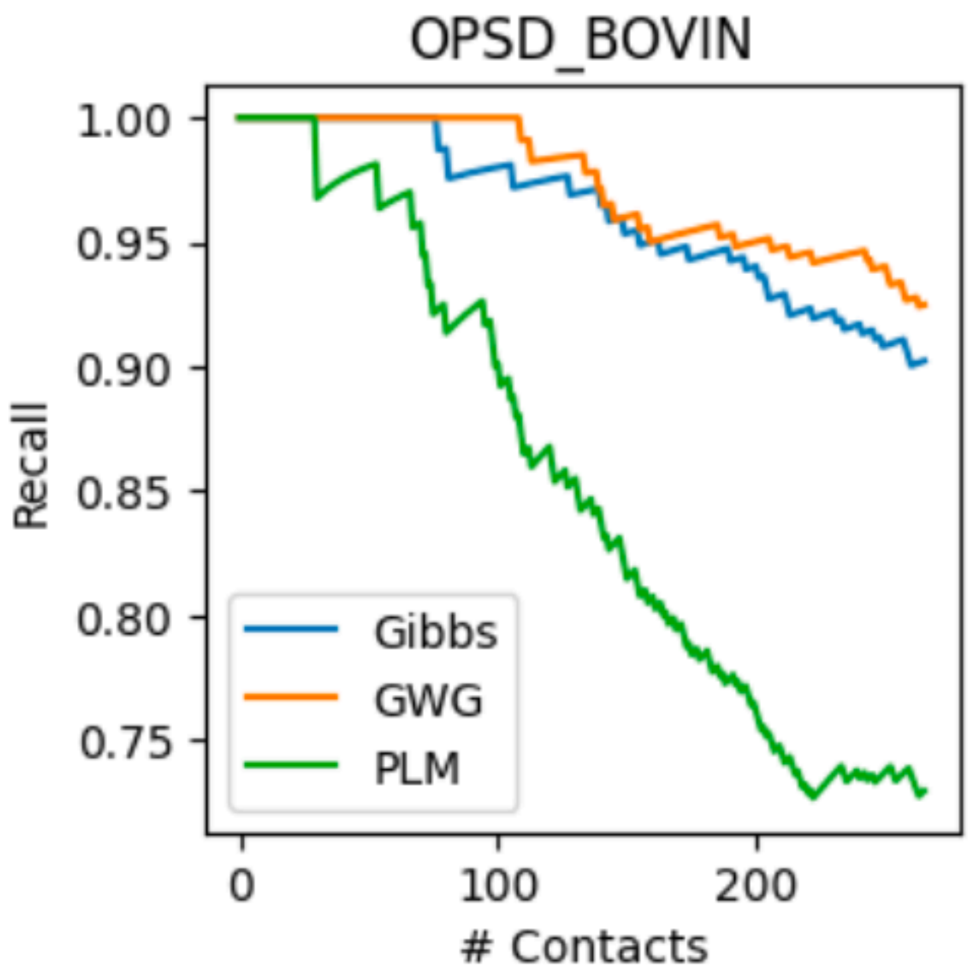


Folded Protein

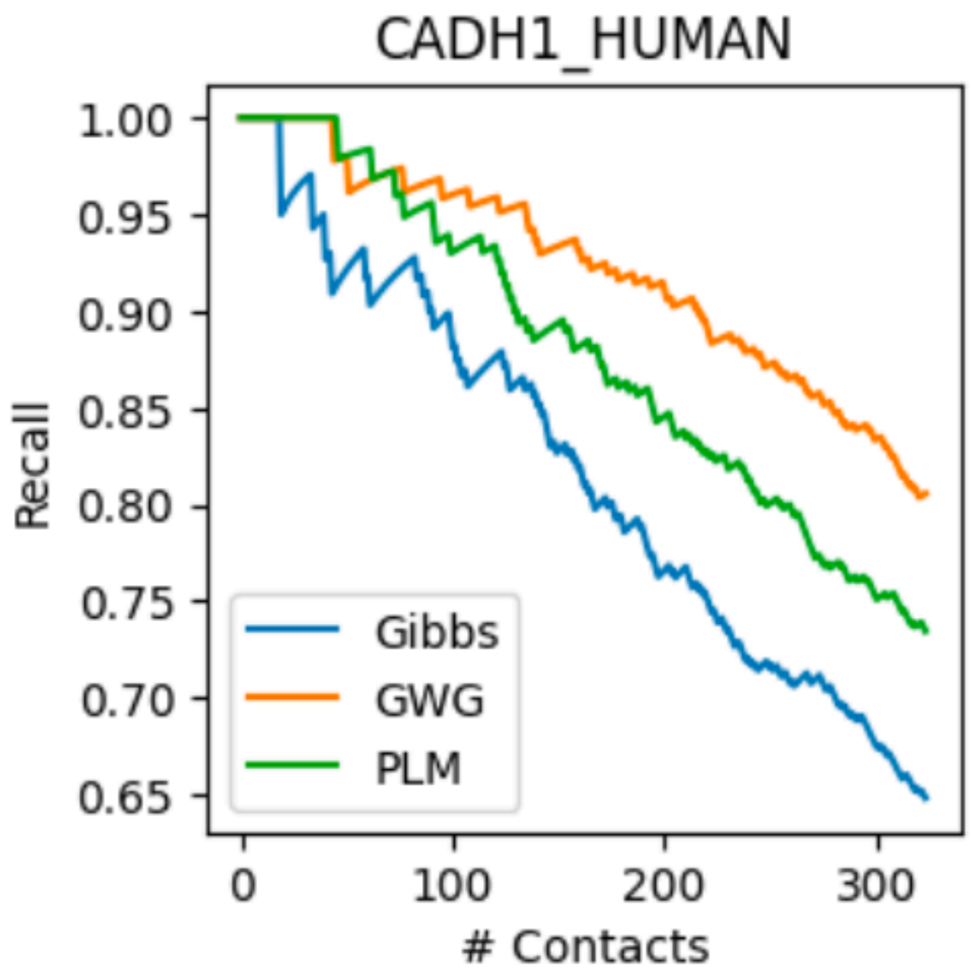
Protein Contact Prediction

- **Compare:**
  - **Maximum likelihood using Gibbs, GWG**
  - **Pseudo-likelihood Maximization (PLM) (standard practice)**

$D = 348$



$D = 882$

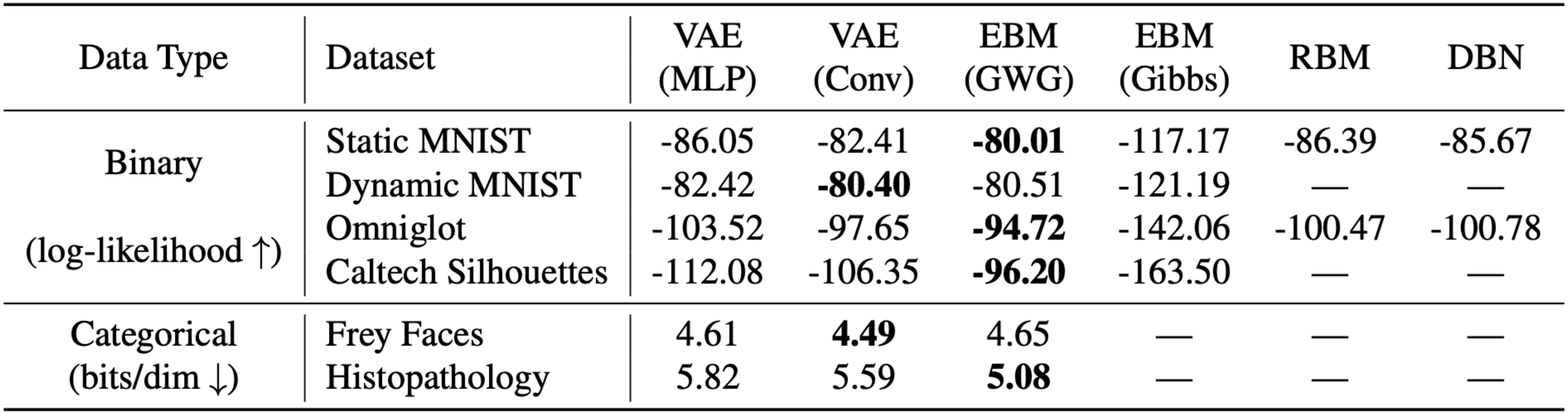


Deep EBMs for Discrete Data

- Recent successful EBMs use neural network energy: $p_\theta(x) = \dfrac{e^{f_\theta(x)}}{Z}$
- We train Deep ResNet EBMs on binary and categorical image data
- Binary pixel values are 0, 1
- For categorical each pixel is 1-of-256 way categorical
  - This means 256 function evals for 1 step of Gibbs!

Deep EBMs for Discrete Data

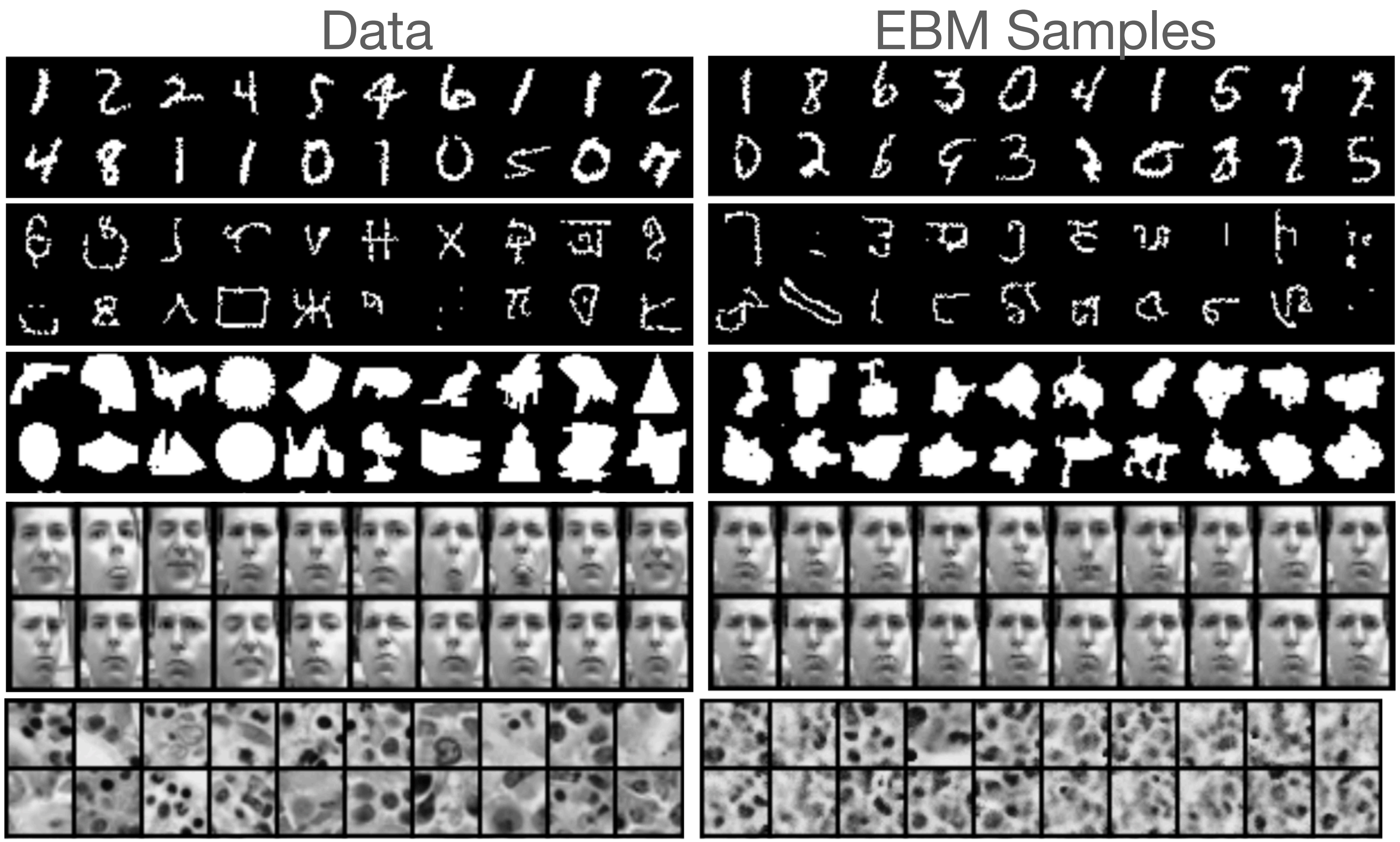
- **Train with PCD**
- **Outperforms VAEs, RBM, and Deep belief net in log-likelihood**
- **GWG greatly outperforms Gibbs on binary data and Gibbs is completely unable to train because of high cost per-iteration**



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

Deep EBMs for Discrete Data

- **Train with PCD**
- **Outperforms VAEs, RBM, and Deep belief net in log-likelihood**
- **GWG greatly outperforms Gibbs on binary data and Gibbs is completely unable to train because of high cost per-iteration**



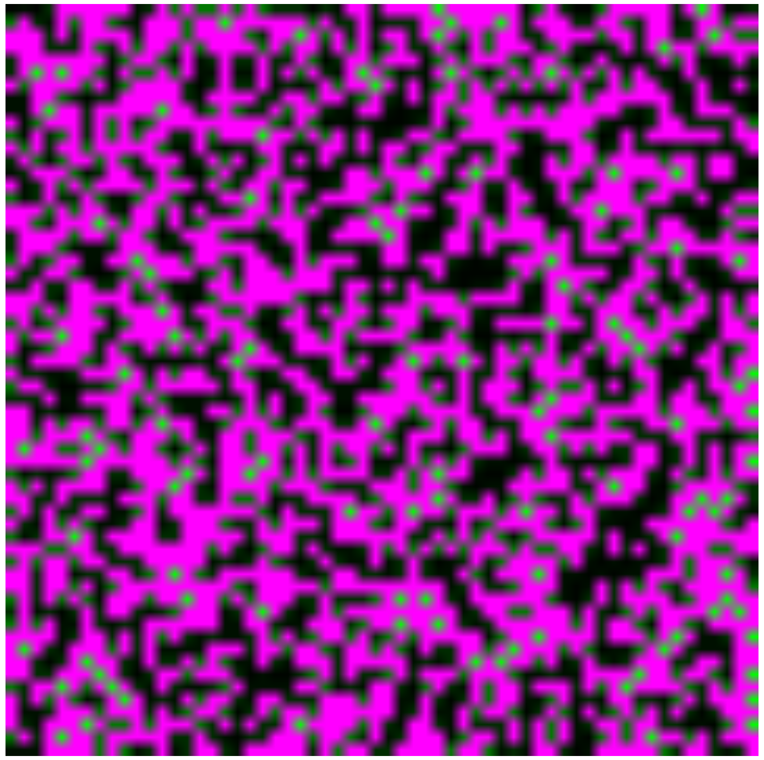
Additional results

- See paper for additional results on:
    - Text EBMs
    - Structure inference in Ising models
    - Additional sampling experiments

Next Steps

- Improvements for large categoricals (text)
- New approximations when gradients can't be computed
- Apply gradients to:
  - Discrete Score Matching
  - Discrete Stein Discrepancies
- Integrate into probabilistic programming frameworks

Thanks!



- Thanks for having me, much love to my co-authors!
- Code available: github.com/wgrathwohl/GWG_release
- You can find me at
  - @wgrathwohl or
  - wgrathwohl@cs.toronto.edu

Kevin Swersky

Milad Hashemi

David Duvenaud

Chris Maddison