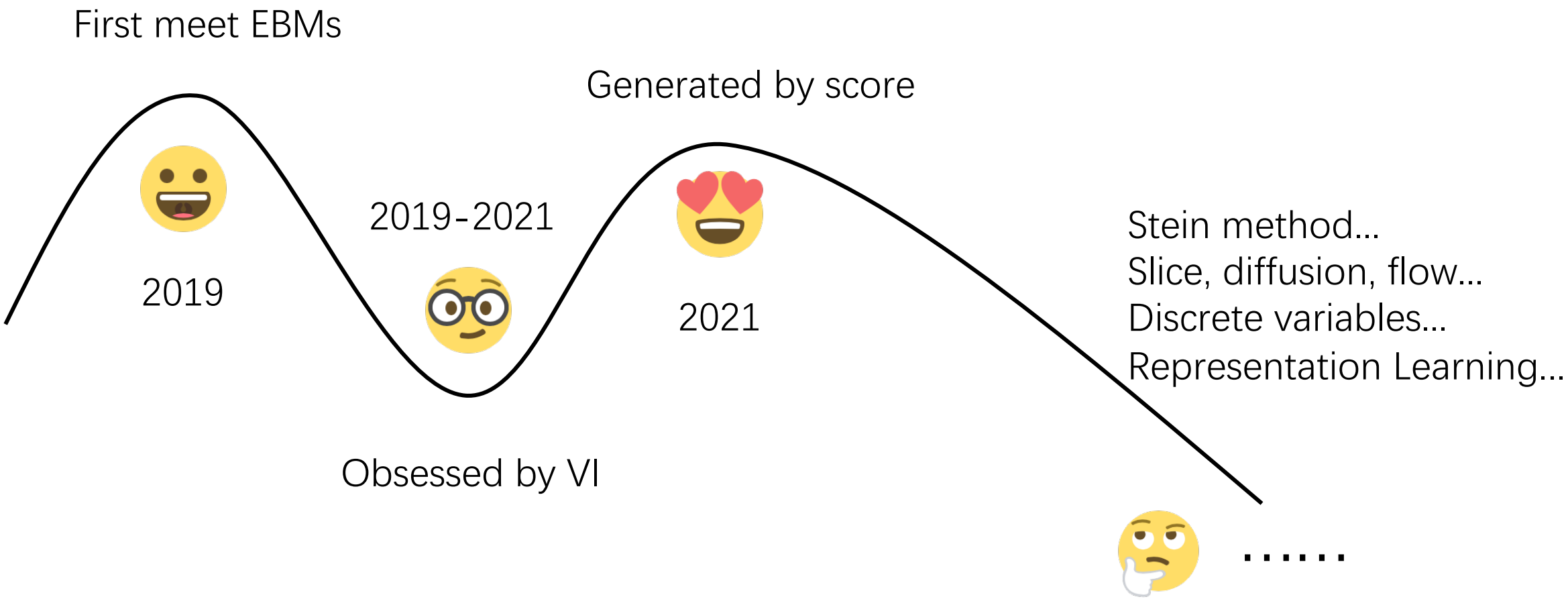
EBM Learning and Inference:
A Retrospection and Beyond

Zijing Ou
ouzj@mail2.sysu.edu.cn

My Story on EBM



Binary representation learning on text

- Generative hashing

$$p_\theta(x, z) = p_\theta(x|z)p(z)$$

$$x = \{w_1, w_2, \ldots, w_{|x|}\} \qquad z = \{0,1\}^m$$

One-hot representation　　Binary codes

- Softmax decoder

$$p_\theta(w_i|z) = \frac{\exp(z^T E w_i + b_i)}{\sum_{j=1}^{|V|} \exp(z^T E w_j + b_j)}$$

$p_\theta(x|z) = \prod_{i=1}^{|x|} p_\theta(w_i|z)$ (iid. assumption)

Variational inference

- Evidence lower bound

$$\log p(x) = ELBO + KL(q_\phi(z|x)||p(z))$$

$$\geq ELBO = E_{q_\phi(z|x)}\left[\log \frac{p_\theta(x|z)p(z)}{q_\phi(z|x)}\right]$$

- Consistent learning & inference

$$\mathcal{L} = E_{p(x)}E_{q_\phi(z|x)}\left[\log \frac{p_\theta(x|z)p(z)}{q_\phi(z|x)}\right]$$

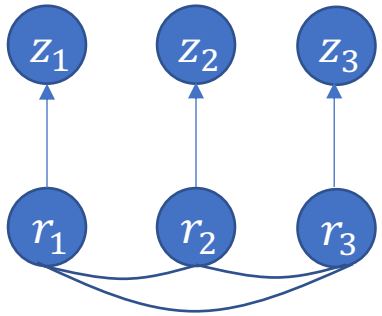
Independent among data

$$q_\phi(z|x) \coloneqq \mathcal{N}(z|\mu_\phi(x), diag(\sigma_\phi^2(x)))$$

Independent among bits

Our focus: breaking the independence assumption!!!

Boltzmann machine as posterior



- Lower rank perturbation

"covariance matrix"

$$b(z) = \frac{1}{Z} exp(\frac{1}{2} z^T \Sigma z + \mu^T z)$$

$$\Sigma = D + UU^T$$ (diagonal + low-rand perturbation matrix)

- Reparameterization

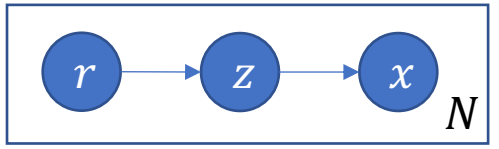
$$b(z) = \int p(z|r)p(r)dr$$

$$p(r) = \frac{1}{z} \prod_{i=1}^{m} (e^{r_i} + 1) \mathcal{N}(r; \mu, \Sigma)$$

$$p(z|r) = \prod_{i=1}^{m} Bernoulli(\sigma(r_i))$$

$$z \sim b(z) = r \sim p(r), z \sim p(z|r)$$

$$q_\phi(z|x) = E_{q_\phi(r|x)}[Bernoulli(z; r)]$$



Boltzmann machine as variational posterior

Cancel out partition function

- Evidence lower bound

$$\mathcal{L} = E_{q_\phi(z|x)}\left[\log \frac{p_\theta(x|z)p(z)}{e^{-E_\phi(z)}}\right] + \log Z_\phi$$

$$E_\phi(z) \coloneqq -\frac{1}{2} z^T \Sigma_\phi(x) z - \mu_\sigma^T(x) z$$

- An asymptotically-exact lower bound

$$h_k(z) \coloneqq \frac{1}{\mathrm{k}} \sum_{i=1}^{k} p(z|r^{(i)})$$

$$\widetilde{\mathcal{L}_k} = \mathcal{L} - E_{q_\phi\left(r^{(1\ldots k)}\middle|x\right)}[KL(h_k(z)||q_\phi(z|x))]$$

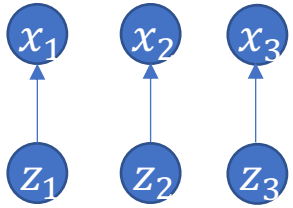
$$= E_{q_\phi(z|x)}[\log p_\theta(x,z)] - E_{q_\phi\left(r^{(1\ldots k)}\middle|x\right)} E_{h_k(z)}[\log h_k(z)]$$

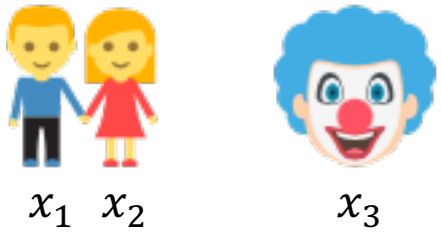
$$\widetilde{\mathcal{L}_k} < \widetilde{\mathcal{L}_{k+1}} \quad \lim_{k\to\infty} \widetilde{\mathcal{L}_k} = \mathcal{L}$$

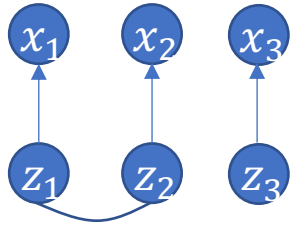
Ranganath et al. Hierarchical Variational Models. ICML 2016
Yin and Zhou. Semi-Implicit Variational Inference. ICML 2018
Zheng and Su. Generative Semantic Hashing Enhanced via Boltzmann Machines. ACL 2020

How to introduce correlation among data points?

Data dependence prior







$$p(X,Z) = p_\theta(X|Z)p_I(Z)$$

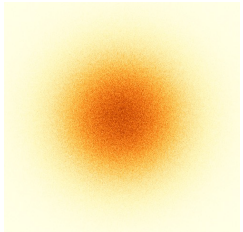
$$X = \{x_1, x_2, x_3\}$$

$$p(X,Z) = p_\theta(X|Z)p_G(Z)$$

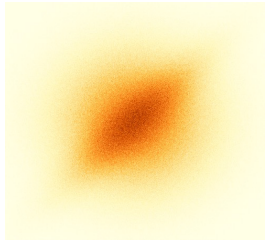
data independence prior

$$p_I(Z) = \mathcal{N}(Z; I \otimes I_d)$$

denote as Σ_I



$p_I(Z)$



$p_G(Z)$

data dependence prior

$$p_G(Z) = \mathcal{N}(Z; (I + \lambda A) \otimes I_d)$$

denote as Σ_G

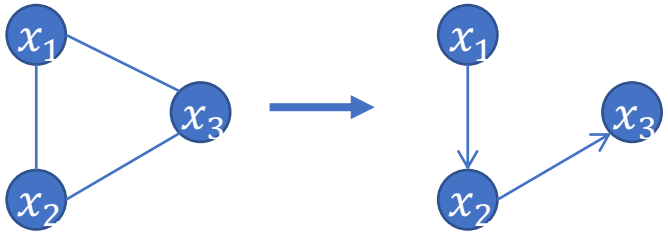
$$\Sigma_I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Sigma_G = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathcal{L} := E_{q_\phi(Z|X)}[\log p_\theta(X|Z)] - KL(q_\phi(Z|X)||p_G(Z))$$

"curse-of-dimensionality"

Spanning-tree approximations



$$p_{\mathrm{T}}(Z) = \prod_{i \in \mathcal{V}} p_G(z_i) \prod_{(i,j) \in \mathcal{E}} \frac{p_G(z_i, z_j)}{p_G(z_i) p_G(z_j)}$$

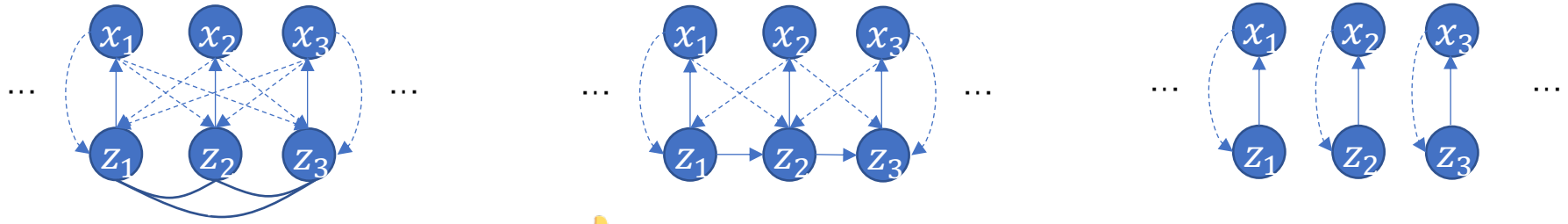
$$q_{\mathrm{T}}(Z|X) = \prod_{i \in \mathcal{V}} q_\phi(z_i|x_i) \prod_{(i,j) \in \mathcal{E}} \frac{q_\phi(z_i, z_j|x_i, x_j)}{q_\phi(z_i|x_i) q_G(z_j|x_j)}$$

Variance of $q_\phi(z_i, z_j|x_i, x_j)$:

$\gamma_{ij} \in (0, 1)$: positive correlated

$$\begin{bmatrix} diag(\sigma_i^2) & diag(\gamma_{ij} \odot \sigma_i \odot \sigma_j) \\ diag(\gamma_{ij} \odot \sigma_i \odot \sigma_j) & diag(\sigma_j^2) \end{bmatrix}$$

Spanning-tree approximations



Graph driven

👍Tree approximation

Mean field

$$\mathcal{L}_T = \sum_{i \in \mathcal{V}} E_{q_\phi}[\log p_\theta(x_i|z_i)] - KL(q_\phi(z_i)||p_G(z_i))$$

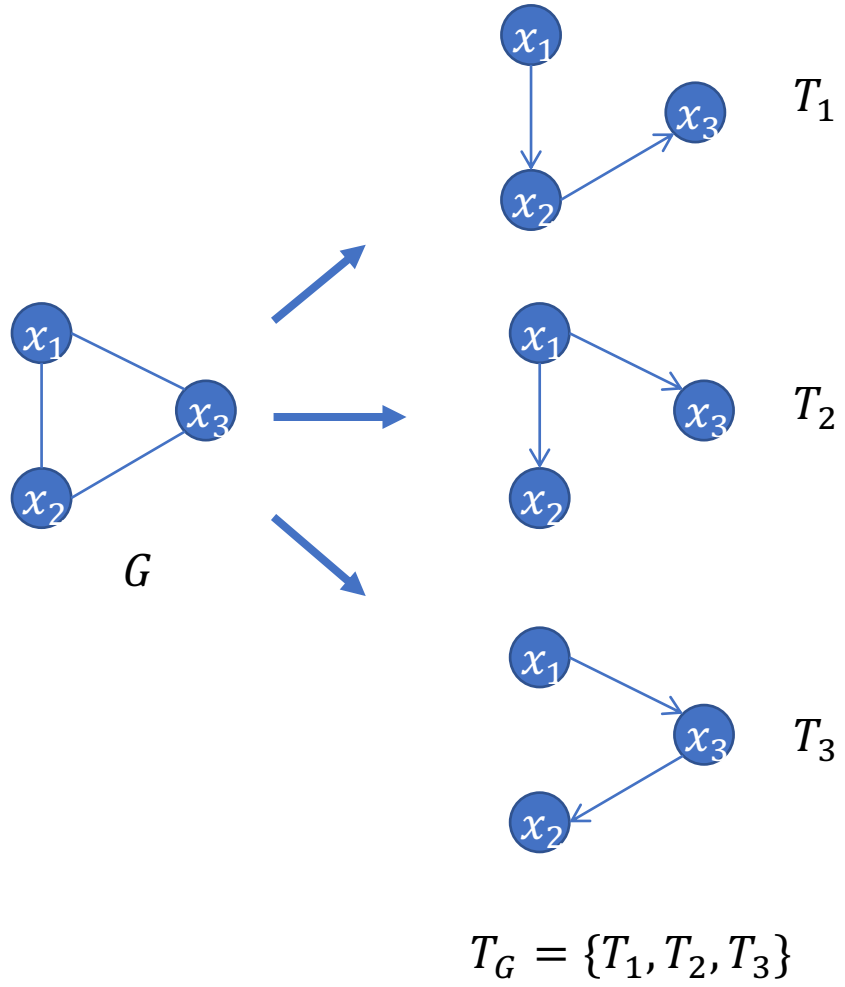
$$= \sum_{(i,j) \in \mathcal{E}} \left(KL\left(q_\phi(z_i, z_j)||p_G(z_i, z_j)\right) - KL\left(q_\phi(z_i)||p_G(z_i)\right) - KL\left(q_\phi(z_j)||p_G(z_j)\right)\right)$$

Extending to multiple trees

$$p_{MT}(Z) = \frac{1}{M}\sum_{T \in T_G} p_T(Z)$$

$$q_{MT}(Z) = \frac{1}{M}\sum_{T \in T_G} q_T(Z|X)$$

$$\widetilde{\mathcal{L}_{MT}} = \frac{1}{M}\mathcal{L}_T$$



$$T_G = \{T_1, T_2, T_3\}$$

Looking forward

Structured representation learning

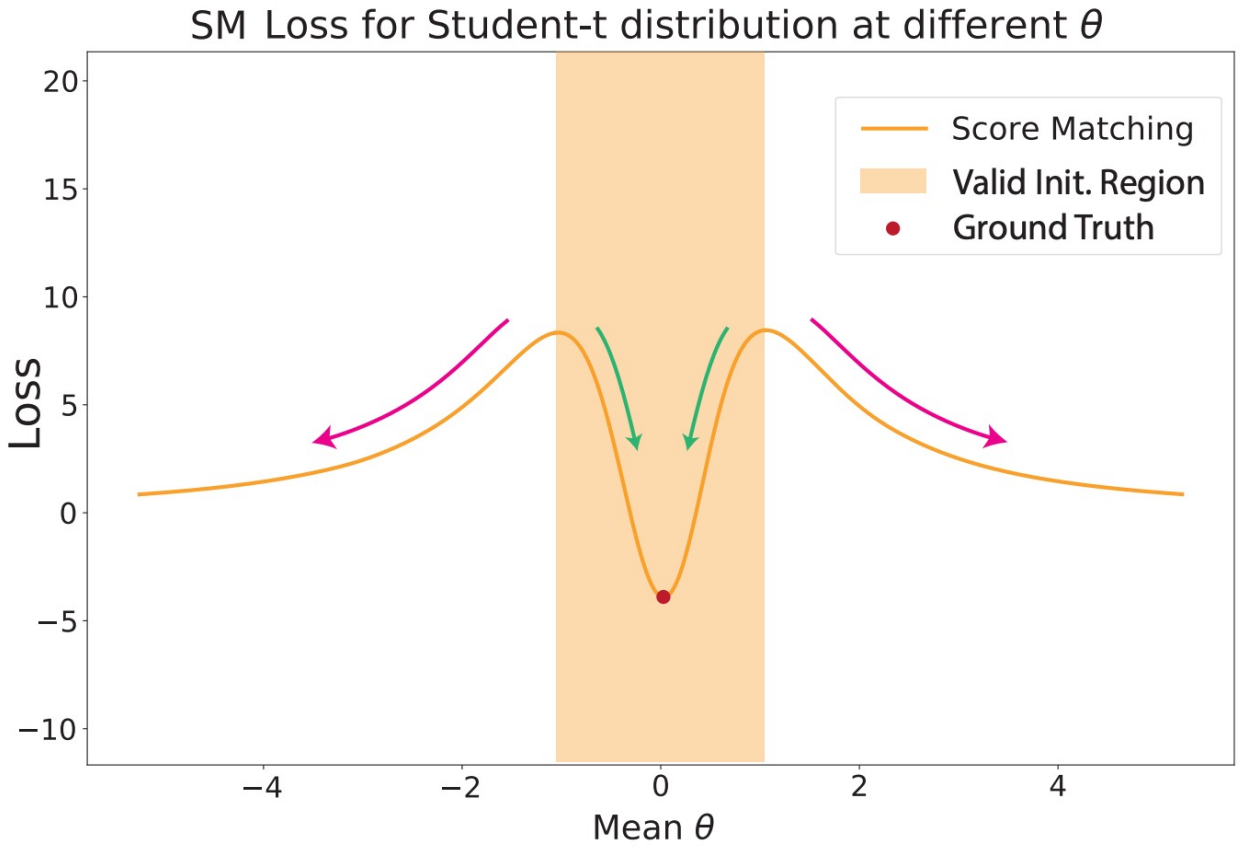
- Expressive prior & posterior
- Combine with advances in GNNs
- Graphical models meet Deep Learning (GNN as message passing)

Enhancing score matching via flow

- Fitting student-t with score matching

$$F(p, q) = \frac{1}{2} E_p[\|\nabla_x \log q(x) - \nabla_x \log p(x)\|^2]$$

$p_{data}(x) = \mathrm{T}_{v=5}(0, 0.3)$ $\quad$ $q_\theta(x) = \mathrm{T}_{v=5}(\theta, 0.3)$

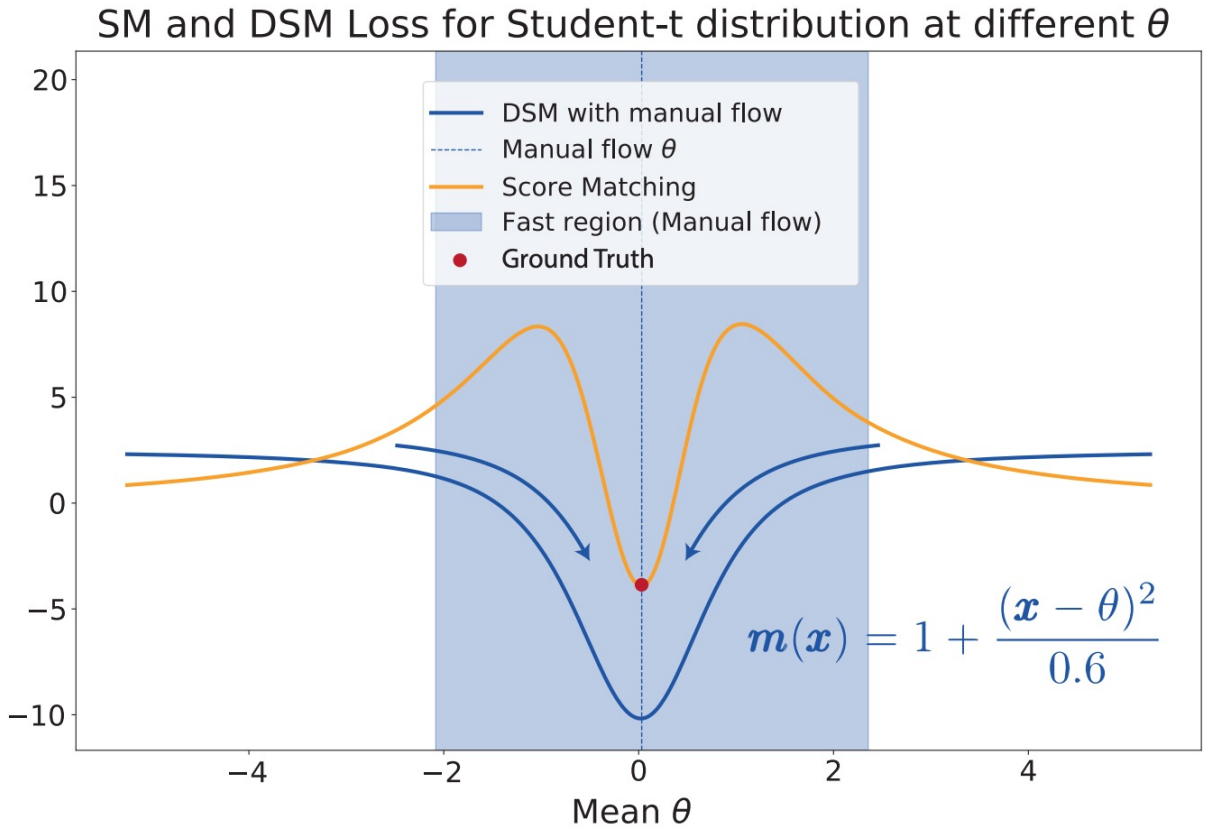


Enhancing score matching via flow

Fisher Divergence → With diffusion matrix $\boldsymbol{m}(\boldsymbol{x})$ → **Diffusion Score Matching**

$$F(p,q)=\frac{1}{2}E_p[\|\nabla_x \log q(x)-\nabla_x \log p(x)\|^2]$$

$$F_m(p,q)=\frac{1}{2}E_p[\left\|\boldsymbol{m}(\boldsymbol{x})^{\mathrm{T}}(\nabla_x \log q(x)-\nabla_x \log p(x))\right\|^2]$$



Barp et al. Minimum Stein Discrepancy Estimators. NIPS 2019

Gong and Li. Interpreting diffusion score matching using normalizing flow. ICML 2021 INNF+ workshop

Enhancing score matching via flow

- Interpreting DSM using flows

$$p_X, q_X \xrightarrow[\mathrm{m}(x) = (\nabla_x T(X))^{-1}]{\text{invertible flow } \mathrm{Y} = T(X)} p_Y, q_Y$$

$$p_Y(y) = p_X\left(T^{-1}(y)\right)\left|det\,\frac{\partial T^{-1}(y)}{\partial y}\right| \qquad q_Y(y) = q_X\left(T^{-1}(y)\right)\left|det\,\frac{\partial T^{-1}(y)}{\partial y}\right|$$

Fisher Divergence $F(p_X, q_X)$ $\xrightarrow{\mathrm{m}(x) = (\nabla_x T(X))^{-1}}$ **Diffusion Score Matching** $F(p_Y, q_Y) = F_m(p_X, q_X)$

Enhancing score matching via flow

- Interpreting DSM using flows

$$p_X, q_X \xrightarrow[\mathrm{m}(x) = (\nabla_x T(X))^{-1}]{\text{invertible flow } \mathrm{Y} = T(X)} p_Y, q_Y$$

$$p_Y(y) = p_X\left(T^{-1}(y)\right)\left|det\,\frac{\partial T^{-1}(y)}{\partial y}\right| \qquad q_Y(y) = q_X\left(T^{-1}(y)\right)\left|det\,\frac{\partial T^{-1}(y)}{\partial y}\right|$$

$$F(p_Y, q_Y) \coloneqq \frac{1}{2}\mathrm{E}_{\mathrm{q_Y}}[\|\nabla_Y \log p_Y(y) - \nabla_y \log q_Y(y)\|^2]$$

$$= \frac{1}{2}E_{q_Y}\left[\left\|\nabla_y \log p_X\left(T^{-1}(y)\right) - \nabla_y \log q_X\left(T^{-1}(y)\right)\right\|^2\right]$$

$$= \frac{1}{2}E_{q_Y}\left[\left\|\nabla_y T^{-1}(y)^T\left(\nabla_{T^{-1}(y)} \log p_X\left(T^{-1}(y)\right) - \nabla_{T^{-1}(y)} \log q_X\left(T^{-1}(y)\right)\right)\right\|^2\right]$$

$$= \frac{1}{2}E_{q_X}[\|\nabla_x T(x)^{-T}(\nabla_x \log p_X(x) - \nabla_x \log q_X(x))\|^2]$$

Enhancing score matching via flow

- Interpreting DSM using flows

$$p_X, q_X \xrightarrow[\mathrm{m}(x) = (\nabla_x T(X))^{-1}]{\text{invertible flow } \mathrm{Y} = T(X)} p_Y, q_Y$$

$$p_Y(y) = p_X\left(T^{-1}(y)\right)\left|det\,\frac{\partial T^{-1}(y)}{\partial y}\right| \qquad q_Y(y) = q_X\left(T^{-1}(y)\right)\left|det\,\frac{\partial T^{-1}(y)}{\partial y}\right|$$

$$F(p_Y, q_Y) := \frac{1}{2}\mathrm{E}_{q_Y}[\|\nabla_Y \log p_Y(y) - \nabla_y \log q_Y(y)\|^2]$$

$$= \frac{1}{2}E_{q_Y}\left[\left\|\nabla_y \log p_X\left(T^{-1}(y)\right) - \nabla_y \log q_X\left(T^{-1}(y)\right)\right\|^2\right]$$

$$= \frac{1}{2}E_{q_Y}\left[\left\|\nabla_y T^{-1}(y)^T\left(\nabla_{T^{-1}(y)} \log p_X\left(T^{-1}(y)\right) - \nabla_{T^{-1}(y)} \log q_X\left(T^{-1}(y)\right)\right)\right\|^2\right]$$

$$= \frac{1}{2}E_{q_X}[\|\nabla_x T(x)^{-T}(\nabla_x \log p_X(x) - \nabla_x \log q_X(x))\|^2]$$

Enhancing score matching via flow

- Interpreting DSM using flows

$$p_X, q_X \xrightarrow[\mathrm{m}(x) = (\nabla_x T(X))^{-1}]{\text{invertible flow } \mathrm{Y} = T(X)} p_Y, q_Y$$

$$p_Y(y) = p_X\left(T^{-1}(y)\right)\left|det\,\frac{\partial T^{-1}(y)}{\partial y}\right| \qquad q_Y(y) = q_X\left(T^{-1}(y)\right)\left|det\,\frac{\partial T^{-1}(y)}{\partial y}\right|$$

$$\begin{aligned} F(p_Y, q_Y) &:= \frac{1}{2}\mathrm{E}_{q_Y}[\|\nabla_Y \log p_Y(y) - \nabla_y \log q_Y(y)\|^2] \\ &= \frac{1}{2}E_{q_Y}\left[\left\|\nabla_y \log p_X\left(T^{-1}(y)\right) - \nabla_y \log q_X\left(T^{-1}(y)\right)\right\|^2\right] \\ &= \frac{1}{2}E_{q_Y}\left[\left\|\nabla_y T^{-1}(y)^T\left(\nabla_{T^{-1}(y)} \log p_X\left(T^{-1}(y)\right) - \nabla_{T^{-1}(y)} \log q_X\left(T^{-1}(y)\right)\right)\right\|^2\right] \\ &= \frac{1}{2}E_{q_X}[\|\nabla_x T(x)^{-T}(\nabla_x \log p_X(x) - \nabla_x \log q_X(x))\|^2] \end{aligned}$$

Enhancing score matching via flow

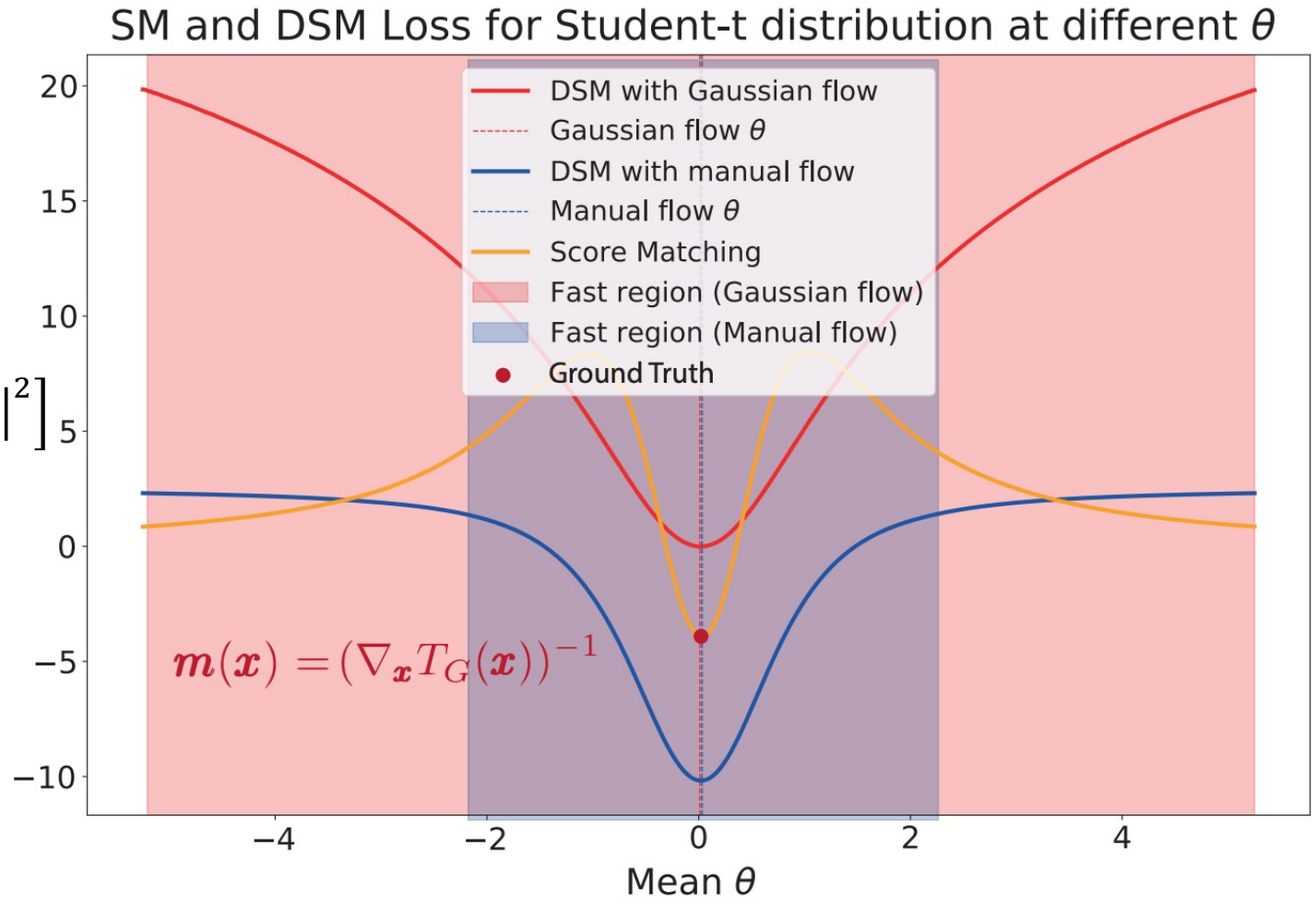
p_X, q_X —invertible flow $\mathrm{Y} = T(X)$, $\mathrm{m}(x) = (\nabla_x T(X))^{-1}$→ p_Y, q_Y

$p_{data}(x) = \mathrm{T}_{v=5}(0, 0.3)$ $\quad$ $q_\theta(x) = \mathrm{T}_{v=5}(\theta, 0.3)$

$$F_m(p, q) = \frac{1}{2} E_p \left[\left\| \boldsymbol{m}(\boldsymbol{x})^{\mathrm{T}} (\nabla_x \log q(x) - \nabla_x \log p(x)) \right\|^2 \right]$$

DSM: $\boldsymbol{m}(\boldsymbol{x}) = \left(\mathbf{1} + \frac{(\boldsymbol{x}-\boldsymbol{\theta})^2}{\mathbf{0.6}}\right)$

Gaussian flow: $\boldsymbol{T_G(x) = F_G^{-1} \circ F_\theta(x)}$



How to design flow?

Sampling from discrete EBMs

$$p_\theta(x) = \frac{exp(f(x))}{Z}, x \in \{0,1\}^D, Z = \int f(x)dx$$

- Metropolis-Hastings sampling

Proposal distribution $q(x_{-i}|x)$:

sample index $i \sim q(i|x)$, set $x_{-i} = flip\_\dim(x, i)$

Acceptance rate:

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

How to design $q(i|x)$?

Sampling from discrete EBMs

$$p_\theta(x) = \frac{exp(f(x))}{Z}, x \in \{0,1\}^D, Z = \int f(x)dx$$

- Metropolis-Hastings sampling

Acceptance rate:

$$\min\left\{\underbrace{\exp\big(f(x_{-i}) - f(x)\big)}_{\text{high to proposals have high likelihood}} \frac{q(i|x_{-i})}{q(i|x)}, 1\right\}$$

computational complexity

$\boldsymbol{O(D)}$ 😳

$$\boldsymbol{q(i|x) \propto \exp(f(x_{-i}) - f(x))}$$

Sampling from discrete EBMs

$$p_\theta(x) = \frac{exp(f(x))}{Z}, x \in \{0,1\}^D, Z = \int f(x)dx$$

- Metropolis-Hastings sampling

Proposal distribution

$$q(i|x) \propto exp(f(x_{-i}) - f(x))$$

computational complexity

$\boldsymbol{O(D)}$ 😳

First order Taylor approximation

$\boldsymbol{1\ if\ x_i = 0;\ -1\ if\ x_i = 1}$

$$f(x_{-i}) - f(x) \approx (x_{-i} - x)^T \nabla_x f(x)$$
$$= \left(-(2x-1) \odot \nabla_x f(x)\right)_i$$

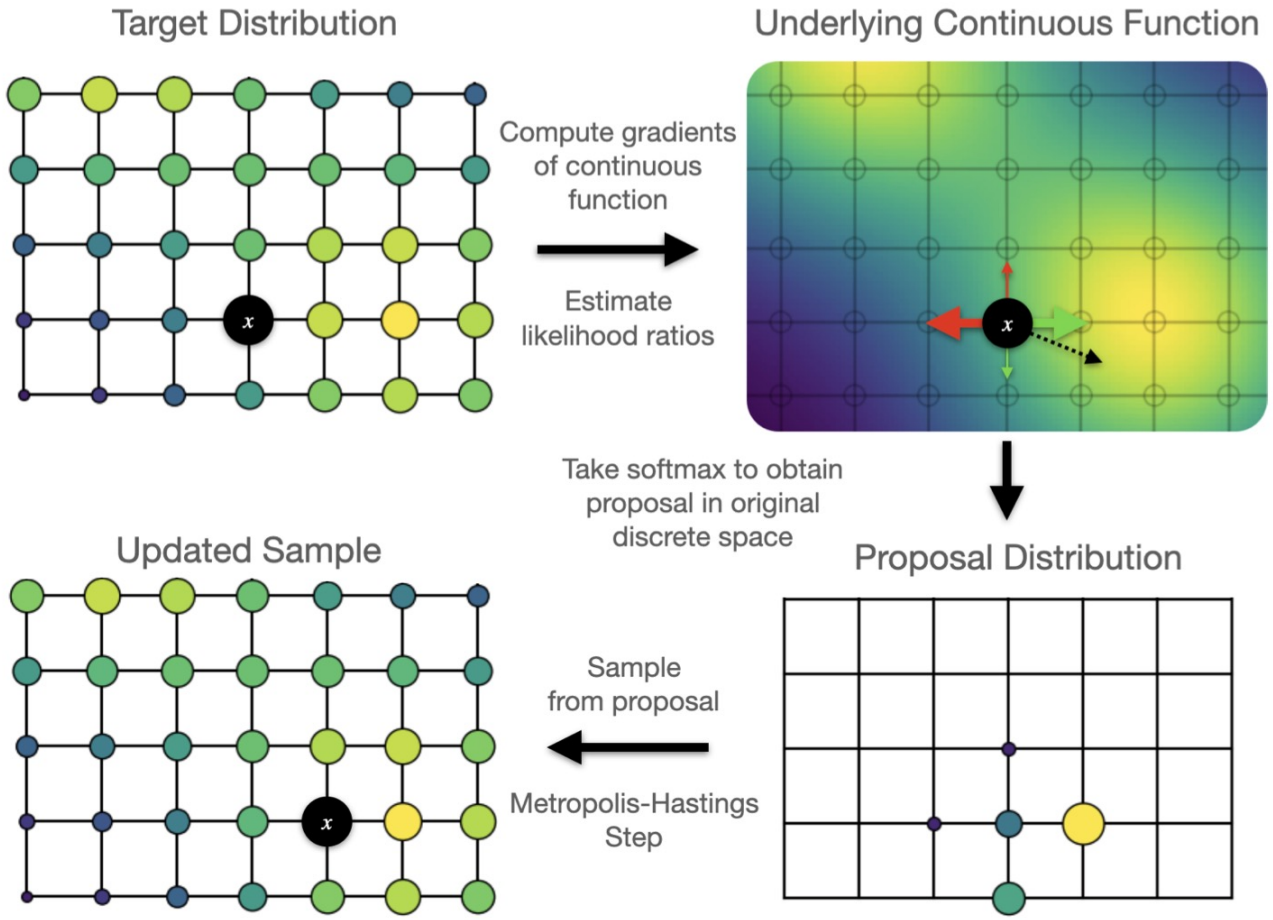
$$q(i|x) \propto \exp\left(\left(-(2x-1) \odot \nabla_x f(x)\right)_i\right)$$

computational complexity

$\boldsymbol{O(1)}$ 😀

Sampling from discrete EBMs

$$q(i|x) \propto \exp\left(\left(-(2x-1)\odot\nabla_x f(x)\right)_i\right)$$



Grathwohl et al. Oops I Took A Gradient: Scalable Sampling for Discrete Distributions. ICML 2021

Applying gradients to ratio matching

- Ratio matching

computational complexity

$$F_R(p;q) = E_p\left[\sum_i \left(\log\frac{q(x_{-i})}{q(x)} - \log\frac{p(x_{-i})}{p(x)}\right)^2\right] \quad \boldsymbol{O(D)}$$

cancel out partition function

- Continuous gradients relaxation

$$\log p(x_{-i}) - \log p(x) \propto (x_{-i} - x)^T \nabla_x \log p(x)$$

$$= (-(2x-1) \odot \nabla_x \log p(x))_i$$

computational complexity

$$\boldsymbol{F_{RL}(p;q) = E_p\left[\left\|-(2x-1)^T(\nabla_x \log q(x) - \nabla_x \log p(x))\right\|^2\right]} \quad \boldsymbol{O(1)}$$

Diffusion/flow score matching: $\boldsymbol{m(x) = \nabla_x T(x)^{-1} = -(2x-1)}$

Applying gradients to ratio matching

$$p(x) \propto \exp(\frac{1}{2}x^T\Sigma x + \mu^T x) \qquad x = \{0,1\}^m$$

$$q_\theta(x) \propto \exp\left(\frac{1}{2}x^T\Sigma_\theta x + \mu_\theta^T x\right)$$

- Ratio matching

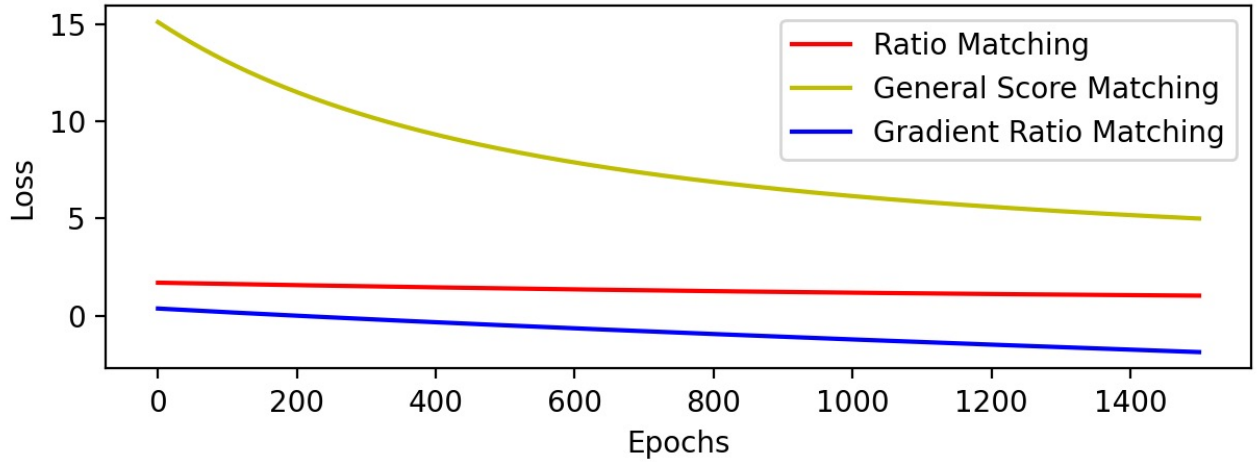
$$\mathcal{L} = E_p\left[\sum_{i=1}^{m} f^2\left(\frac{q(x)}{q(x_{-i})}\right)\right], f(x) = \frac{1}{1+x^2}$$

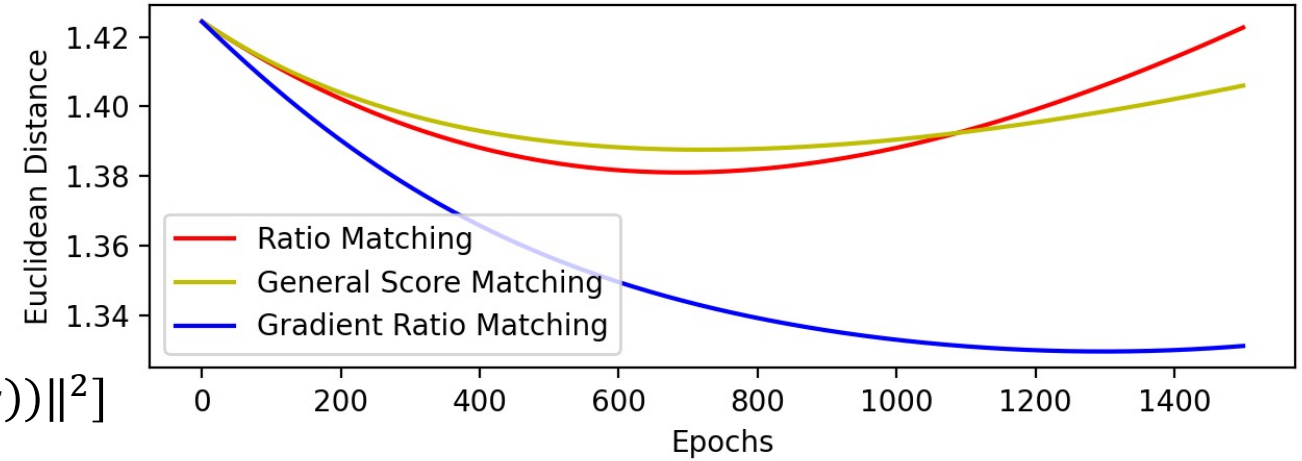
- General score matching

$$\mathcal{L} = E_p\left[\sum_{i=1}^{m}\left(\frac{q(x)}{q(x_{-i})}\right)^2\right]$$

- Gradient ratio matching

$$\mathcal{L} = E_p[\|-(2x-1)^T(\nabla_x \log q(x) - \nabla_x \log p(x))\|^2]$$





Looking forward

- EBM learning
  - Efficient and accurate gradient estimation (slice, diffusion, stein method)
  - Discrete variables (relaxed by gradients, enhanced by flow)
- EBM inference
  - Sampling from EBMs (graph generation)
  - Latent variables (representation learning)