Probabilistic Learning and Inference Using Stein Discrepancy 😄

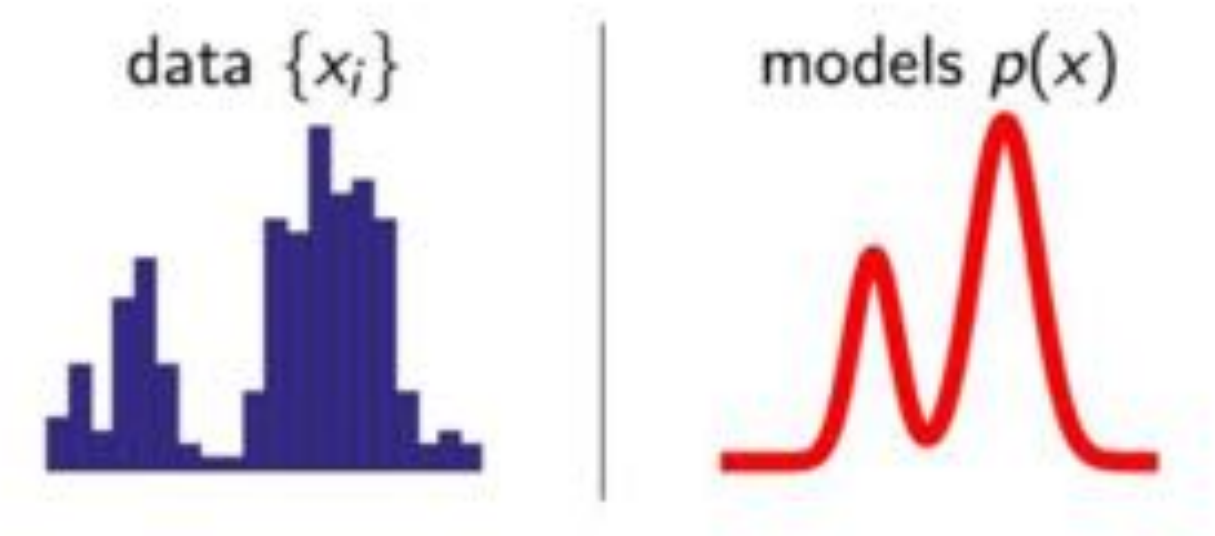
Reference

- Stein Variational Gradient Descent: Theory and Applications
- A Short Introduction to Kernelized Stein Discrepancy
- /~lqiang/PDF/steinslides16.pdf

QianWu , Oct.15, 2019

Data-Model Discrepancy 🧐

MCMC / Variational Inference/GAN



- Learning: Given $\{xi\}$, find an optimal p:

$$\min\_p \mathbf{D}(\{xi\}, p)$$

- Sampling (or numerical quadrature): Given p, find optimal $\{xi\}$:

$$\min\_\{xi\}, \mathbf{D}(\{xi\}, p)$$

- Model checking : Given both p and $\{xi\}$, tell if they are consistent:

$$\mathbf{D}(\{xi\}, p) \longrightarrow 0$$

Stein’s Method 🧐

MCMC: Asymptotically “correct” , but slow.

VI: Need parametric assumption: fast, but “wrong” .

GAN: hard to train, mode collapse

Stein’s Method 🧐

MCMC: Asymptotically “correct” , but slow.

VI: Need parametric assumption: fast, but “wrong” .

GAN: hard to train, mode collapse

Stein:

是一种结合了Variational Inference和Markov Chain Monte Carlo的Approximate Inference方法。

首先，这个方法类似于变分推断，是一个优化方法，其次，这个方法保留MCMC的渐进一致性，是一个无参方法，从而在运算速度上有优势。

Stein's Method 🧐

Stein is focuses on inference(sampling) problem:

Given p, find $\{x_i\}$ to approximation p.

$$p = q \quad \Longleftrightarrow \quad \mathbb{E}_{x\sim q}[\mathcal{T}_p \phi(x)] = 0.$$

$$\mathcal{T}_p \phi(x) \overset{def}{=} \langle \nabla_x \log p(x), \;\; \phi(x) \rangle + \nabla_x \cdot \phi(x).^1$$

Stein's Method 🧐

Given p, find $\{x_i\}$ to approximation p.

$$p = q \quad \Longleftrightarrow \quad \mathbb{E}_{x\sim q}[\mathcal{T}_p \phi(x)] = 0.$$

$$\mathcal{T}_p \phi(x) \overset{def}{=} \langle \nabla_x \log p(x), \ \ \phi(x) \rangle + \nabla_x \cdot \phi(x).^1$$

$\Big\downarrow$? 正确性

$$\text{Stein's Identity} : \mathbb{E}_{x\sim p}[\langle \nabla_x \log p(x), \ \ \phi(x)\rangle + \nabla_x \cdot \phi(x)] = 0$$

$$\int p(x) \nabla_x \phi(x) + \phi(x) \nabla_x p(x) dx = p(x)\phi(x)\Big|_{-\infty}^{+\infty} = 0.$$

Stein's Method 🧐

$$p = q \quad \Longleftrightarrow \quad \mathbb{E}_{x\sim q}[\mathcal{T}_p\phi(x)] = 0.$$

$$p \neq q \quad \Rightarrow \quad \exists \text{ some } \phi, \quad \mathbb{E}_{x\sim q}[\mathcal{T}_p\phi(x)] \neq 0$$

Stein's Method 🧐

$$p = q \iff \mathbb{E}_{x\sim q}[\mathcal{T}_p \phi(x)] = 0.$$

$$p \neq q \Rightarrow \exists \text{ some } \phi, \quad \mathbb{E}_{x\sim q}[\mathcal{T}_p \phi(x)] \neq 0$$

$$\begin{aligned}\mathbb{E}_{x\sim q}[\mathcal{T}_p \phi(x)] &= \mathbb{E}_{x\sim q}[\mathcal{T}_p \phi(x)] - \mathbb{E}_{x\sim q}[\mathcal{T}_q \phi(x)] \\ &= \mathbb{E}_{x\sim q}[\langle \boldsymbol{s}_p(x) - \boldsymbol{s}_q(x), \quad \phi(x)\rangle]\end{aligned}$$

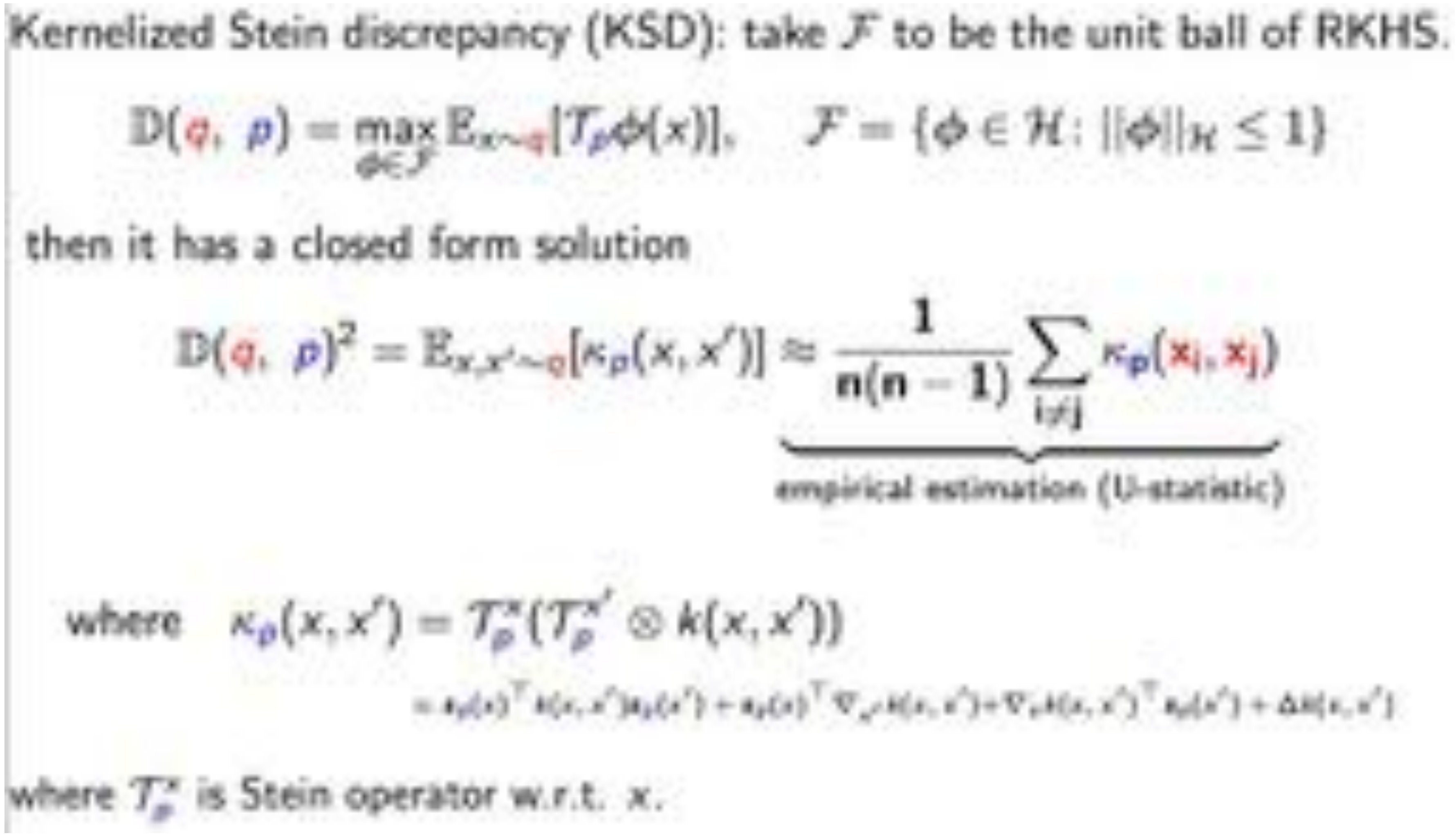
除非$\nabla_x \log p(x) = \nabla_x \log q(x)$

Define Stein discrepancy between p and q:

$$\mathbb{D}(q, p) = \max_{\phi\in\mathcal{F}} \mathbb{E}_{x\sim q}[\mathcal{T}_p \phi(x)]$$

$\mathcal{F}$: a rich enough set of functions.

Kernelized Stein Discrepancy 🧐



Kernelized Stein discrepancy (KSD): take $\mathcal{F}$ to be the unit ball of RKHS.

$$\mathbb{D}(q,\ p) = \max_{\phi \in \mathcal{F}} \mathbb{E}_{x \sim q}[\mathcal{T}_p \phi(x)], \qquad \mathcal{F} = \{\phi \in \mathcal{H} : \|\phi\|_{\mathcal{H}} \le 1\}$$

then it has a closed form solution

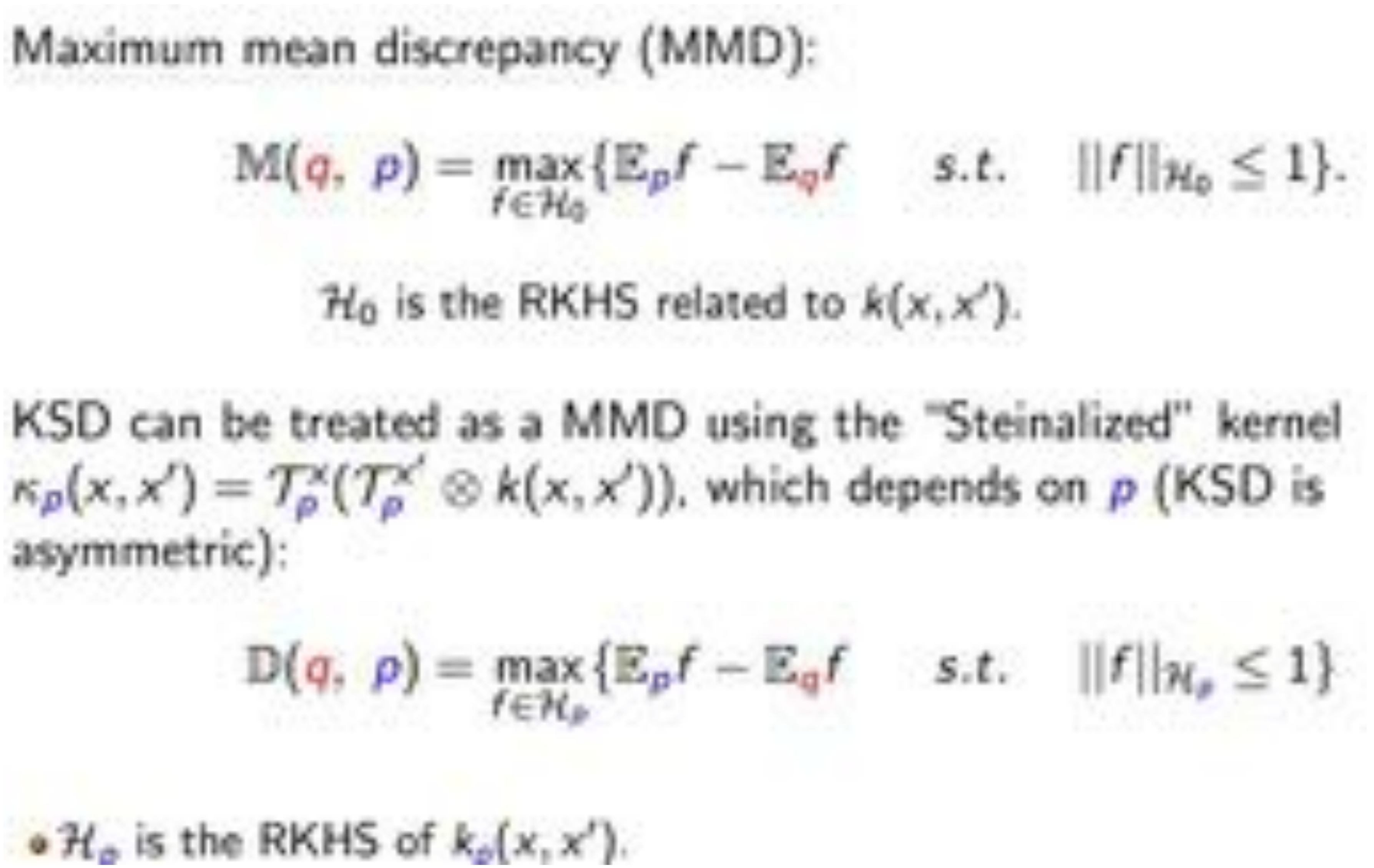
$$\mathbb{D}(q,\ p)^2 = \mathbb{E}_{x,x' \sim q}[\kappa_p(x, x')] \approx \underbrace{\frac{1}{n(n-1)} \sum_{i \neq j} \kappa_p(x_i, x_j)}_{\text{empirical estimation (U-statistic)}}$$

where $\quad \kappa_p(x, x') = \mathcal{T}_p^x(\mathcal{T}_p^{x'} \otimes k(x, x'))$

$$= s_p(x)^\top k(x,x') s_p(x') + s_p(x)^\top \nabla_{x'} k(x,x') + \nabla_x k(x,x')^\top s_p(x') + \Delta k(x,x')$$

where $\mathcal{T}_p^x$ is Stein operator w.r.t. x.

Connection with MMD🧐



Maximum mean discrepancy (MMD):

$$\mathrm{M}(q,\ p) = \max_{f \in \mathcal{H}_0} \{\mathbb{E}_p f - \mathbb{E}_q f \quad s.t. \quad ||f||_{\mathcal{H}_0} \leq 1\}.$$

$\mathcal{H}_0$ is the RKHS related to $k(x, x')$.

KSD can be treated as a MMD using the "Steinalized" kernel $\kappa_p(x, x') = \mathcal{T}_p^x(\mathcal{T}_p^{x'} \otimes k(x, x'))$, which depends on p (KSD is asymmetric):

$$\mathrm{D}(q,\ p) = \max_{f \in \mathcal{H}_p} \{\mathbb{E}_p f - \mathbb{E}_q f \quad s.t. \quad ||f||_{\mathcal{H}_p} \leq 1\}$$

- $\mathcal{H}_p$ is the RKHS of $k_p(x, x')$.

Target🧐

find a set of good point $\{x_i\}$ to approximate p

Stein Variational Gradient Descent

Directly minimize $\mathrm{KL}(\{x_i\} \parallel p)$.

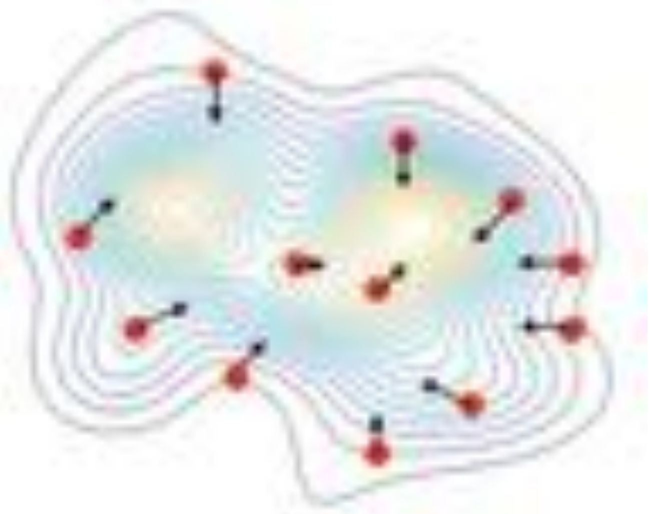
Idea: Iteratively move $\{x_i\}_{i=1}^{n}$ towards the target p by updates of form

$$x_i' \leftarrow x_i + \epsilon\phi(x_i),$$

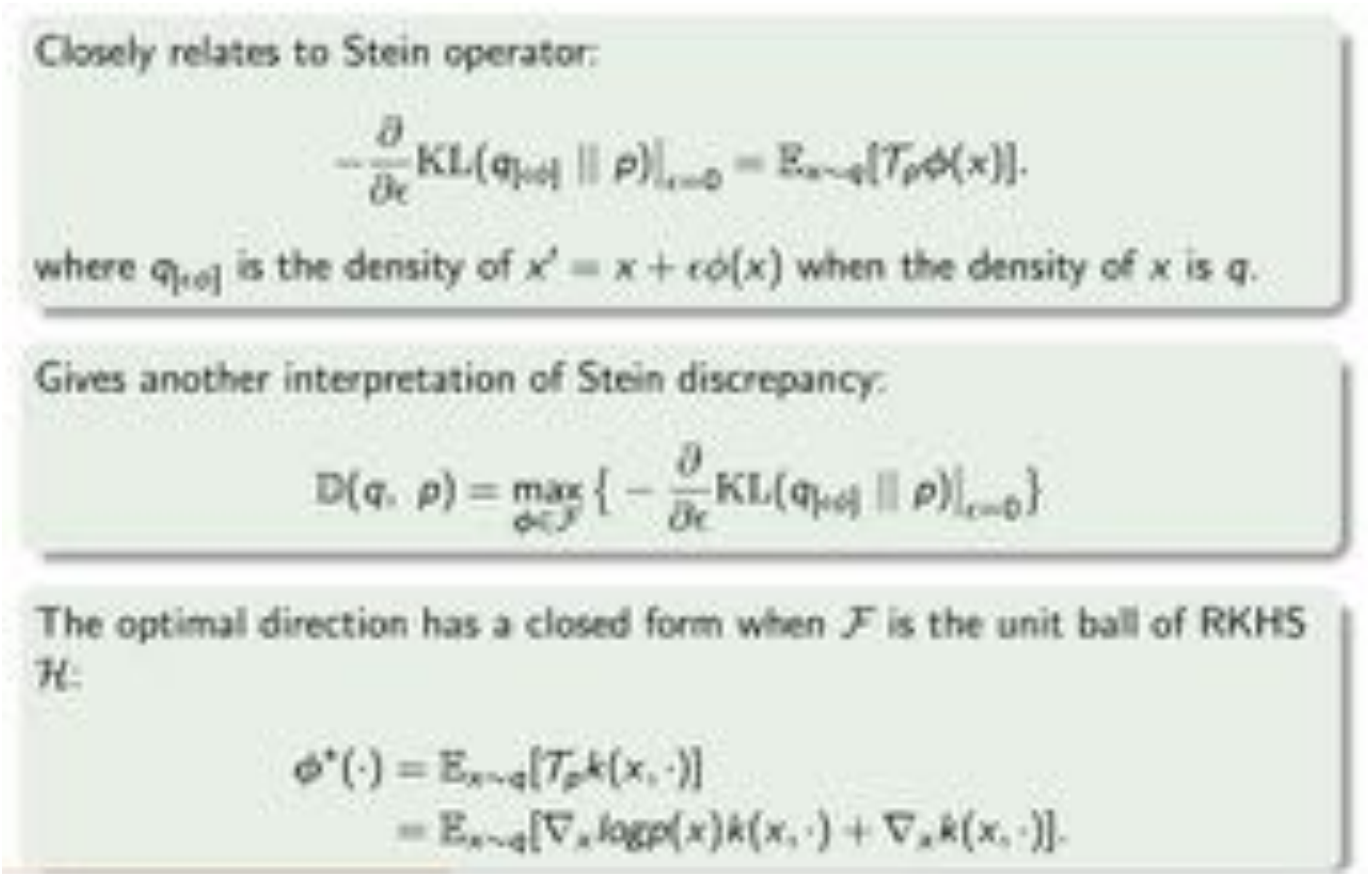
where ϕ is a perturbation direction chosen to maximumly decrease the KL divergence with p, that is,

$$\phi = \arg\max_{\phi\in\mathcal{F}} \left\{ -\frac{\partial}{\partial\epsilon}\mathrm{KL}(q_{[\epsilon\phi]} \parallel p)\big|_{\epsilon=0} \right\},$$

where $q_{[\epsilon\phi]}$ is the density of $x' = x + \epsilon\phi(x)$ when the density of x is q.



SVGD🧐



Closely relates to Stein operator:

$$-\frac{\partial}{\partial\epsilon}\mathrm{KL}(q_{[\epsilon\phi]}\,\|\,p)\big|_{\epsilon=0} = \mathbb{E}_{x\sim q}[\mathcal{T}_p\phi(x)].$$

where $q_{[\epsilon\phi]}$ is the density of $x' = x + \epsilon\phi(x)$ when the density of x is q.

Gives another interpretation of Stein discrepancy:

$$\mathbb{D}(q,\ p) = \max_{\phi\in\mathcal{F}}\left\{-\frac{\partial}{\partial\epsilon}\mathrm{KL}(q_{[\epsilon\phi]}\,\|\,p)\big|_{\epsilon=0}\right\}$$

The optimal direction has a closed form when $\mathcal{F}$ is the unit ball of RKHS $\mathcal{H}$:

$$\begin{aligned}\phi^*(\cdot) &= \mathbb{E}_{x\sim q}[\mathcal{T}_p k(x,\cdot)]\\ &= \mathbb{E}_{x\sim q}[\nabla_x \log p(x) k(x,\cdot) + \nabla_x k(x,\cdot)].\end{aligned}$$

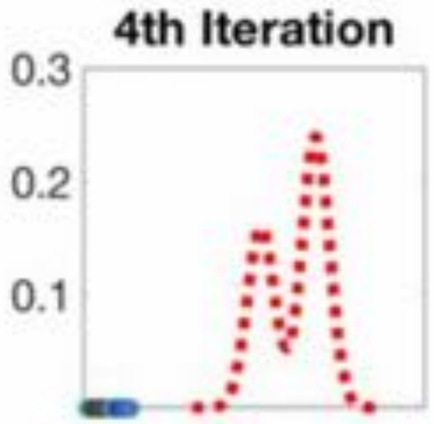
Target🧐

Approximating $\mathbb{E}_{x\sim q}[\cdot]$ with empirical averaging over the current points gives:

$$x_i \leftarrow x_i + \epsilon \hat{\mathbb{E}}_{x\sim\{x_i\}_{i=1}^n}[\underbrace{\nabla_x \log p(x)}_{\text{gradient}}\, k(x, x_i) + \underbrace{\nabla_x k(x, x_i)}_{\text{repulsive force}}], \quad \forall i = 1, \dots, n.$$

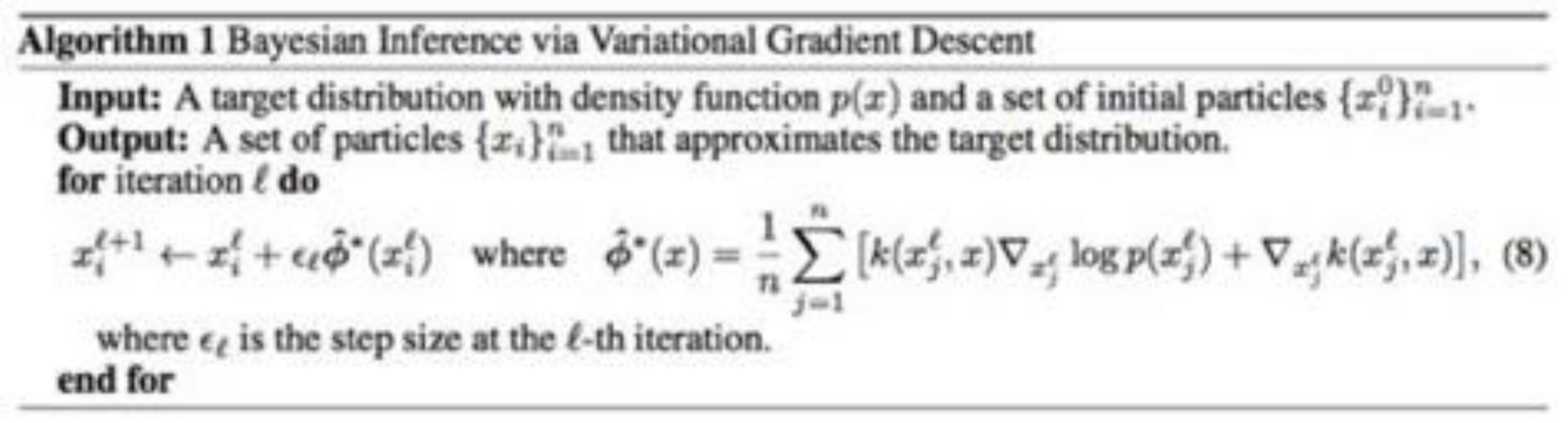
Two terms:

- $\nabla_x \log p(x)$: moves the particles $\{x_i\}$ towards high probability regions of $p(x)$.
- $\nabla_x k(x, x')$: enforce diversity in $\{x_i\}$ (otherwise all x_i collapse to modes of $p(x)$).



Deterministically transport probability mass from initialize q_0 to target p.

Algorithm&&Experiments🧐



Algorithm 1 Bayesian Inference via Variational Gradient Descent

```
Input: A target distribution with density function $p(x)$ and a set of initial particles $\{x_i^0\}_{i=1}^n$.
Output: A set of particles $\{x_i\}_{i=1}^n$ that approximates the target distribution.
for iteration $\ell$ do
    $x_i^{\ell+1} \leftarrow x_i^\ell + \epsilon_\ell \hat{\phi}^*(x_i^\ell)$   where   $\hat{\phi}^*(x) = \frac{1}{n}\sum_{j=1}^{n}\left[k(x_j^\ell, x)\nabla_{x_j^\ell}\log p(x_j^\ell) + \nabla_{x_j^\ell}k(x_j^\ell, x)\right]$,   (8)
      where $\epsilon_\ell$ is the step size at the $\ell$-th iteration.
end for
```

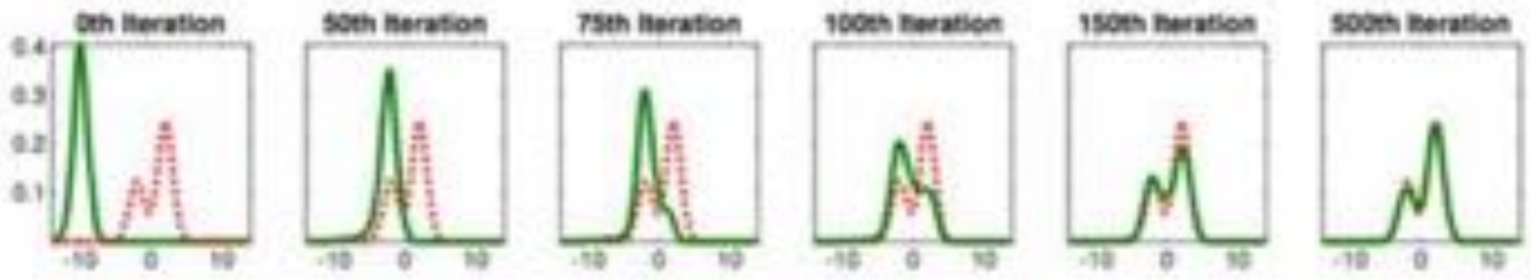
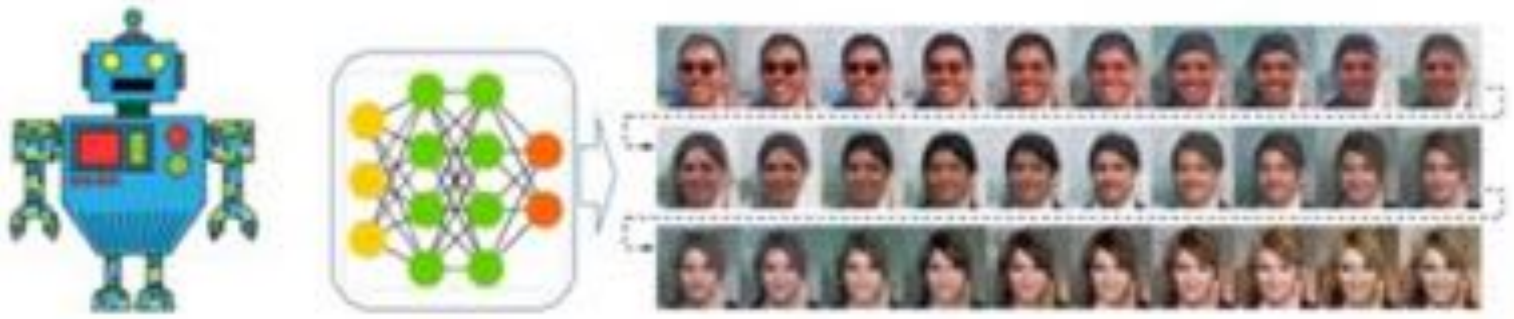


Figure 1: Toy example with 1D Gaussian mixture. The red dashed lines are the target density function and the solid green lines are the densities of the particles at different iterations of our algorithm (estimated using kernel density estimator) . Note that the initial distribution is set to have almost zero overlap with the target distribution, and our method demonstrates the ability of escaping the local mode on the left to recover the mode on the left that is further away. We use $n = 100$ particles.

Applications🧐

- Training neural networks to generate natural images.



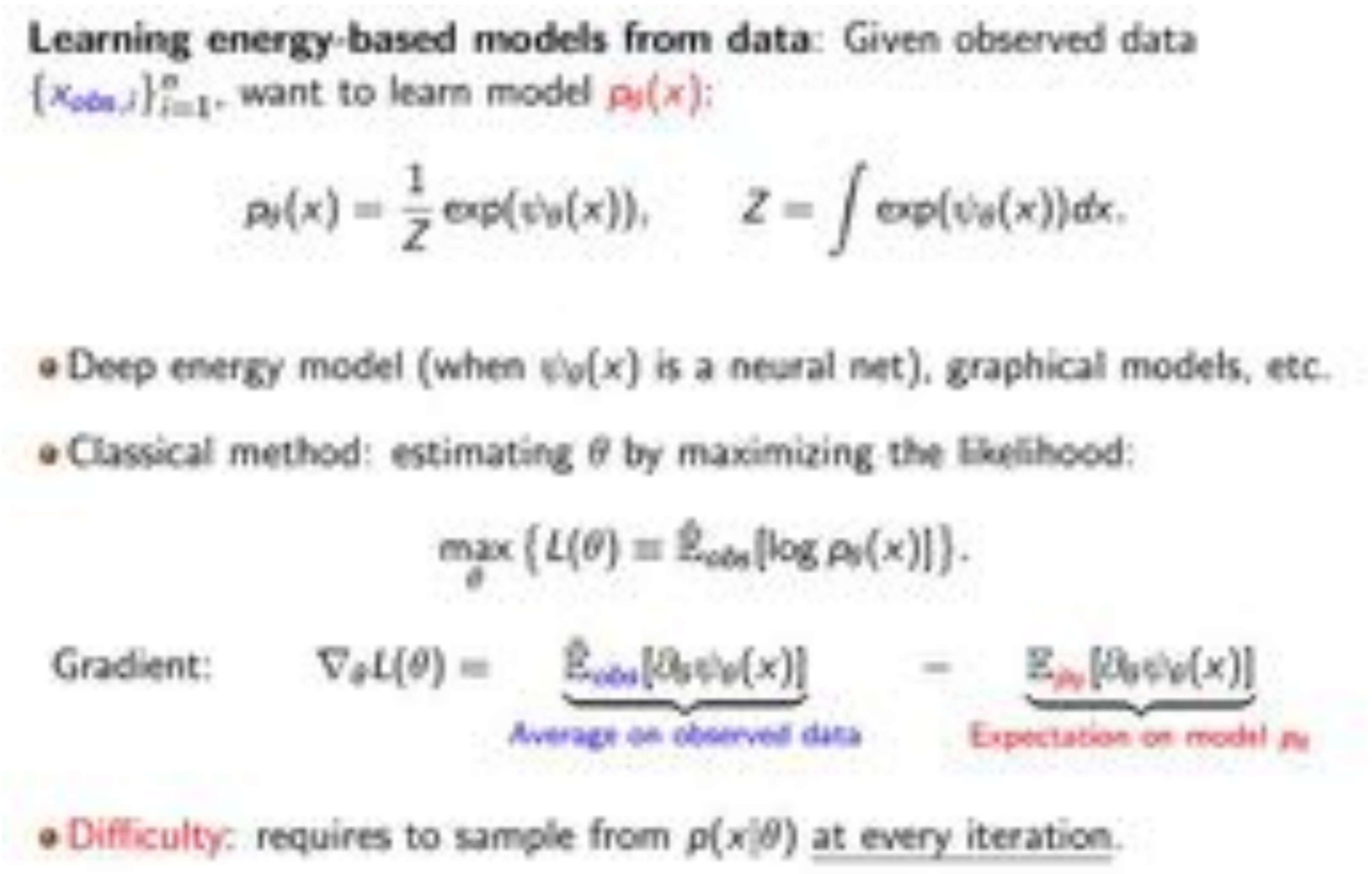
- Policy optimization in reinforcement learning.

  Particles collaborate to explore large space.

  Can be used to solve challenging non-convex optimization problems.

Learning to Sample -> Learning to Optimise

Stein For Generative Adversarial learning🧐



Learning energy-based models from data: Given observed data $\{x_{obs,i}\}_{i=1}^{n}$, want to learn model $p_\theta(x)$:

$$p_\theta(x) = \frac{1}{Z}\exp(\psi_\theta(x)), \qquad Z = \int \exp(\psi_\theta(x))dx.$$

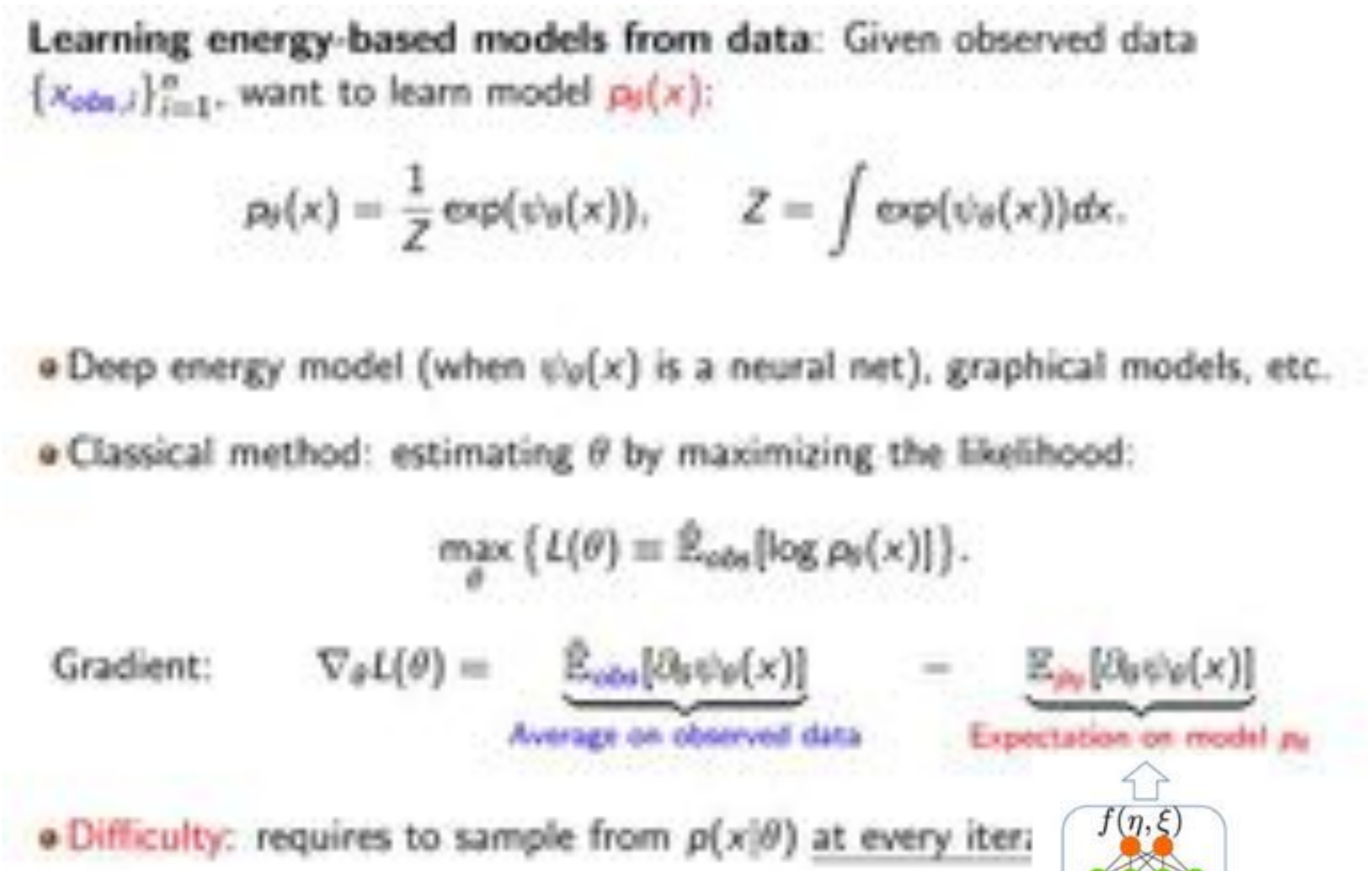
- Deep energy model (when $\psi_\theta(x)$ is a neural net), graphical models, etc.
- Classical method: estimating θ by maximizing the likelihood:

$$\max_\theta \left\{ L(\theta) \equiv \hat{\mathbb{E}}_{obs}[\log p_\theta(x)] \right\}.$$

Gradient: $$\nabla_\theta L(\theta) = \underbrace{\hat{\mathbb{E}}_{obs}[\partial_\theta \psi_\theta(x)]}_{\text{Average on observed data}} - \underbrace{\mathbb{E}_{p_\theta}[\partial_\theta \psi_\theta(x)]}_{\text{Expectation on model } p_\theta}$$

- Difficulty: requires to sample from $p(x|\theta)$ <u>at every iteration</u>.

Stein For Generative Adversarial learning🧐



Learning energy-based models from data: Given observed data $\{x_{obs,i}\}_{i=1}^{n}$, want to learn model $p_\theta(x)$:

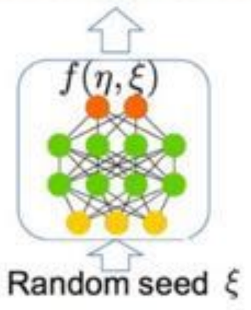
$$p_\theta(x) = \frac{1}{Z}\exp(\psi_\theta(x)), \qquad Z = \int \exp(\psi_\theta(x))dx.$$

- Deep energy model (when $\psi_\theta(x)$ is a neural net), graphical models, etc.
- Classical method: estimating θ by maximizing the likelihood:

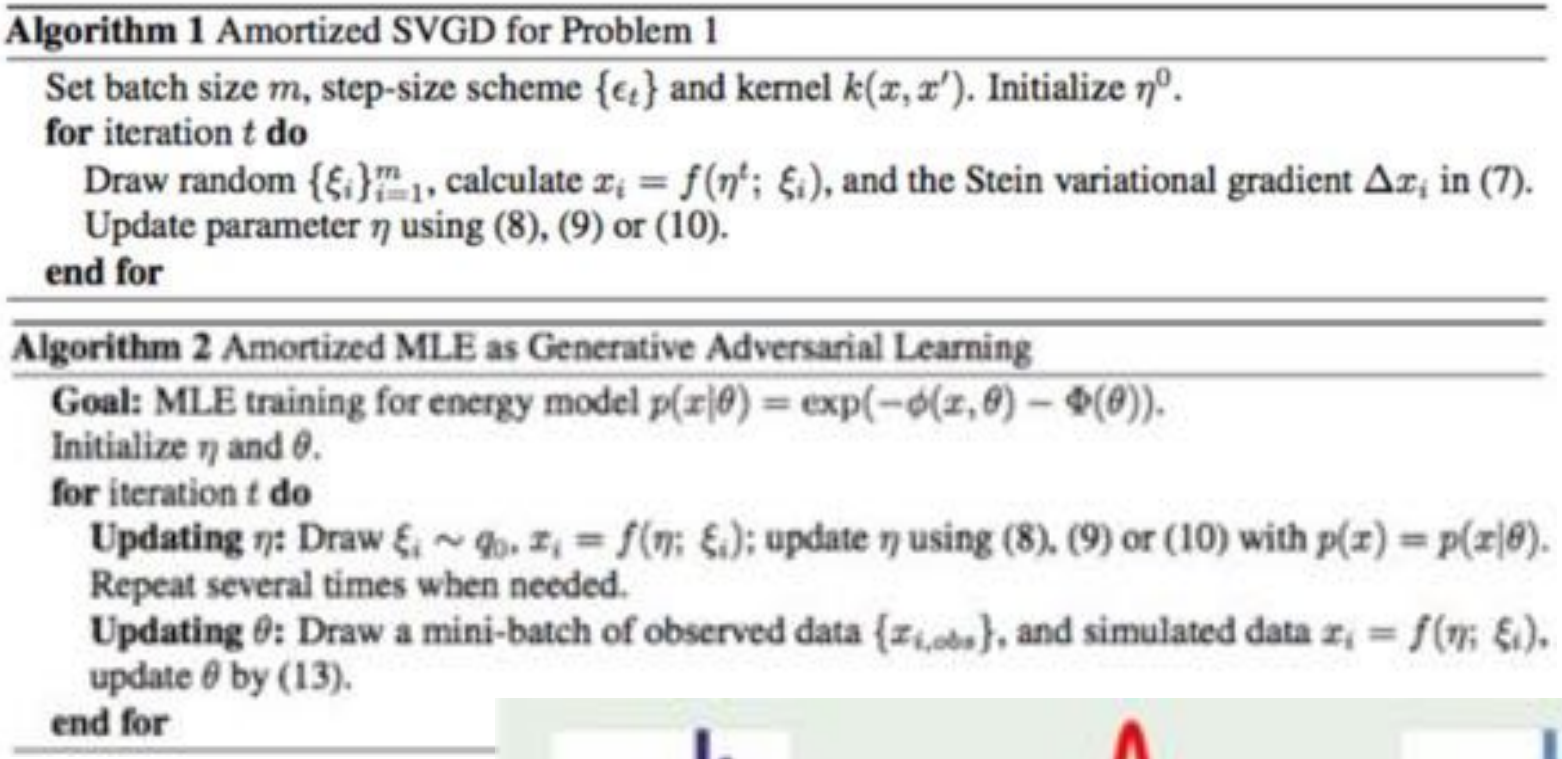
$$\max_\theta \left\{ L(\theta) \equiv \hat{\mathbb{E}}_{obs}[\log p_\theta(x)] \right\}.$$

Gradient: $$\nabla_\theta L(\theta) = \underbrace{\hat{\mathbb{E}}_{obs}[\partial_\theta \psi_\theta(x)]}_{\text{Average on observed data}} - \underbrace{\mathbb{E}_{p_\theta}[\partial_\theta \psi_\theta(x)]}_{\text{Expectation on model } p_\theta}$$

- Difficulty: requires to sample from $p(x|\theta)$ at every itera



Stein For Generative Adversarial learning🧐

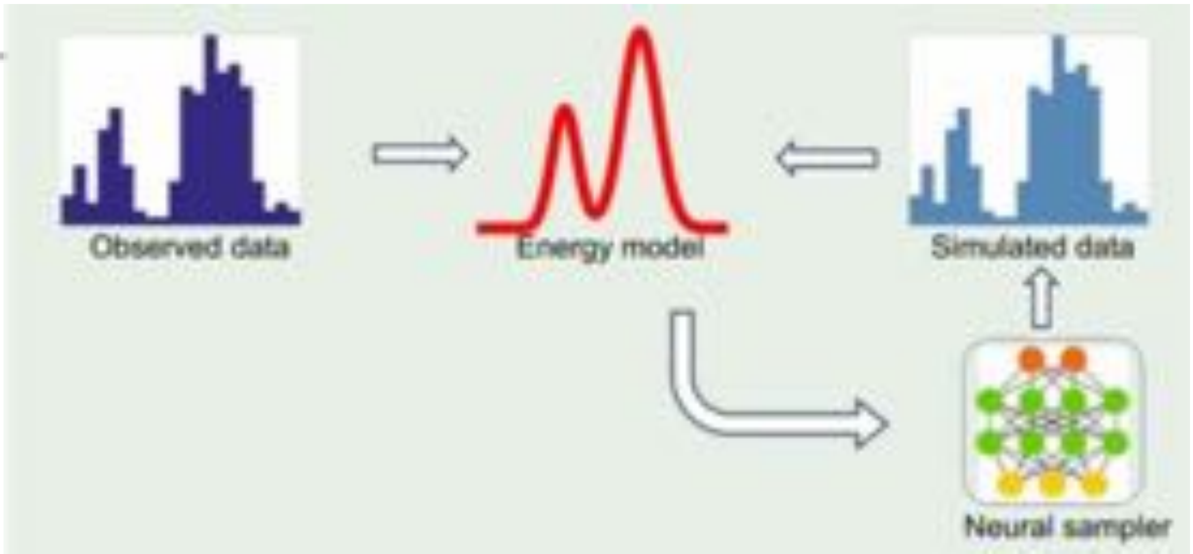


Algorithm 1 Amortized SVGD for Problem 1

```
Set batch size $m$, step-size scheme $\{\epsilon_t\}$ and kernel $k(x, x')$. Initialize $\eta^0$.
for iteration $t$ do
    Draw random $\{\xi_i\}_{i=1}^m$, calculate $x_i = f(\eta^t;\ \xi_i)$, and the Stein variational gradient $\Delta x_i$ in (7).
    Update parameter $\eta$ using (8), (9) or (10).
end for
```

Algorithm 2 Amortized MLE as Generative Adversarial Learning

```
Goal: MLE training for energy model $p(x|\theta) = \exp(-\phi(x, \theta) - \Phi(\theta))$.
Initialize $\eta$ and $\theta$.
for iteration $t$ do
    Updating $\eta$: Draw $\xi_i \sim q_0$, $x_i = f(\eta;\ \xi_i)$; update $\eta$ using (8), (9) or (10) with $p(x) = p(x|\theta)$.
    Repeat several times when needed.
    Updating $\theta$: Draw a mini-batch of observed data $\{x_{i,obs}\}$, and simulated data $x_i = f(\eta;\ \xi_i)$,
    update $\theta$ by (13).
end for
```



Experiments 🧐

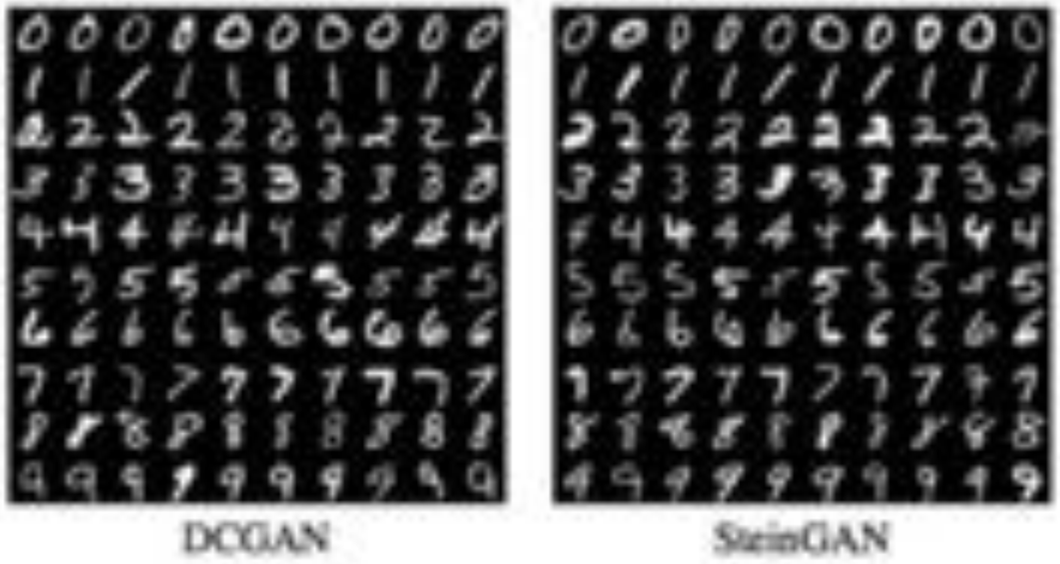


Figure 1: MNIST images generated by DCGAN and our SteinGAN. We use the joint model in (15) to allow us to generate images for each digit. We set $m = 0.2$.

Experiments

Inception Score				
	Real Training Set	500 Duplicate	DCGAN	SteinGAN
Model Trained on ImageNet	11.237	11.100	6.581	6.351
Model Trained on CIFAR-10	9.848	9.807	7.368	7.428

Testing Accuracy			
Real Training Set	500 Duplicate	DCGAN	SteinGAN
92.58 %	44.96 %	44.78 %	63.81 %

Stein Variational Policy Gradient 🧐

- Stein variational policy gradient: find a group of $\{\theta_i\}$ by

$$\theta_i \leftarrow \theta_i + \frac{\epsilon}{n}\sum_{j=1}^{n}[\underbrace{\nabla_{\theta_j} J(\theta_j)k(\theta_j,\theta_i)}_{\text{gradient sharing}} + \alpha \underbrace{\nabla_{\theta_j} k(\theta_j,\theta_i)}_{\text{repulsive force}}]$$

- Can be viewed as sampling $\{\theta_i\}$ from a Boltzmann distribution:

$$p(\theta) \propto \exp(\frac{1}{\alpha}J(\theta)) = \arg\max_q \{\mathbb{E}_q[J(\theta)] \underbrace{+ \alpha H(q)}_{\substack{\text{entropy regularization}\\ \text{encourage exploration}}}\},$$

α : temperature parameter. $H(q)$: entropy.

Summary 🧐

- nonparametric variational inference.
- deterministic sampling.
- gradient-based quadrature method.