

Stein's Method for Modern Machine Learning

From Gradient Estimation to Generative Modeling

Jiaxin Shi

Google DeepMind
2024/3/14 @ OT-Berlin

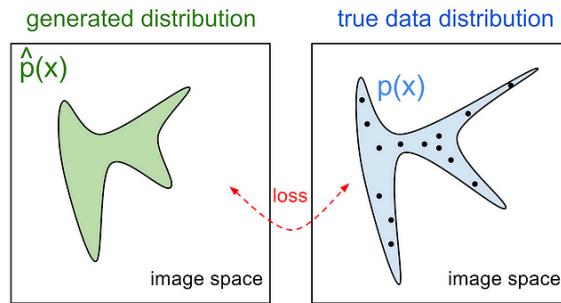
jiaxins.io

Outline

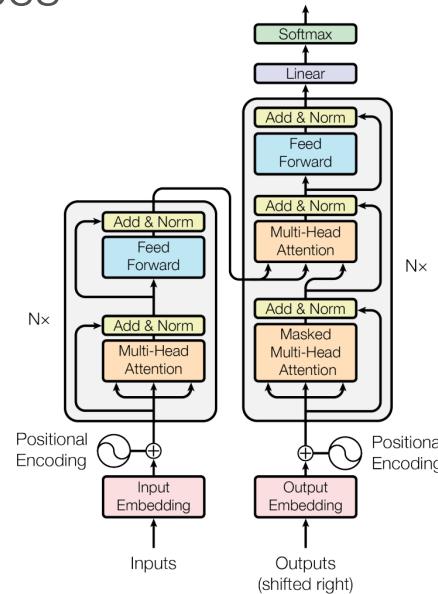
- Stein's method: Foundations
- Stein's method and machine learning
 - Sampling
 - Gradient estimation
 - Score-based modeling

Divergences between Probability Distributions

- How well does my model fit the data?
- Parameter estimation by minimizing divergences
- Sampling as optimization



GANs, Diffusion models



Transformers

Integral Probability Metrics (IPM)

$$d_{\mathcal{H}}(q, p) = \sup_{h \in \mathcal{H}} |\mathbb{E}_q[h(X)] - \mathbb{E}_p[h(Y)]|$$

- When \mathcal{H} is sufficient large, convergence in $d_{\mathcal{H}}(q_n, p)$ implies q_n weakly converges to p
- Examples: Total variation distance, Wasserstein distance
- **Problem:** Often p is our model and integration under p is intractable
- **Idea:** Only consider functions with $\mathbb{E}_p[h(Y)] = 0$

Stein's Method

- Identify an operator \mathcal{T} that generates mean-zero functions under target distribution p .

$$\mathbb{E}_p[(\mathcal{T}g)(X)] = 0 \text{ for all } g \in \mathcal{G}$$



- Define the Stein discrepancy:

$$\mathcal{S}(q, \mathcal{T}, \mathcal{G}) \triangleq \sup_{g \in \mathcal{G}} \mathbb{E}_q[(\mathcal{T}g)(X)] - \mathbb{E}_p[(\mathcal{T}g)(X)]$$

- Show that the Stein discrepancy is lower bounded by an IPM. For example, if for any $h \in \mathcal{H}$, a solution $g \in \mathcal{G}$ exists for the equation $h(x) - \mathbb{E}_p[h(Y)] = (\mathcal{T}g)(x)$, then $d_{\mathcal{H}}(q, p) \leq \mathcal{S}(q, p, \mathcal{G})$.

[Stein, 1972]

Identifying a Stein Operator

Stein's Lemma

If p is a standard normal distribution, then

$$\mathbb{E}_p[g'(X) - Xg(X)] = 0 \text{ for all } g \in C_b^1$$

The corresponding Stein operator: $\mathcal{T}(x) = g'(x) - xg(x)$

Identifying a Stein Operator

Barbour's generalization via stochastic processes

- The (infinitesimal) generator A of a stochastic process $(X_t)_{t \geq 0}$ is defined as

$$(Af)(x) = \lim_{t \rightarrow 0} \frac{\mathbb{E}[f(X_t) | X_0 = x] - f(x)}{t}.$$

- The generator of a stochastic process with stationary distribution p satisfies $\mathbb{E}_p[(Af)(X)] = 0$.

[Barbour, 1988 & 1990]

Langevin Stein Operator

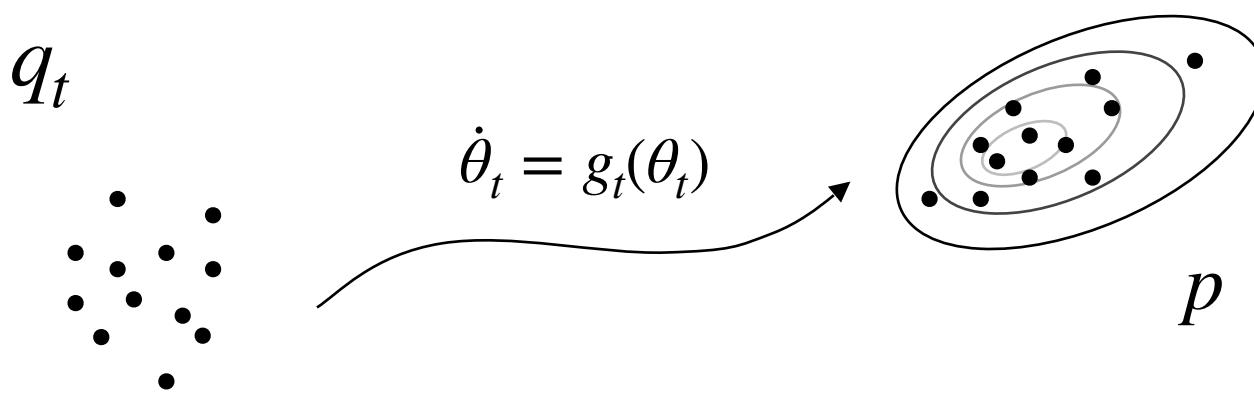
- Langevin diffusion on \mathbb{R}^d : $dX_t = \nabla \log p(X_t) dt + \sqrt{2} dW_t$
- Generator:

$$(Af)(x) = \nabla \log p(x)^\top \nabla f(x) + \nabla \cdot \nabla f(x)$$

- Convenient form with a vector-valued function $g : \mathbb{R}^d \rightarrow \mathbb{R}^d$:
$$(\mathcal{T}_p g)(x) = \nabla \log p(x)^\top g(x) + \nabla \cdot g(x)$$
- Depends on p only through $\nabla \log p$, computable even for unnormalized p

[Gorham & Mackey, 2015]

Stein Operators and Sampling



Find the direction that most quickly decreases the KL divergence to p

$$\frac{d}{dt} \text{KL}(q_t \| p) = -\mathbb{E}_{q_t}[(\mathcal{T}_p g_t)(X)]$$

[Liu & Wang, 2016]

Wasserstein Gradient Flow and SVGD

$$\inf_{g_t \in \mathcal{G}} \frac{d}{dt} \text{KL}(q_t \| p) = - \sup_{g_t \in \mathcal{G}} \mathbb{E}_{q_t}[(\mathcal{T}_p g_t)(X)]$$

- $\mathcal{G} = \mathcal{L}^2(q_t)$: Wasserstein Gradient Flow

$$g_t^* \propto \nabla \log p - \nabla \log q_t,$$

Same density evolution as Langevin diffusion

- $\mathcal{G} = \text{RKHS of kernel } K$: Stein Variational Gradient Descent [Liu & Wang, 2016]

$$g_t^* \propto \mathbb{E}_{q_t}[K(\cdot, X) \nabla \log p(X) + \nabla_X \cdot K(\cdot, X)]$$

Convergence Analysis of SVGD

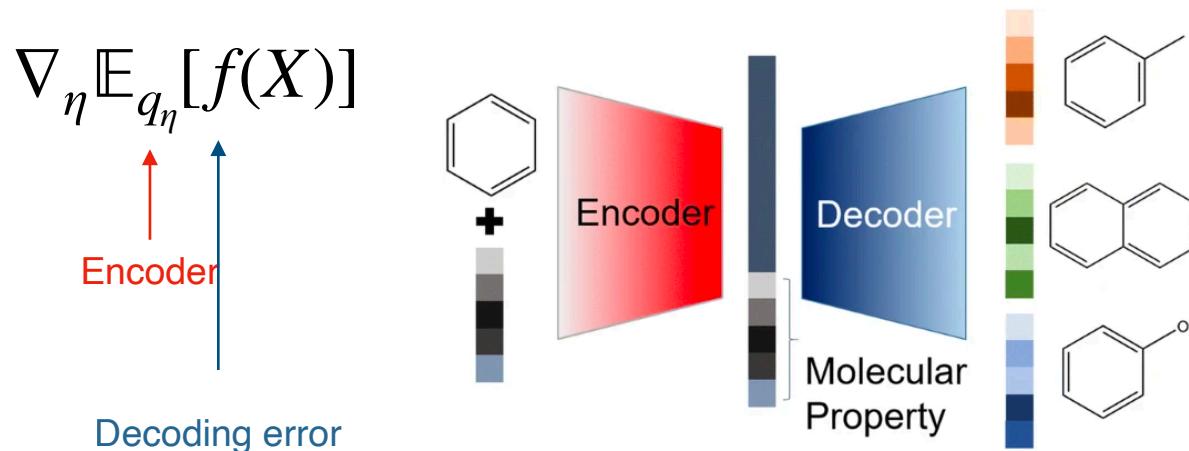
- Korba, A., Salim, A., Arbel, M., Luise, G., & Gretton, A. A non-asymptotic analysis for Stein variational gradient descent. *NeurIPS* (2020).
- Chewi, S., Le Gouic, T., Lu, C., Maunu, T., & Rigollet, P. SVGD as a kernelized Wasserstein gradient flow of the chi-squared divergence. *NeurIPS* (2020).
- Shi, J., & Mackey, L. A finite-particle convergence rate for Stein variational gradient descent. *NeurIPS* (2023).

Convergence rate for discrete-time, finite-particle SVGD

Stein's Method and Gradient Estimation

The Gradient Estimation Problem

A common problem in training generative models and reinforcement learning



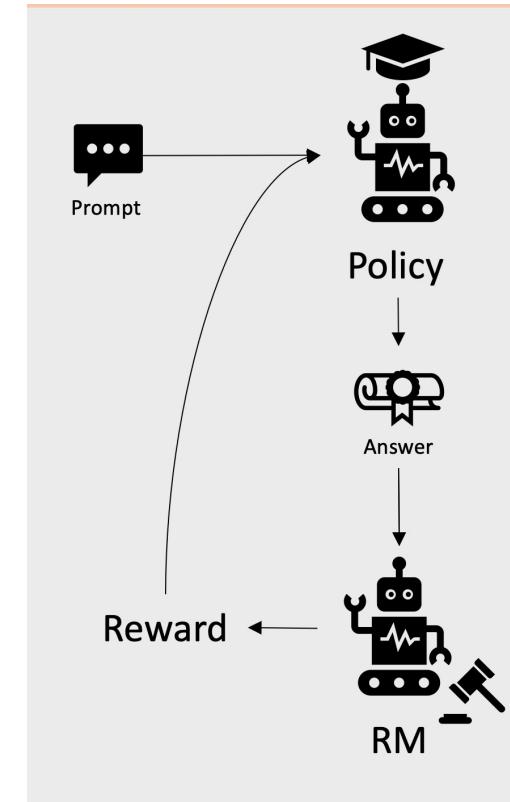
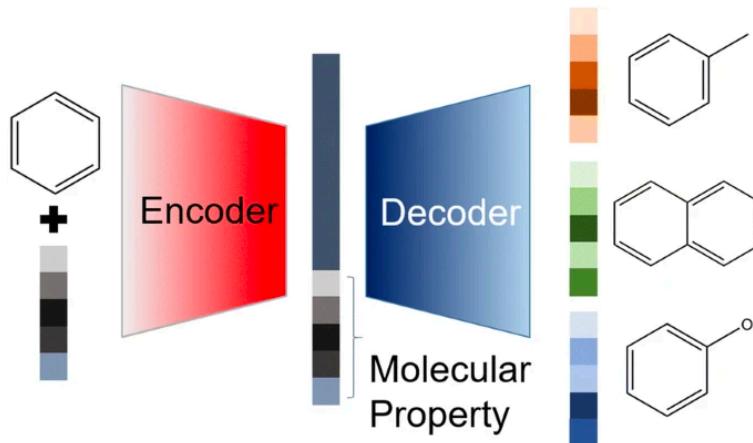
The Gradient Estimation Problem

A common problem in training generative models and reinforcement learning

$$\nabla_{\eta} \mathbb{E}_{q_{\eta}}[f(X)]$$

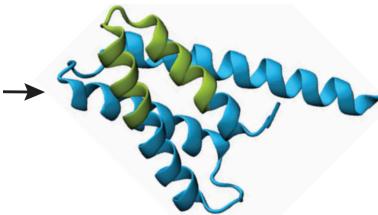
Policy

Reward Model

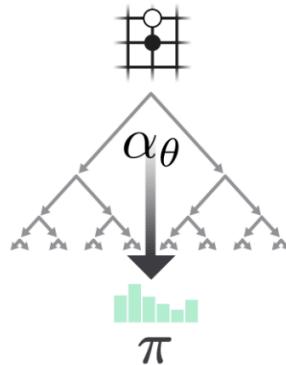


Discrete Gradient Estimation

- Discrete data, states, and actions



MDDTLFSILNSELLSLINDMPITNDQK
KLMSNNFVKMANDLKGEFGDENYY
YVNQTTKYVYIYEARQLLGPTLSD
KIYQKILIRINEKLSRNFNIEIQKNKI



- Computing exact gradients is often intractable

$$\nabla_\eta \mathbb{E}_{q_\eta}[f(X)] = \nabla_\eta \sum_{x \in \{0,1\}^d} q_\eta(x) f(x)$$

d-dimensional binary vector Intractable sum over 2^d configurations Complex, nonlinear function

Model Input
Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output
A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✓

Gradient Estimation and Variance Reduction

$$\hat{g}_1 = \frac{1}{K} \sum_{k=1}^K f(x_k) \nabla_\eta \log q_\eta(x_k) \quad (\text{REINFORCE}) \quad \text{High variance!}$$

$$\hat{g}_2 = \frac{1}{K} \sum_{k=1}^K [f(x_k) \nabla_\eta \log q_\eta(x_k) + cv(x_k)] - \mathbb{E}_{q_\eta}[cv(X)]$$

Control Variates

- Strong correlation is required for effective variance reduction
- Fundamental tradeoff: cv needs to be very *flexible* but still have *analytic expectation* under q_η .

$$\hat{g}_2 = \frac{1}{K} \sum_{k=1}^K [f(x_k) \nabla_\eta \log q_\eta(x_k) + (Ah)(x_k)] - \mathbb{E}_{q_\eta}[(Ah)(X)] \stackrel{=} 0$$

A: Stein Operator

Discrete Stein Operators

How: Apply Barbour's idea to discrete-state Markov chains.

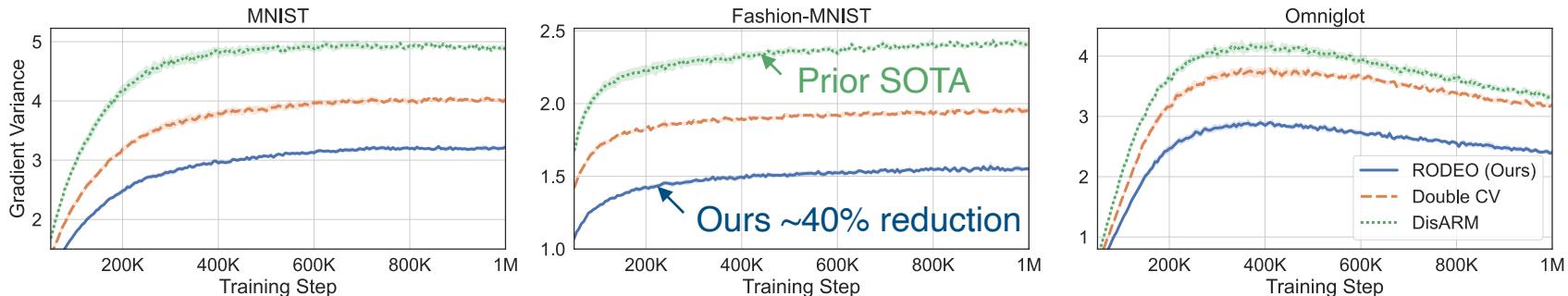
$$\mathbb{E}_q[((K - I)h)(X)] = 0 \xrightarrow{\text{cont. time}} \mathbb{E}_q[(Ah)(X)] = 0$$

K : transfer operator A : generator

Stein Operator	$(Ah)(x)$
Gibbs (4)	$\frac{1}{d} \sum_{i=1}^d \sum_{y_{-i}=x_{-i}} q(y_i x_{-i})h(y) - h(x)$
MPF (6)	$\sum_{y \in \mathcal{N}_x, y \neq x} \sqrt{q(y)/q(x)}(h(y) - h(x))$
Barker (6)	$\sum_{y \in \mathcal{N}_x, y \neq x} \frac{q(y)}{q(x)+q(y)}(h(y) - h(x))$
Difference (8)	$\frac{1}{d} \sum_{i=1}^d h(\mathbf{dec}_i(x)) - \frac{q(\mathbf{inc}_i(x))}{q(x)} h(x)$

Experiments: Training Binary Latent VAEs

		Bernoulli Likelihoods			Gaussian Likelihoods		
		MNIST	Fashion-MNIST	Omniglot	MNIST	Fashion-MNIST	Omniglot
$K = 2$	DisARM	-102.75 ± 0.08	-237.68 ± 0.13	-116.50 ± 0.04	668.03 ± 0.61	182.65 ± 0.47	446.61 ± 1.16
	Double CV	-102.14 ± 0.06	-237.55 ± 0.16	-116.39 ± 0.10	676.87 ± 1.18	186.35 ± 0.64	447.65 ± 0.87
	RODEO (Ours)	-101.89 ± 0.17	-237.44 ± 0.09	-115.93 ± 0.06	681.95 ± 0.37	191.81 ± 0.67	454.74 ± 1.11
$K = 3$	ARMS	-100.84 ± 0.14	-237.05 ± 0.12	-115.21 ± 0.07	683.55 ± 1.01	193.07 ± 0.34	457.98 ± 1.03
	Double CV	-100.94 ± 0.09	-237.40 ± 0.11	-115.06 ± 0.12	686.48 ± 0.68	193.93 ± 0.20	457.44 ± 0.79
	RODEO (Ours)	-100.46 ± 0.13	-236.88 ± 0.12	-115.01 ± 0.05	692.37 ± 0.39	196.56 ± 0.42	461.87 ± 0.90
RELAX (3 evals)		-101.99 ± 0.04	-237.74 ± 0.12	-115.70 ± 0.08	688.58 ± 0.52	196.38 ± 0.66	462.23 ± 0.63



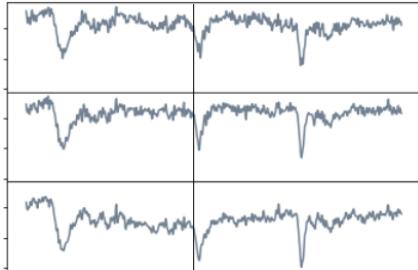
Stein's Method and Score-Based Modeling

Stein's Method and Independent Component Analysis

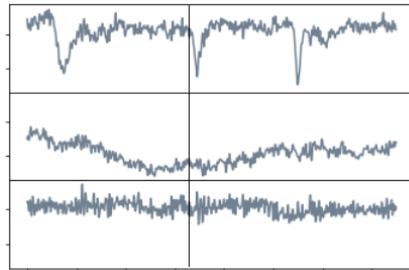
Blind source separation

$$\mathbf{x} = \mathbf{A} \mathbf{s}$$

observation



source



Unmixing:

$$y = A^{-1}x \approx Wx$$

Recovered source signal



FastICA: Maximize Kurtosis of recovered source y :

$$\max_W \text{Kurt}(y) := \mathbb{E}[y^4] - 3(\mathbb{E}[y^2])^2$$

Stein Discrepancy as a Learning Rule

Model fitting:

$$\min_{\theta} \left| \mathbb{E}_q [h(x)^\top \nabla_x \log p_\theta(x) + \nabla \cdot h(x)] \right|$$

The diagram illustrates the components of the Stein discrepancy formula. A blue arrow points from the term \mathbb{E}_q to the label "Model distribution". A red arrow points from the term $h(x)^\top \nabla_x \log p_\theta(x)$ to the label "Data distribution". A question mark "?" is positioned above the term $\nabla \cdot h(x)$.

Data distribution

Model distribution

Score Matching

[Hyvärinen, 2005]

Model fitting:

$$\min_{\theta} \sup_{h \in L^2(q)} |\mathbb{E}_q[h(x)^\top \nabla_x \log p_\theta(x) + \nabla \cdot h(x)]|$$

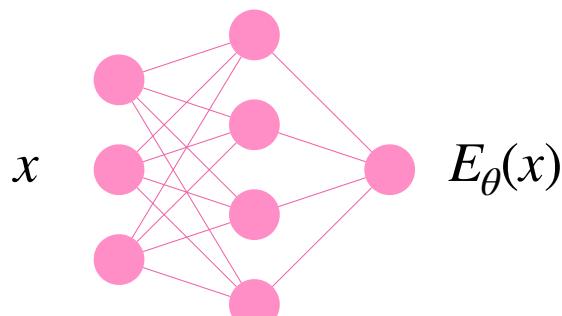
↓ ↓
Data distribution Model distribution

$$\rightarrow \min_{\theta} \mathbb{E}_{q_{\text{data}}} [\|\nabla \log p_\theta(x) - \nabla \log q_{\text{data}}(x)\|^2]$$



Training Energy-Based Models

$$p_\theta(x) = \frac{e^{-E_\theta(x)}}{Z_\theta}$$



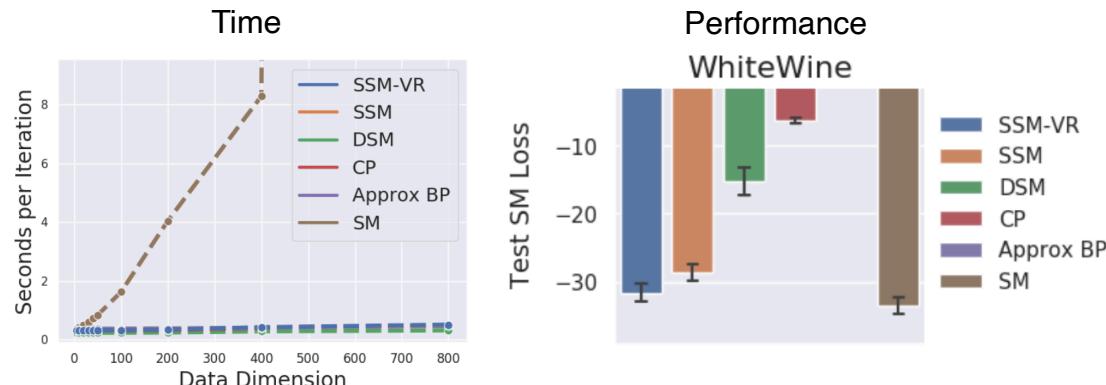
Sliced Score Matching

[Song*, Garg*, Shi & Ermon, UAI'19]

Key insight: The score does not depend on normalizing constant Z_θ

$$\nabla_x \log p_\theta(x) = -\nabla E_\theta(x) + \cancel{\nabla_x \log Z_\theta}$$

- Score Matching is more suitable for training such models than maximum likelihood!



Score-Based Modeling

Idea: Model the score $s := \nabla \log p$ instead of the density

Advantages:

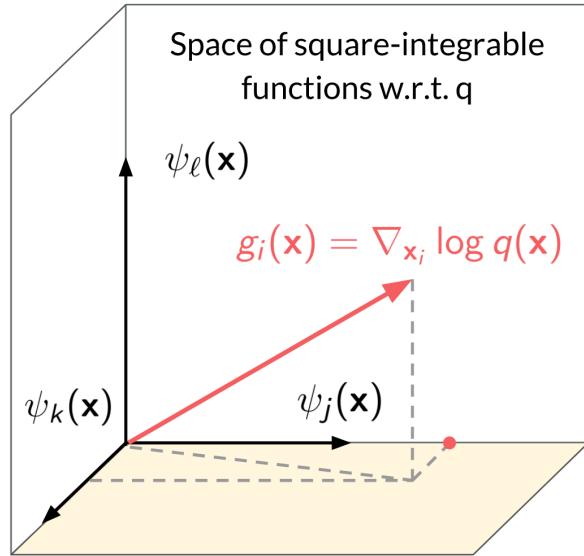
1. less computation than energy-based modeling
2. enable more flexible models

Nonparametric Score Model

$$\min_{s \in \mathcal{H}} \mathbb{E}_{q_{\text{data}}} \|s(x) - \nabla \log q_{\text{data}}(x)\|^2 + \frac{\lambda}{2} \|s\|_{\mathcal{H}}^2$$

The spectral estimator (Shi et al., 18)
is a special case.

A Spectral Method for Score Estimation



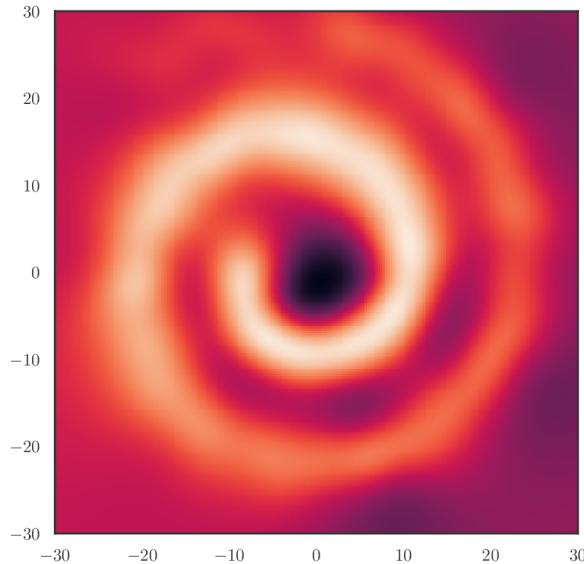
$$\nabla_{\mathbf{x}} \log q(\mathbf{x}) = - \sum_{j \geq 1} \mathbb{E}_q \left[\nabla \psi_j(\mathbf{x}) \right] \psi_j(\mathbf{x})$$

density gradients (score) eigenfunction

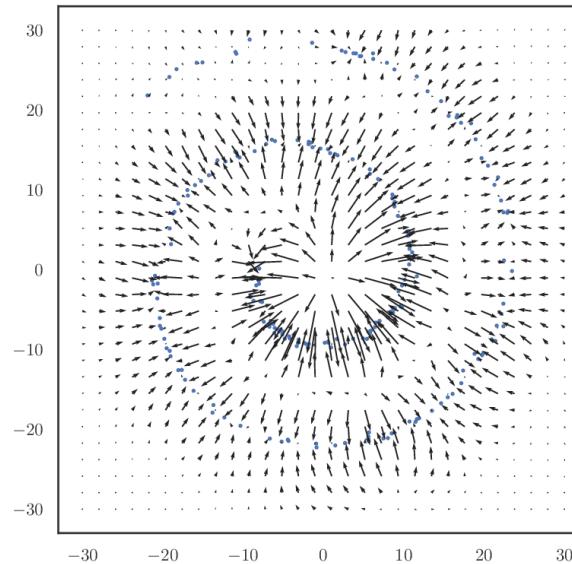
$$\langle \nabla \log q, \psi_j \rangle_{L^2(q)} = - \mathbb{E}_q [\nabla \psi_j(x)]$$

$$\mathbb{E}_{\mathbf{x}' \sim q} [k(\mathbf{x}, \mathbf{x}') \psi_j(\mathbf{x}')] = \lambda_j \psi_j(\mathbf{x})$$

A Spectral Method for Score Estimation



$q(\mathbf{x})$ (unknown)



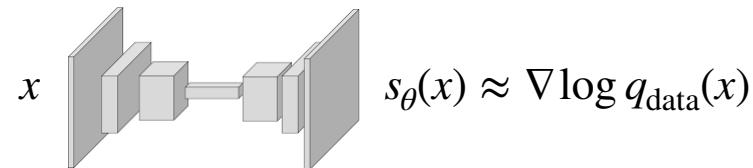
$\{\mathbf{x}^j\}_{j=1}^M \stackrel{\text{i.i.d.}}{\sim} q \rightarrow \nabla_{\mathbf{x}} \log q(\mathbf{x})$
Score function

Score-Based Modeling

Idea: Model the score $s := \nabla \log p$ instead of the density

Advantages:

1. less computation than energy-based modeling
2. enable more flexible models



Nonparametric Score Model

$$\min_{s \in \mathcal{H}} \mathbb{E}_{q_{\text{data}}} \|s(x) - \nabla \log q_{\text{data}}(x)\|^2 + \frac{\lambda}{2} \|s\|_{\mathcal{H}}^2$$

The spectral estimator (Shi et al., 18)
is a special case.

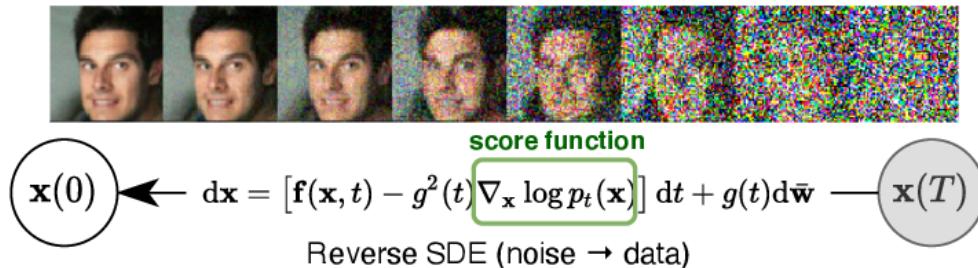
Score Network

Use neural networks to model score,
trained by sliced score matching

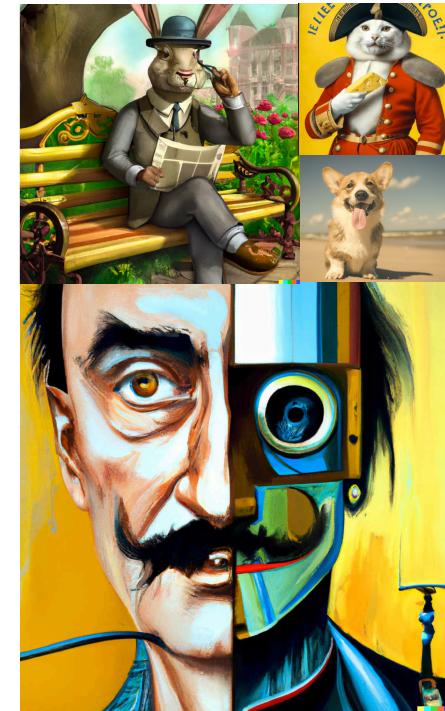
$$\min_{\theta} \mathbb{E}_{q_{\text{data}}} \|s_\theta(x) - \nabla \log q_{\text{data}}(x)\|^2$$

From Score Networks to Diffusion Models

Updates produced by score networks transform noise to data



[Song et al., ICLR'20]



Images created by OpenAI's DALLE-2.
DALLE-2 is based on diffusion models.

Open Problems

- Improving finite-particle rates of SVGD
- Approximately solving the Stein equation for improved gradient estimation
- Lower bounding the discrete Stein discrepancy
- Learning the features in nonparametric score models
- Finding the “right” discrete correspondence of the score matching objective

Joint work with Yuhao Zhou, Jessica Hwang, Shengyang Sun, Yang Song, Sahaj Garg, Michalis K. Titsias, Lester Mackey, Jun Zhu

Main References

- Shi & Mackey. A finite-particle convergence rate for Stein variational gradient descent. NeurIPS 2023.
- Shi, Zhou, Hwang, Titsias, & Mackey. Gradient estimation with discrete Stein operators. NeurIPS 2022
- Titsias & Shi. Double control variates for gradient estimation in discrete latent-variable models. AISTATS 2022
- Shi, Sun, & Zhu. A spectral approach to gradient estimation for implicit distributions. ICML 2018
- Song, Garg, Shi, & Ermon. Sliced score matching: A scalable approach to density and score estimation. UAI 2019
- Zhou, Shi, Zhu. Nonparametric score estimators. ICML 2020