Spectral Methods and Generative Modeling: A Unifying Perspective

Jiaxin Shi

jiaxins.io

2024/5/24 @ University of Warwick

Unsupervised Learning is Efficient Learning

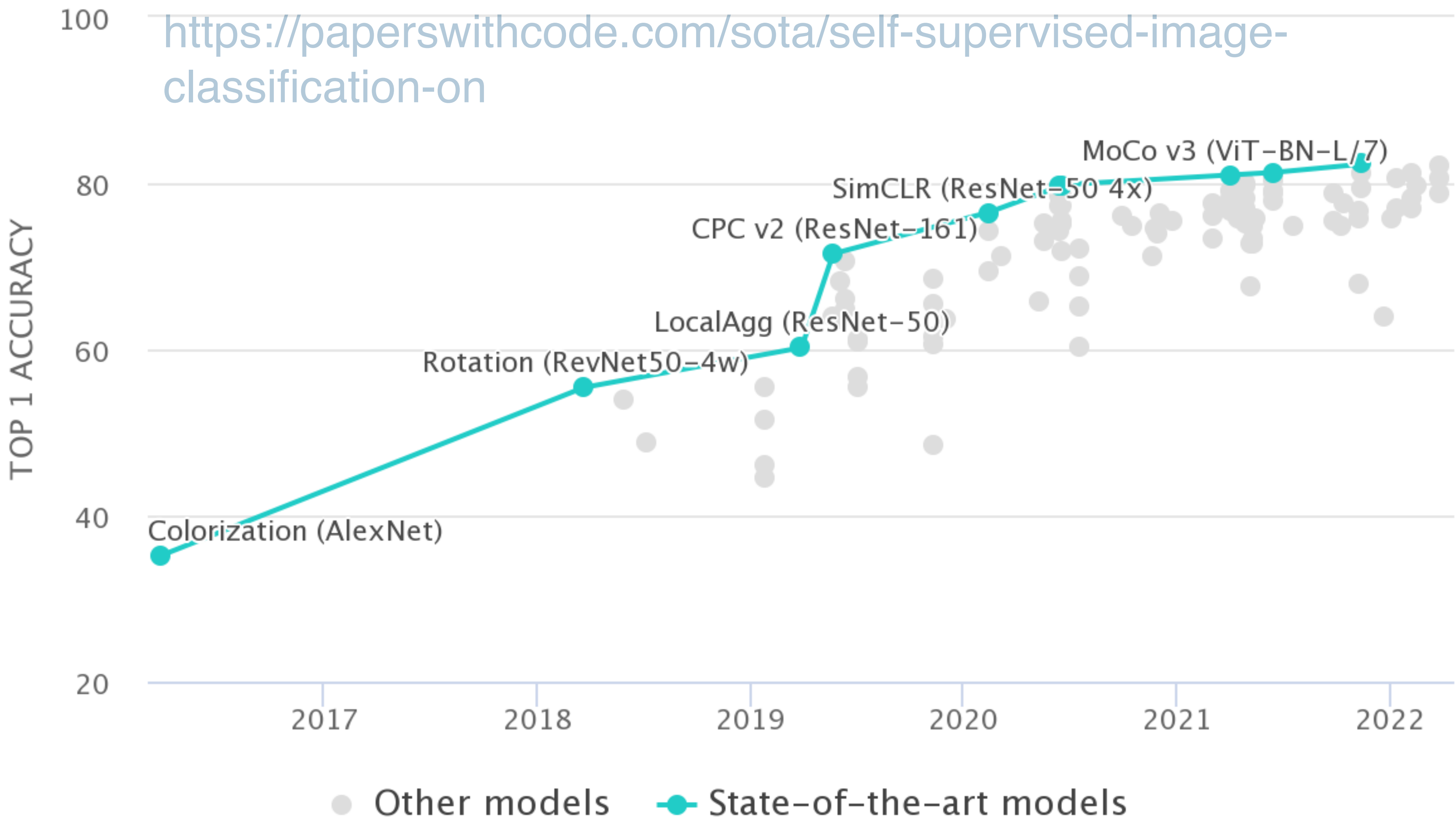
Icing: supervised learning
(10 bits per sample)

Cake: unsupervised learning
(Millions of bits per sample)

Yann LeCun's Cake Analogy

Figure credit: Ian Goodfellow

AI generated face images

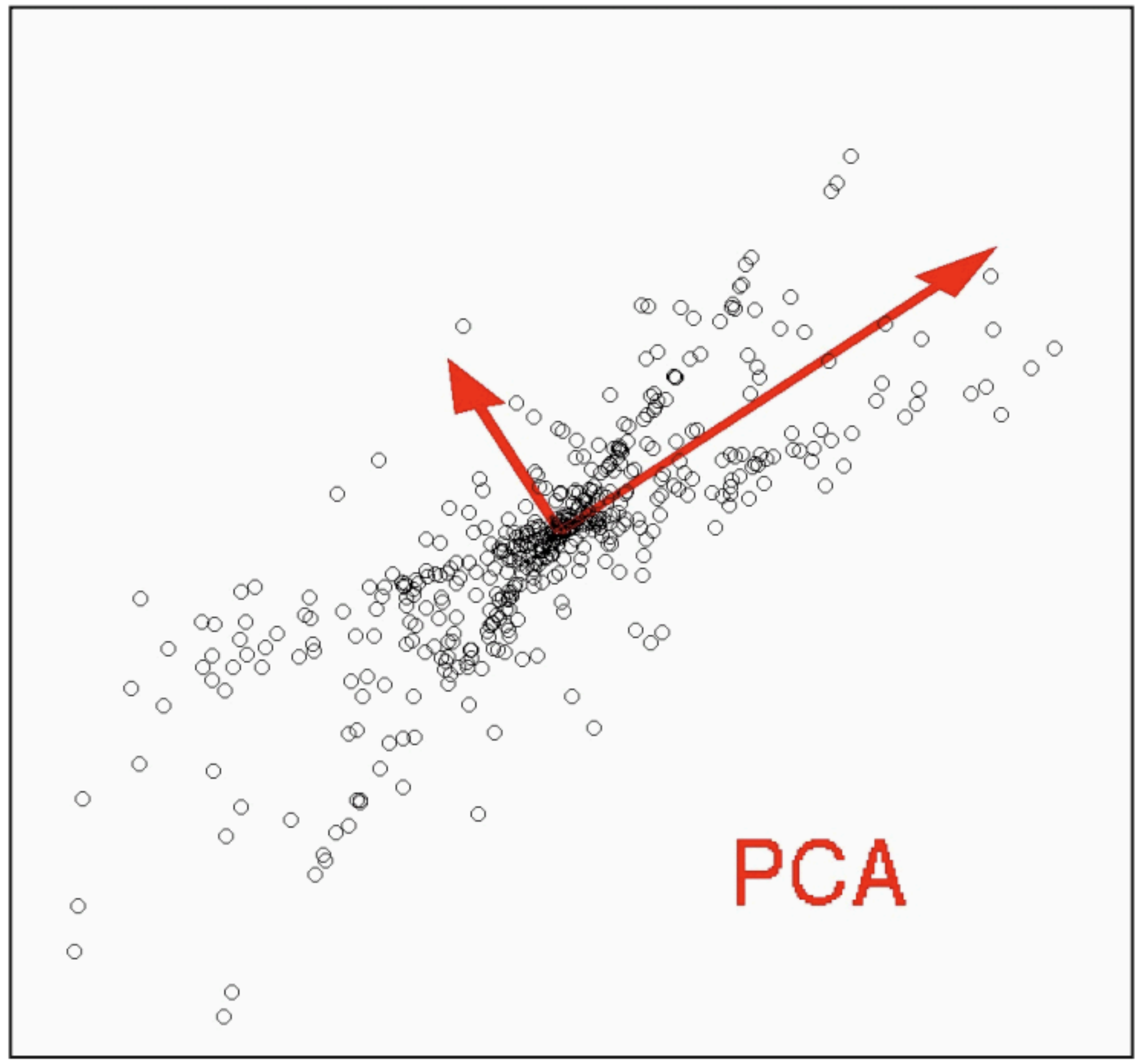


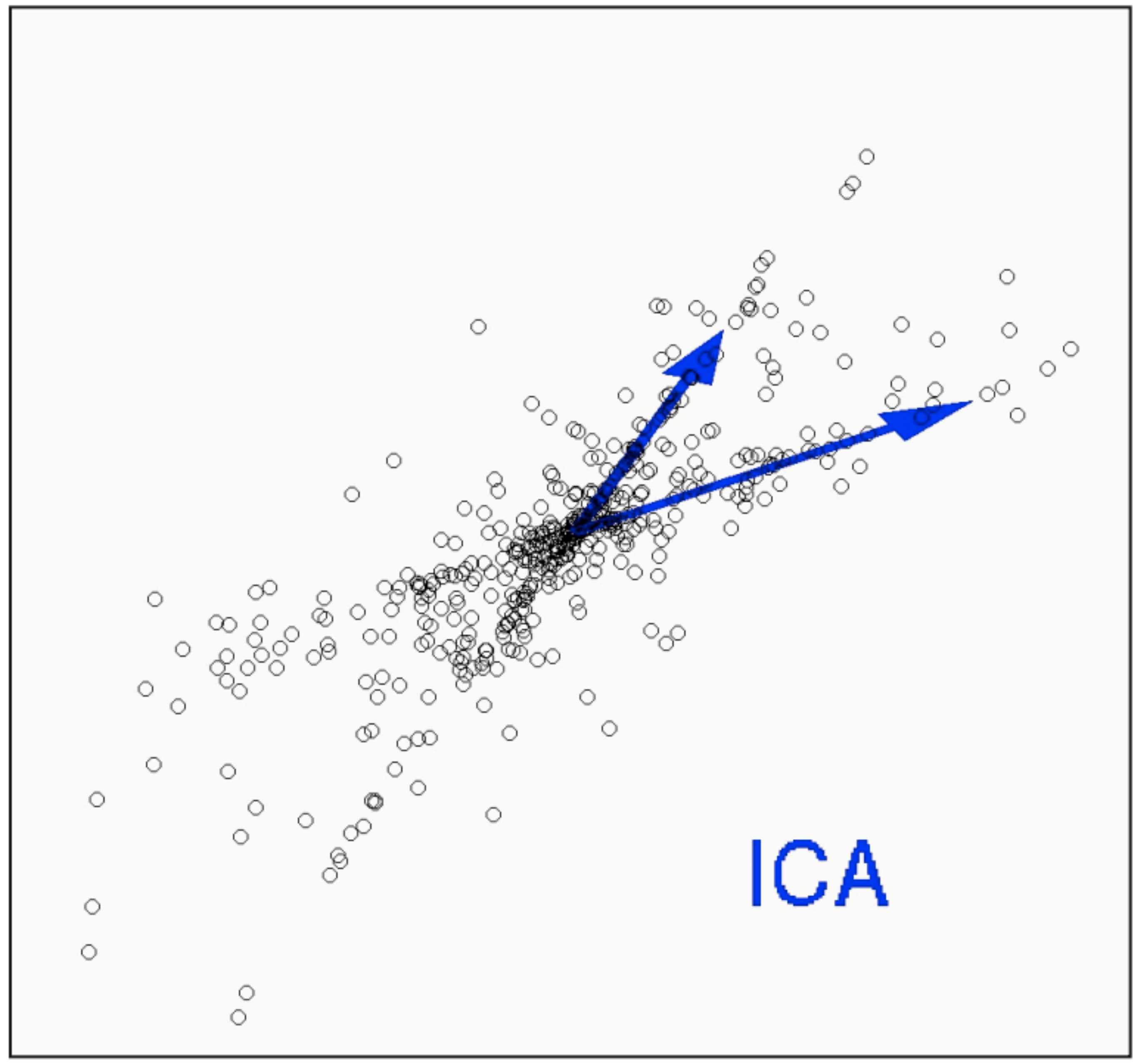
Representation learning performance on ImageNet

Progress has been made. Yet we have not reached a consensus on

- the goal for unsupervised learning
- which learning rule leads to intelligence

An Extreme Example





$$\mathbb{E}_{x'\sim p}[k(x,x')\psi(x')] = \lambda\psi(x)$$

$$Ku = \lambda u$$

$$\min D(p_{\text{model}} \| p_{\text{data}})$$

Spectral Methods

Learn eigenfunctions

usually nonparametric,
no distributional assump.

Generative Modeling

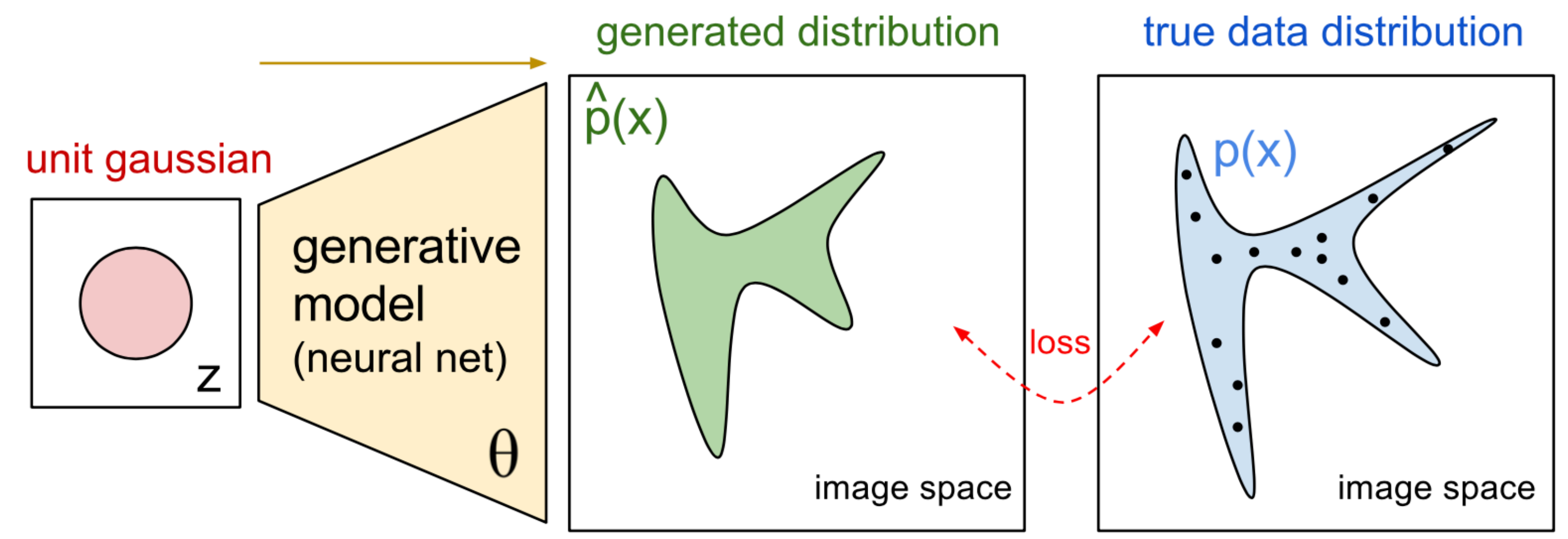
Estimate densities

usually parametric

sklearn.decomposition.KernelPCA

sklearn.manifold.SpectralEmbedding

sklearn.manifold.LocallyLinearEmbedding



https://openai.com/blog/generative-models/

VAE, Normalizing Flow, GAN, EBM.

Outline

Spectral Methods — ① Spectral Representation of Density Gradients $\nabla \log p(X)$ (Score) — Generative Models

Representation Learning
(Self-Supervised Learning)

Score-based Modeling

Why Care About Density Gradients $\nabla \log p(X)$ (Score)

- It contains all information about the data distribution

$$dX_t = \nabla \log p(X_t)\, dt + \sqrt{2} dB_t \quad \text{(Langevin diffusion)}$$

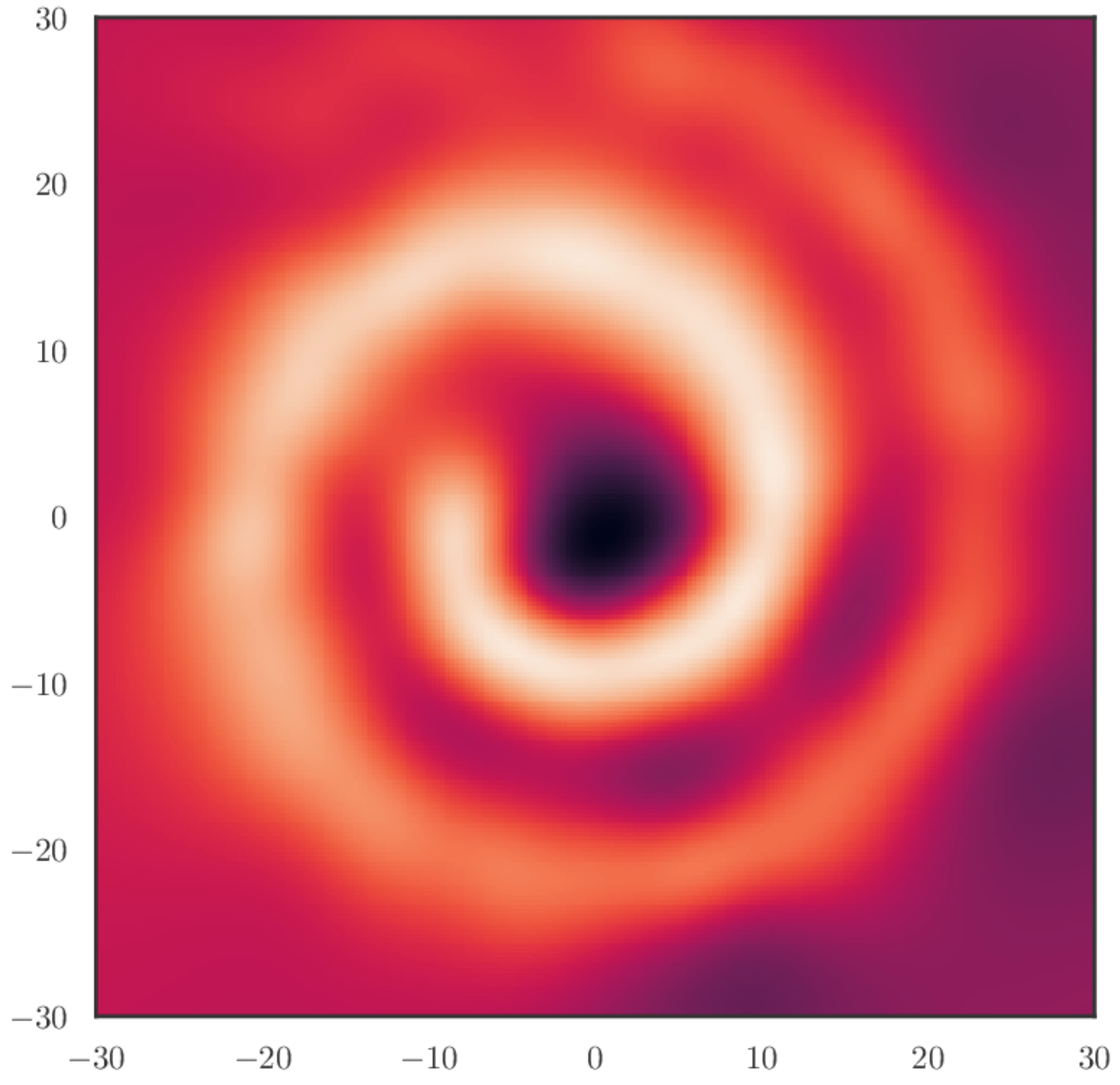
- In many learning problems, this is the only quantity related to the data distribution that needs to be calculated, such as mutual information-based learning

$$\nabla_\phi I(X;Y) = \mathbb{E}_{X\sim P_X}[\nabla_Y \log p_{X,Y} \nabla_\phi g_\phi(X)] - \mathbb{E}_{X\sim P_X}[\nabla \log p_Y \nabla_\phi g_\phi(X)]$$

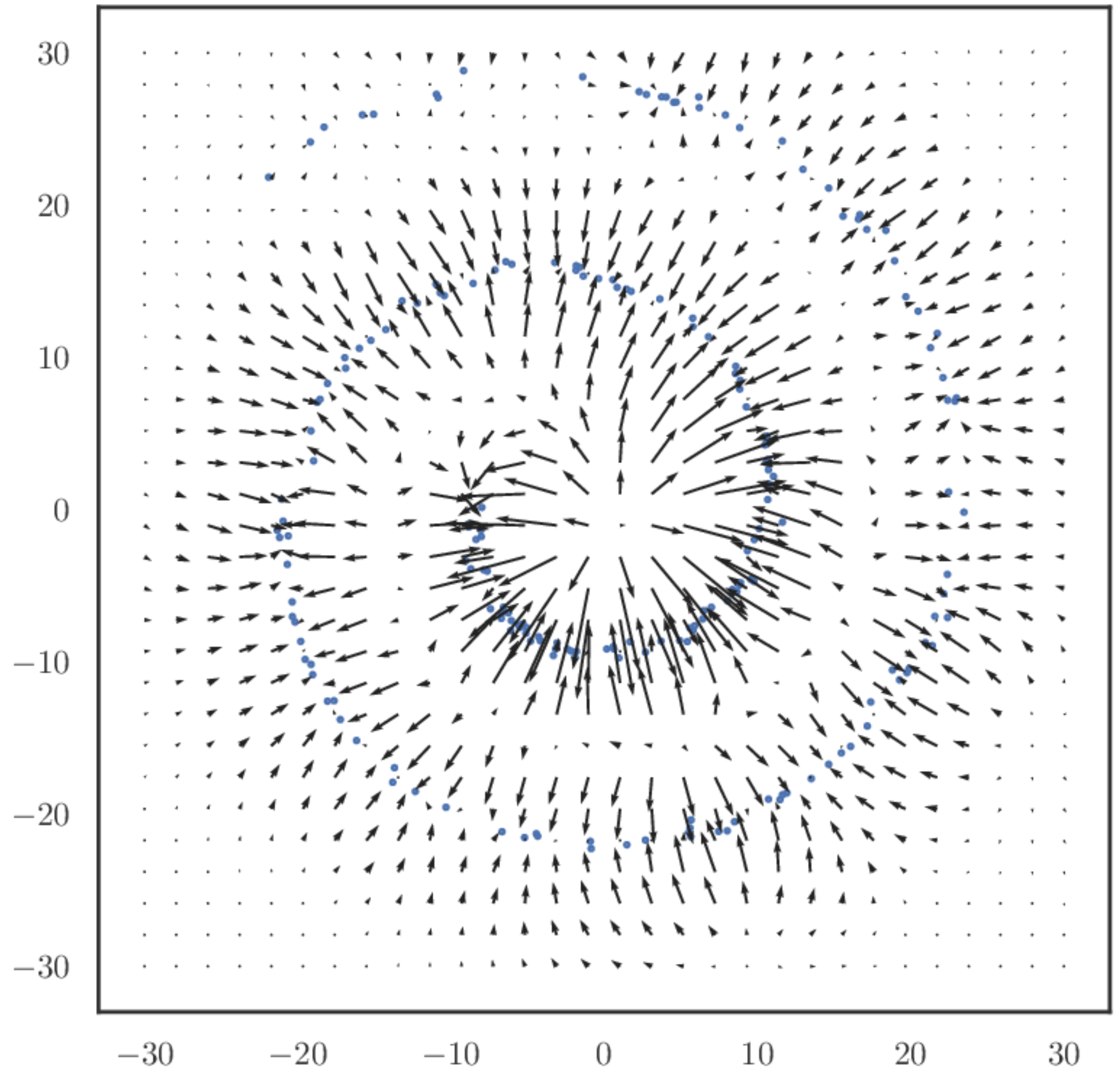
[Li & Turner, 17; Hjelm et al., 19; Tschannen et al., 19; Wen et al., 20]

- Free of normalization, so easier to model than the distribution itself

Spectral Methods for Estimating Density Gradients
(Score Estimation)



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



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

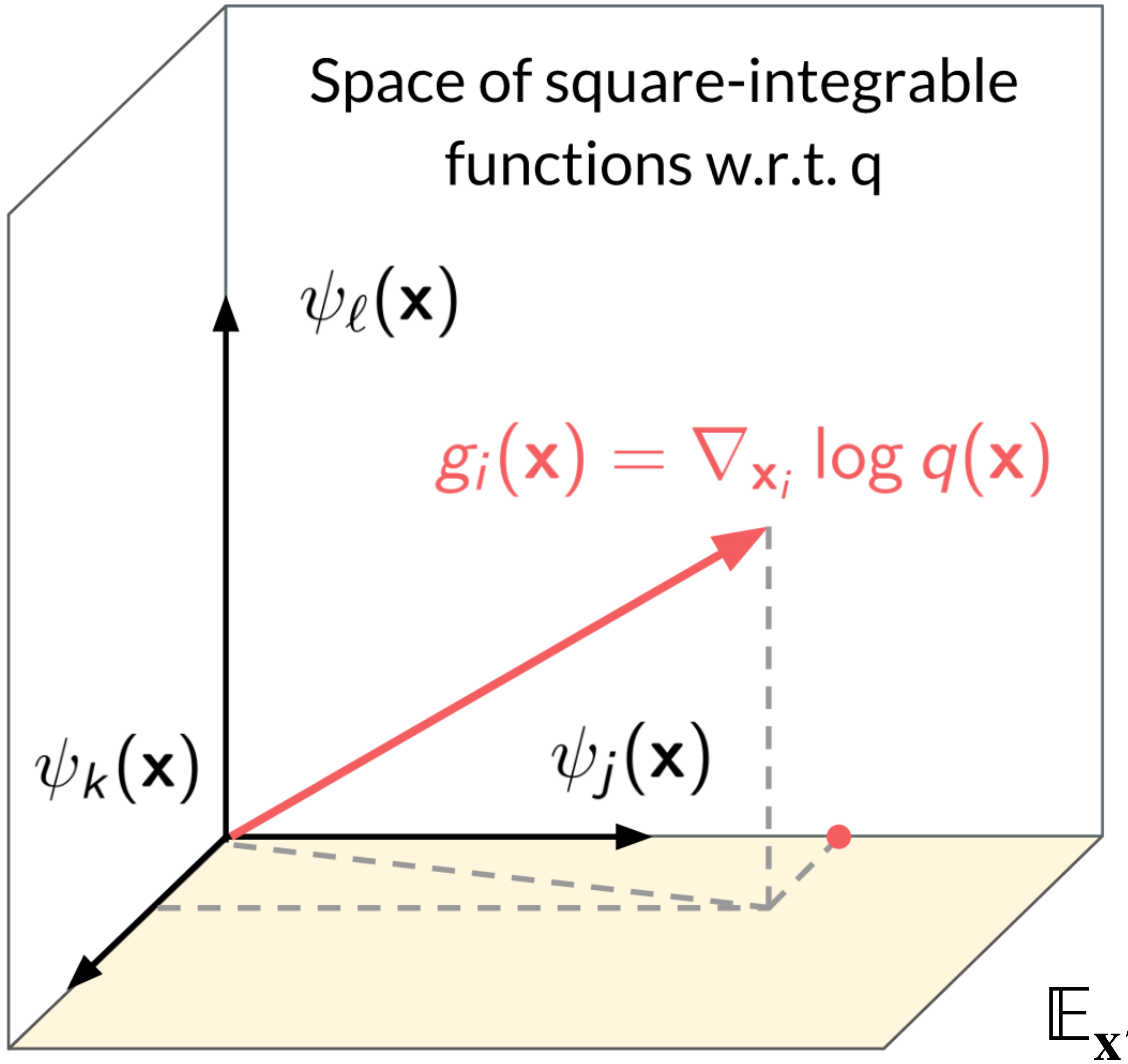
Score function

Shi, Sun & Zhu. A spectral approach to gradient estimation for implicit distributions. ICML 2018

Zhou, Shi & Zhu. Nonparametric score estimators. ICML 2020

Spectral Methods for Estimating Density Gradients (Score Estimation)

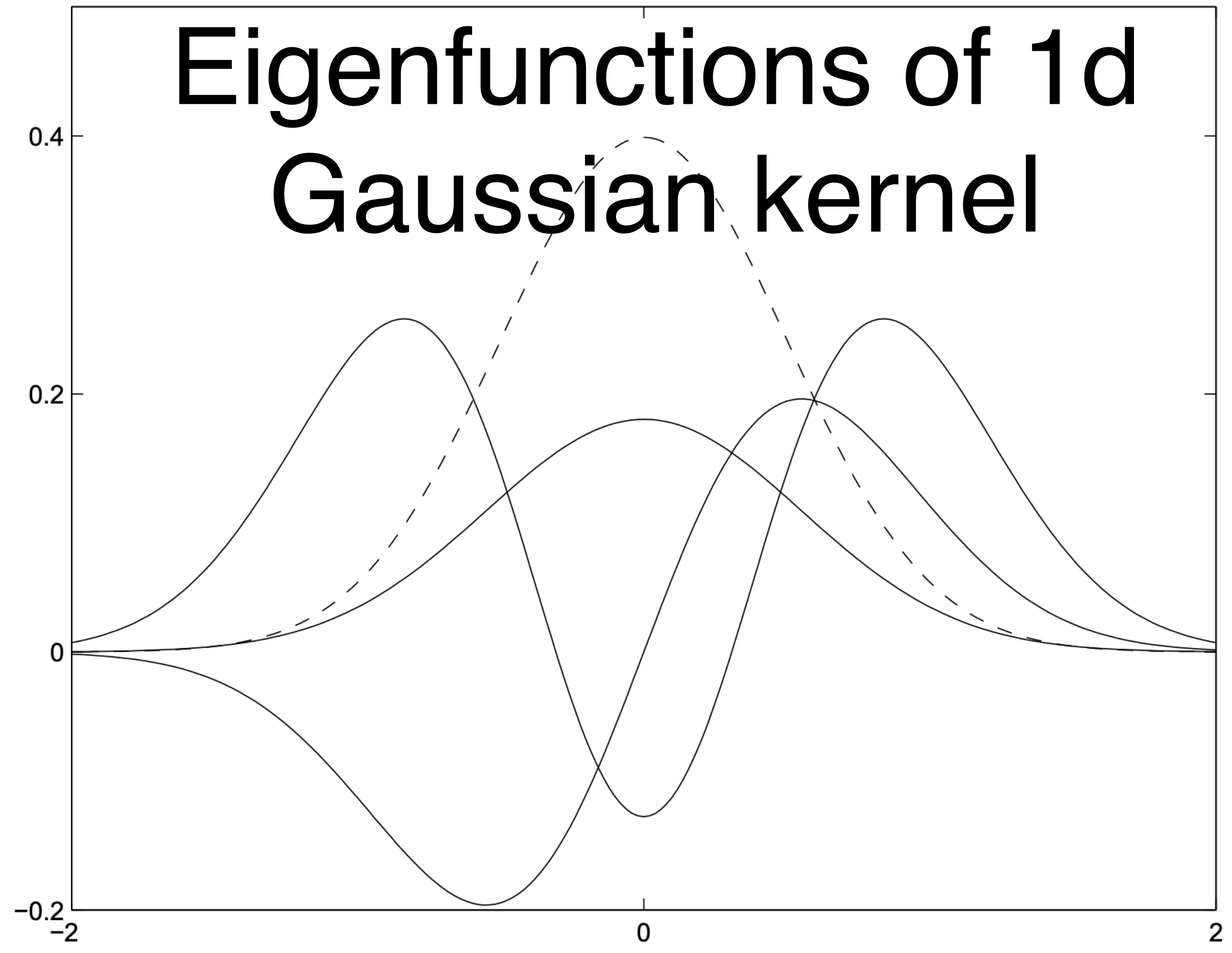
S, Sun & Zhu, ICML'18



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

density gradients (score) — eigenfunction

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



Eigenfunctions $\{\psi_j\}_{j\geq 1}$ form a basis of the function space

Shi, Sun & Zhu. A spectral approach to gradient estimation for implicit distributions. ICML 2018

Application: Mutual Information Gradient Estimation for Representation Learning

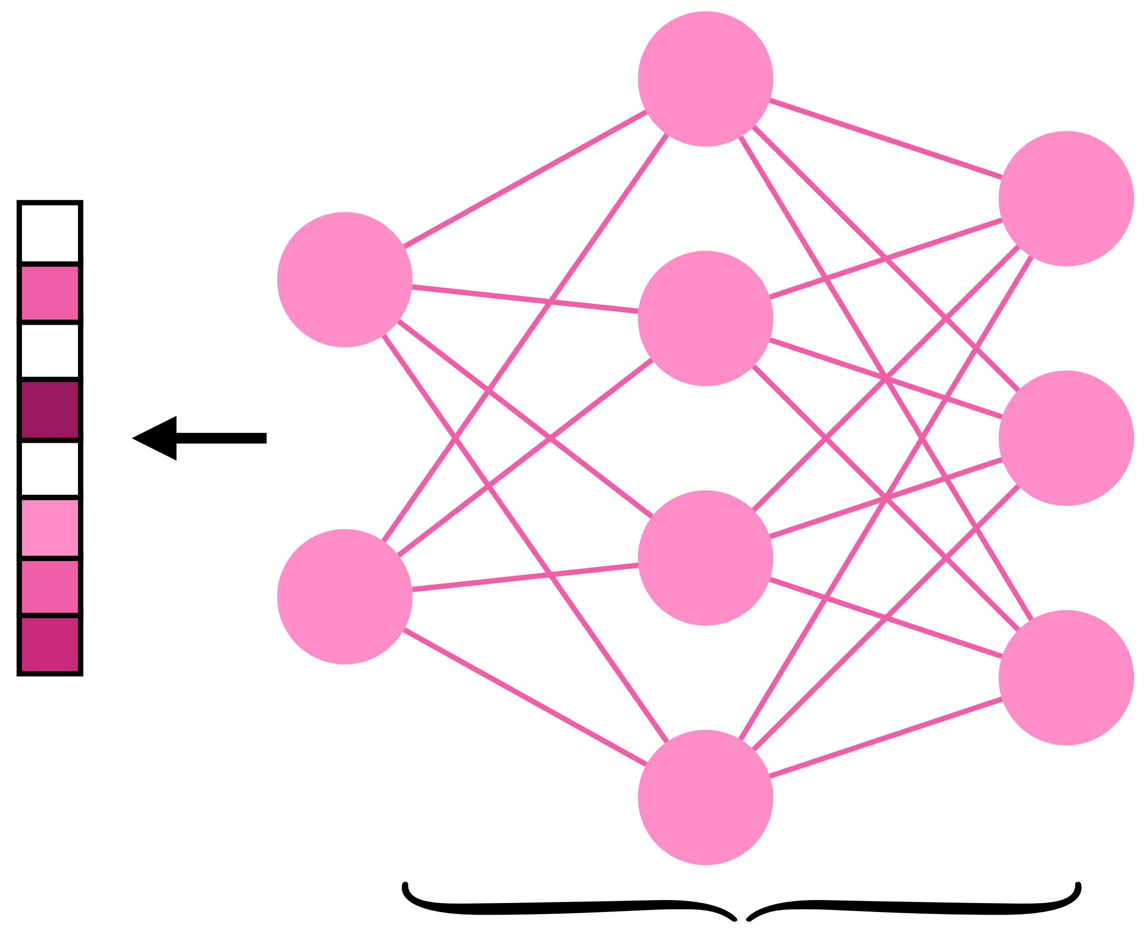
[Wen et al., ICLR'20]

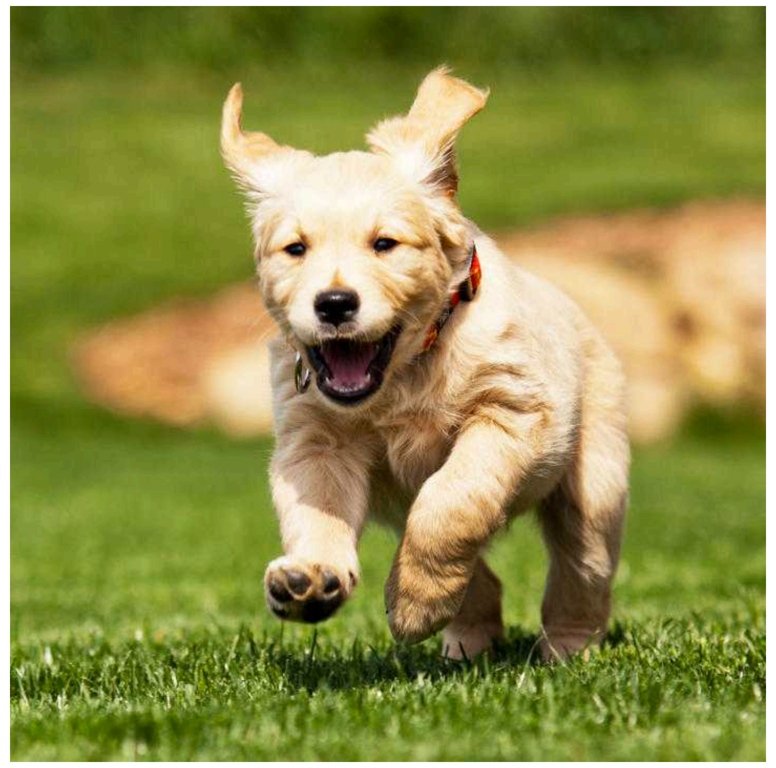
representation

Encoder g_ϕ

$X \sim P_X$

Y





learn by maximizing mutual information

$I(X, Y)$

Model	STL-10 conv	STL-10 fc(1024)	STL-10 Y(64)
DIM (JSD)	42.03%	30.28%	28.09%
DIM (infoNCE)	43.13%	35.80%	34.44%
MIGE + RP to 512d	52.00%	48.14%	44.89%

Model	CIFAR-10			CIFAR-100		
	conv	fc(1024)	Y(64)	conv	fc(1024)	Y(64)
DIM (JSD)	55.81%	45.73%	40.67%	28.41%	22.16%	16.50%
DIM (JSD + PM)	52.2%	52.84%	43.17%	24.40%	18.22%	15.22%
DIM (infoNCE)	51.82%	42.81%	37.79%	24.60%	16.54%	12.96%
DIM (infoNCE + PM)	56.77%	49.42%	42.68%	25.51%	20.15%	15.35%
MIGE	**57.95%**	**57.09%**	**53.75%**	**29.86%**	**27.91%**	**25.84%**

Used in winning solution of NeurIPS 2021 BEETL Competition: Benchmarks for EEG Transfer Learning

Key Insight：Stein's Lemma

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

- Introduced by Stein (1972) for characterizing distributional convergence.
- The identity he studied for normal distribution $x \sim N(0, \sigma^2)$:

$$\mathbb{E}[xh(x)] = \sigma^2 \mathbb{E}[h'(x)] \quad \text{for} \quad x \sim N(0, \sigma^2)$$

Outline

Spectral Methods — Spectral Representation of Density Gradients $\nabla \log p(X)$ (Score) — Generative Models

Representation Learning
(Self-Supervised Learning)

② Score-based Modeling

Stein's Lemma as a Learning Rule

$$\mathbb{E}_{q}[h(x)^\top \nabla \log p(x) + \nabla \cdot h(x)] = 0$$ for any suitable h if $q = p$

- Let q←data distribution, p←model distribution, minimize |LHS|
  Result: Fit generative model to data
  Question: How to choose h?

Stein's Lemma as a Learning Rule

?

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

Data distribution

Model distribution

Score Matching

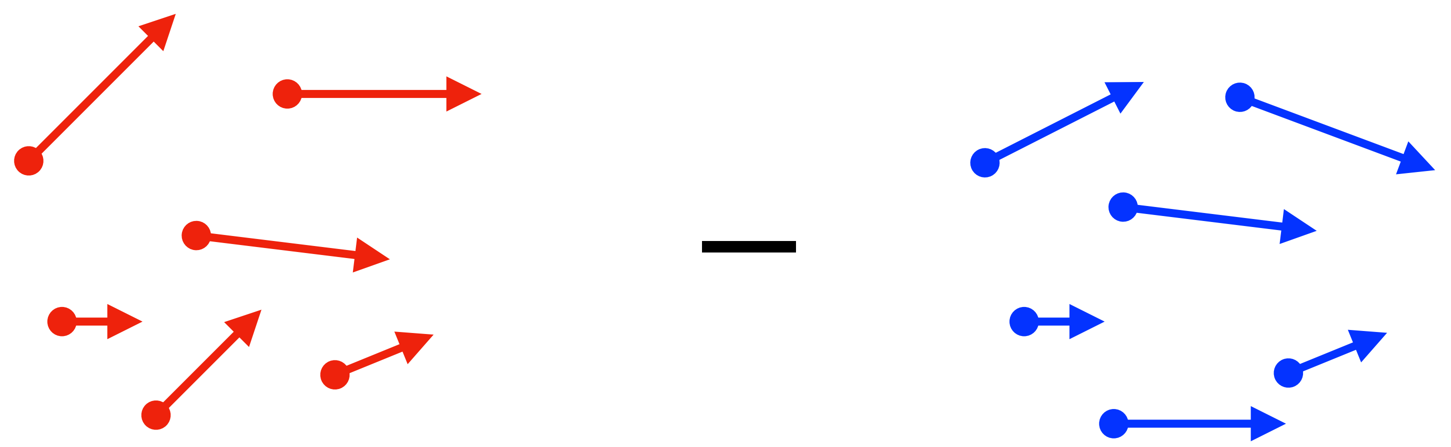
[Hyvärinen, 2005]

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

Data distribution

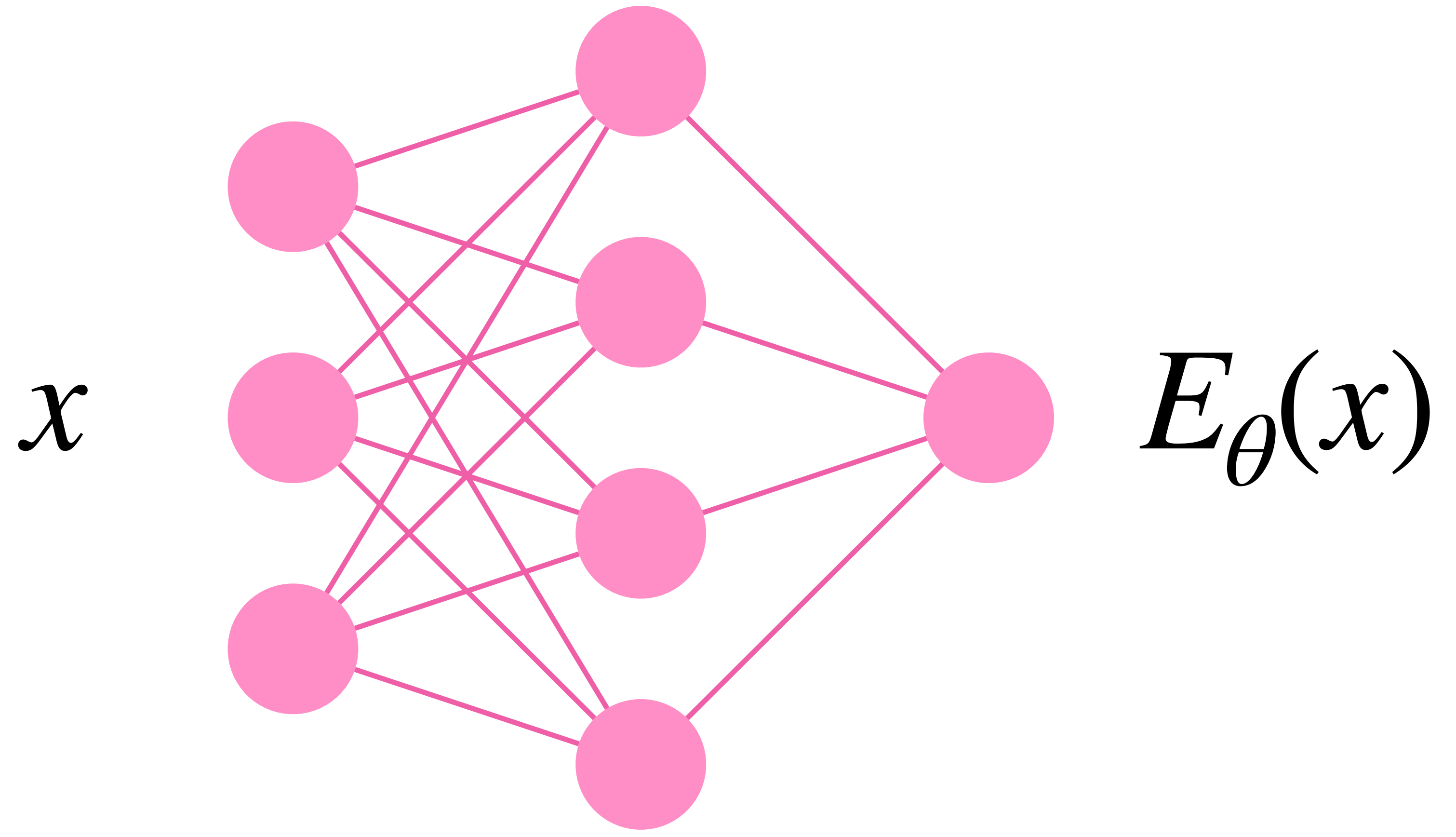
Model distribution

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



Training Energy-Based Models

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



Sliced Score Matching

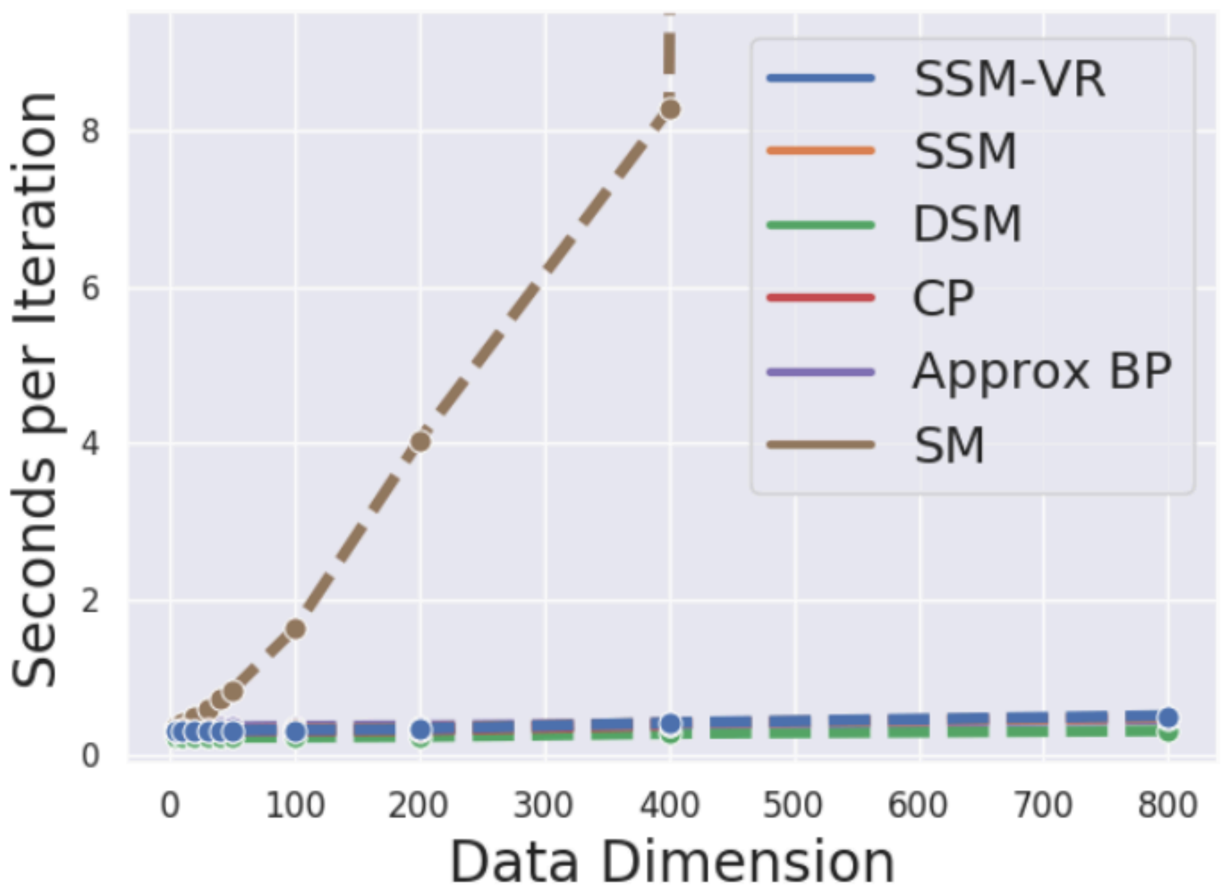
[Song*, Garg*, S & Ermon, UAI'19]

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

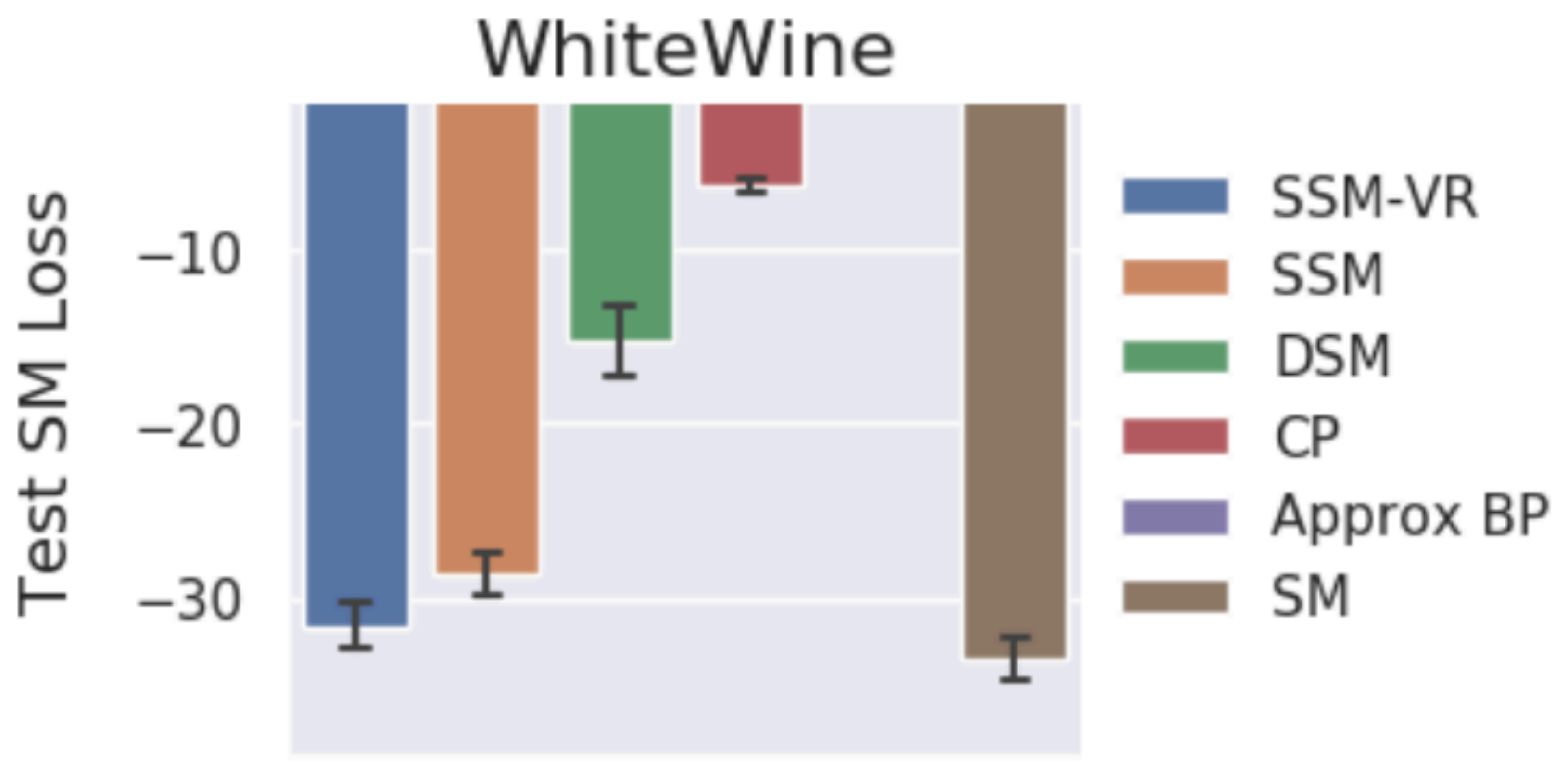
$$\nabla_x \log p_\theta(x) = -\nabla E_\theta(x) + \cancel{\nabla_x \log Z_\theta}$$

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

Time



Performance



Song*, Garg*, Shi & Ermon. Sliced score matching: A scalable approach to density and score estimation. UAI 2019

Score-Based Modeling

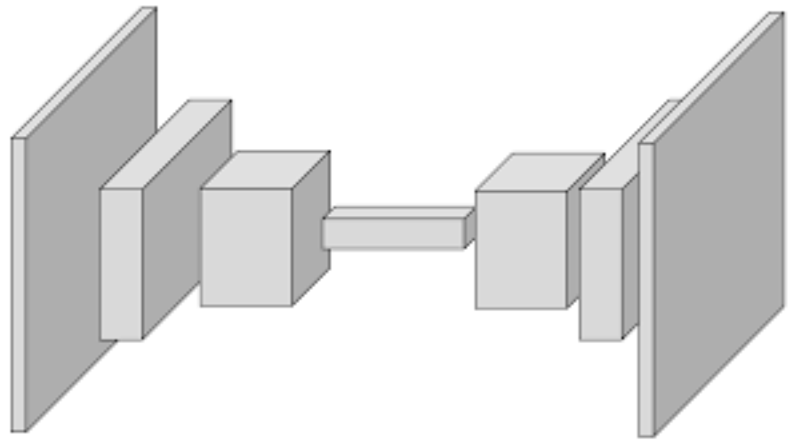
Song*, Garg*, S & Ermon, UAI'19; Zhou, S & Zhu, ICML'20

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

x



$s_\theta(x) \approx \nabla \log q_{\mathrm{data}}(x)$

Nonparametric Score Model

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

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

Score Network

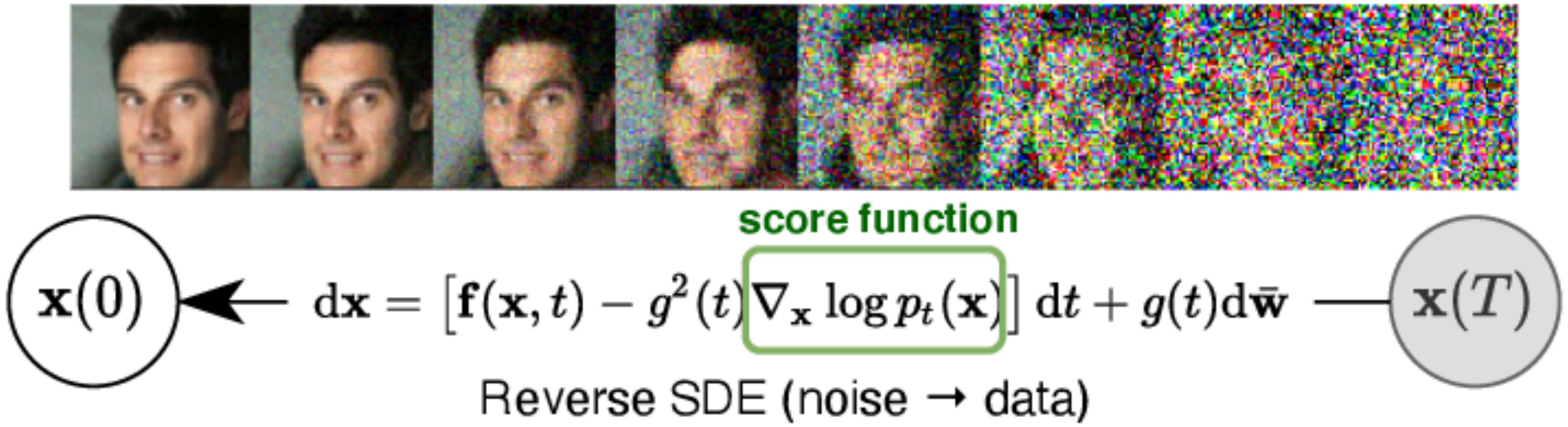
Use neural networks to model score, trained by sliced score matching

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

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

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.

Challenge: Discrete Domains

- No continuous density; scores won't exist.

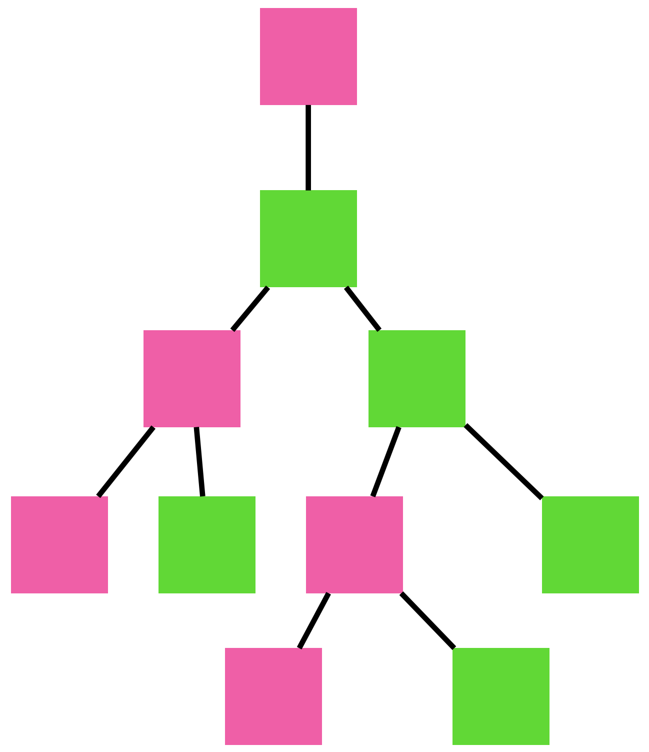
Molecules

Natural Language

```
def max_sum_slice(xs):
    if not xs:
        return 0

    max_ending = max_slice = 0
    for x in xs:
        max_ending = max(0, max_ending + x)
        max_slice = max(max_slice, max_ending)
    return max_slice
```

Computer Programs



Choices & Decision

Generalize into Discrete Domains

Shi et al., NeurIPS'22

What Stein's method really means:

The formula

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

is the change rate of $\mathbb{E}[f(x_t)]$ at $x_t = x$ when x_t follows the Langevin diffusion with stationary p

$$\begin{tikzcd}
\text{Langevin diffusion} \arrow[r] \arrow[d] & \text{Discrete state-space Markov chains} \arrow[d] \\
\text{Stein's method} \arrow[r] \arrow[d] & ? \arrow[d] \\
\text{Score-based modeling} \arrow[r] & ?
\end{tikzcd}$$

Shi, Zhou, Hwang, Titsias & Mackey. Gradient estimation with discrete Stein operators, **NeurIPS 2022 Outstanding Paper Award**.

Discrete Stein Operators

Shi et al., NeurIPS'22

Discrete state-space Markov chains	Stein operator $(\mathcal{A}h)(x)$
Gibbs	$\frac{1}{d}\sum_{i=1}^{d}\sum_{y_{-i}=x_{-i}} q(y_i \mid x_{-i})h(y) - h(x)$
MPF	$\sum_{y\in\mathcal{N}_x, y\neq x}\sqrt{q(y)/q(x)}(h(y)-h(x))$
Barker	$\sum_{y\in\mathcal{N}_x, y\neq x}\frac{q(y)}{q(x)+q(y)}(h(y)-h(x))$
Birth-death	$\frac{1}{d}\sum_{i=1}^{d} h(\mathtt{dec}_i(x)) - \frac{q(\mathtt{inc}_i(x))}{q(x)}h(x)$

$$E_q[(Ah)(x)] = 0$$

Applications

- Learning discrete energy-based/diffusion models
- Gradient estimation for discrete optimization:
  discrete latent-variable models, combinatorial optimization, reinforcement learning, etc.

Stochastic gradients

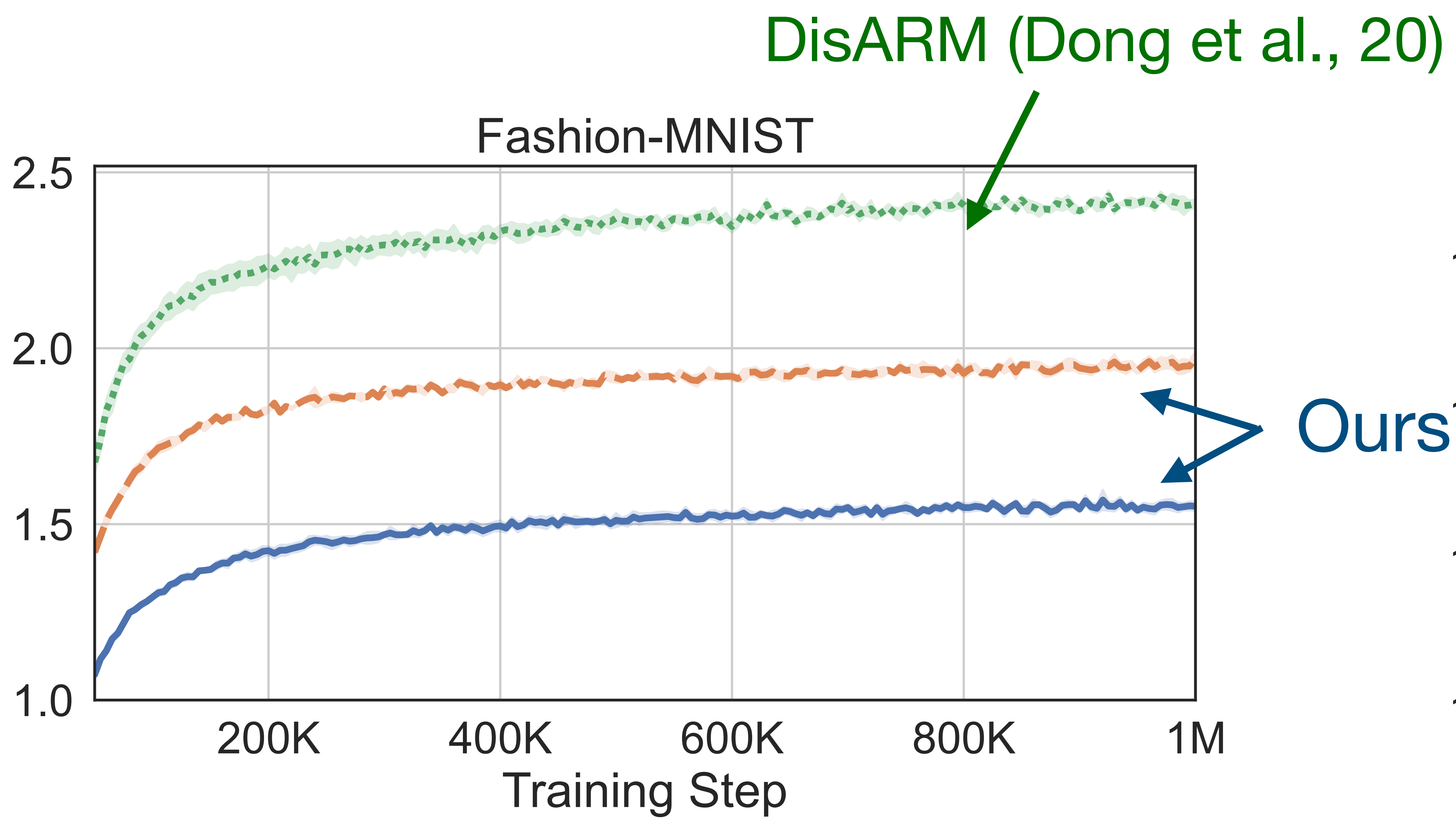
$$\min_h \mathrm{Var}(\hat{g}(x) + (Ah)(x))$$

Shi, Zhou, Hwang, Titsias & Mackey. Gradient estimation with discrete Stein operators, **NeurIPS 2022 Outstanding Paper Award**.

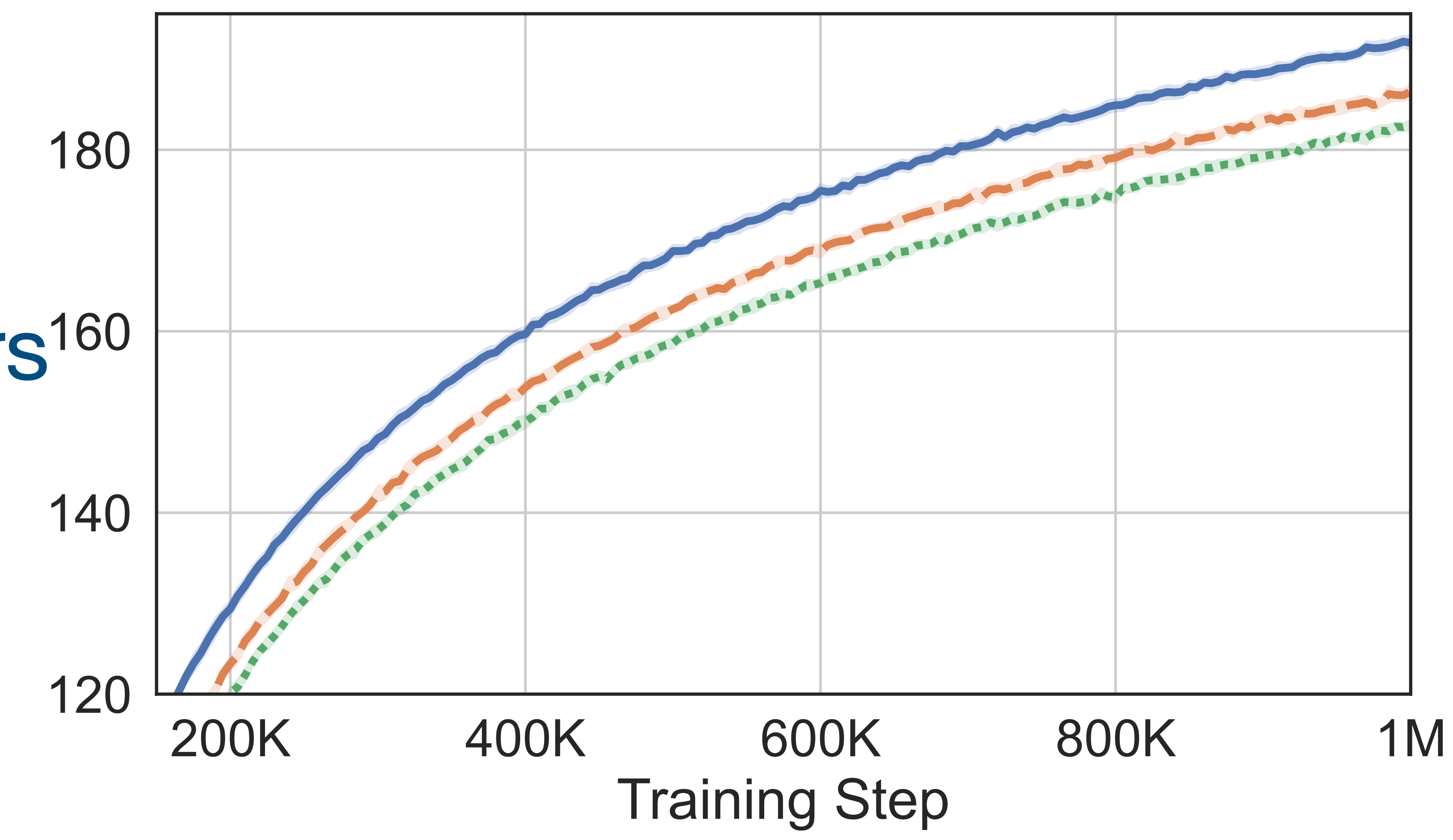
SOTA Gradient Estimators for Learning Discrete Latent-Variable Models

via discrete Stein operators (Shi et al., NeurIPS'22)

Variance of gradient estimates



Training objective



Learning discrete representation with VAEs, 200 latent dimensions

Shi, Zhou, Hwang, Titsias & Mackey. Gradient estimation with discrete Stein operators, **NeurIPS 2022 Outstanding Paper Award**.

Outline

Spectral Methods — Spectral Representation of Density Gradients $\nabla \log p(X)$ (Score) — Generative Models

Representation Learning (Self-Supervised Learning)

③

Score-based Modeling

A Parametric Approach to Spectral Learning?

$$\psi_j(x) \longrightarrow \nabla \log p(x)$$

Eigenfunction → Score

- Scaling is a problem for nonparametric methods
- Nonparametric methods do not leverage inductive bias such as equivariance

Probably the reason why spectral learning are less used today even if they seem to capture more information than generative modelling.

NeuralEF：Learning Neural Eigenfunctions

Deng, S & Zhu, ICML'22

- NeuralEF:

$$\max_{\psi_j} R_{jj} - \sum_{i=1}^{j-1} \frac{R_{ij}^2}{R_{ii}} \quad s.t. \, \mathbb{E}[\psi_j(x)^2] = 1, \quad j = 1,\ldots,J$$

L2-BatchNorm

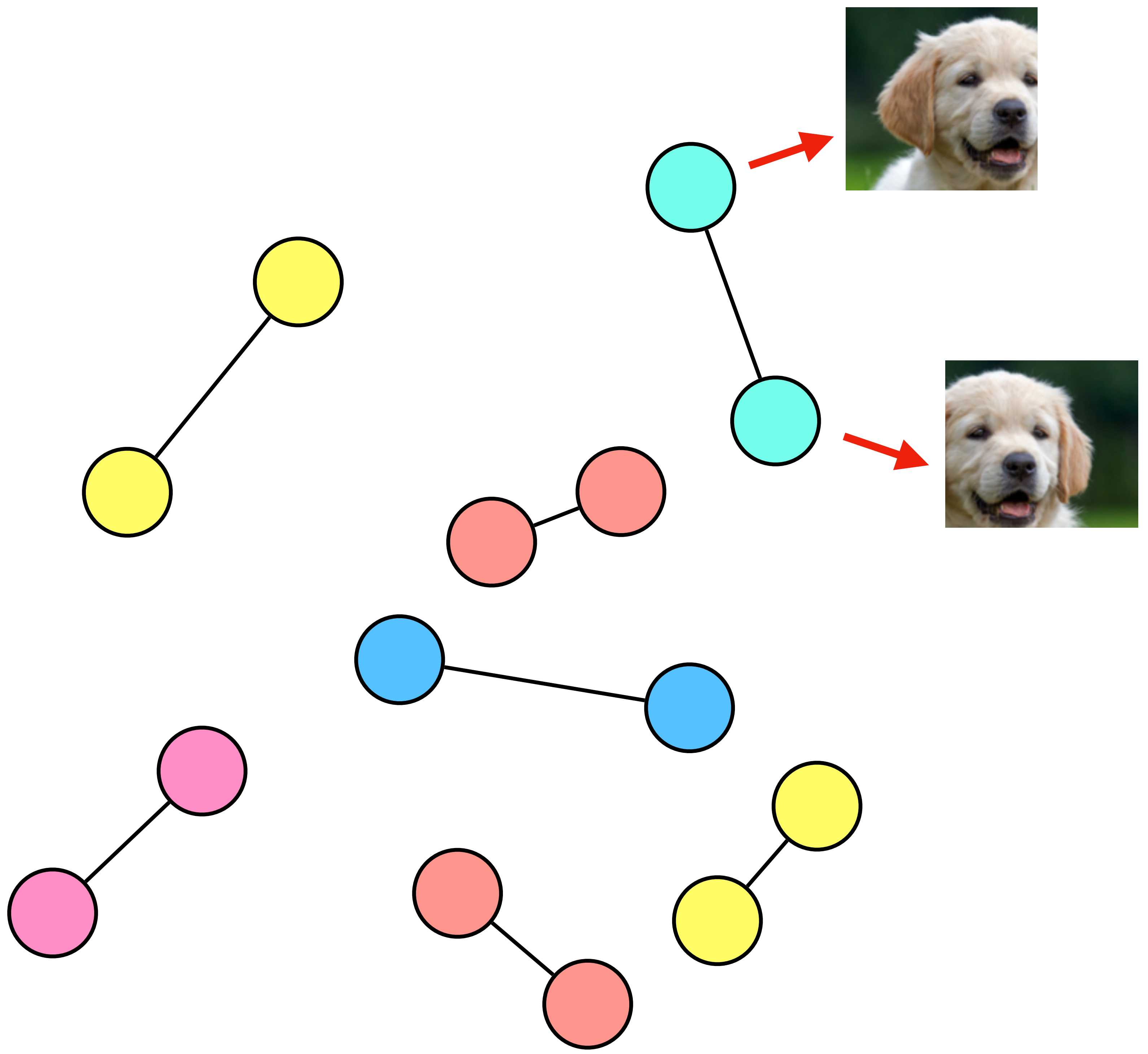
$$R_{ij} = \mathbb{E}[\psi_i(x) k(x, x') \psi_j(x')]$$

- Can be seen as a function-space extension to EigenGame (Gemp et al., 2020)

Deng, Shi & Zhu. NeuralEF: Deconstructing kernels by deep neural networks. ICML 2022

Neural Eigenmaps

Eigenfunctions are strong self-supervised learners



$$\kappa(x, x') = \frac{E_{p(z)}[p(x\,|\,z)p(x'\,|\,z)]}{p(x)p(x')}$$

$p(x\,|\,z)$: data augmentation

[HaoChen et al., 2021; Johnson et al., 2022]

Method	batch size	top-1 accuracy
SimCLR	4096	66.5
MoCo v2	256	67.4
BYOL	4096	66.5
SimSiam	256	68.1
SCL	384	67.0
Neural Eigenmap	2048	67.6
Neural Eigenmap w/o `stop_grad`	2048	**68.4**

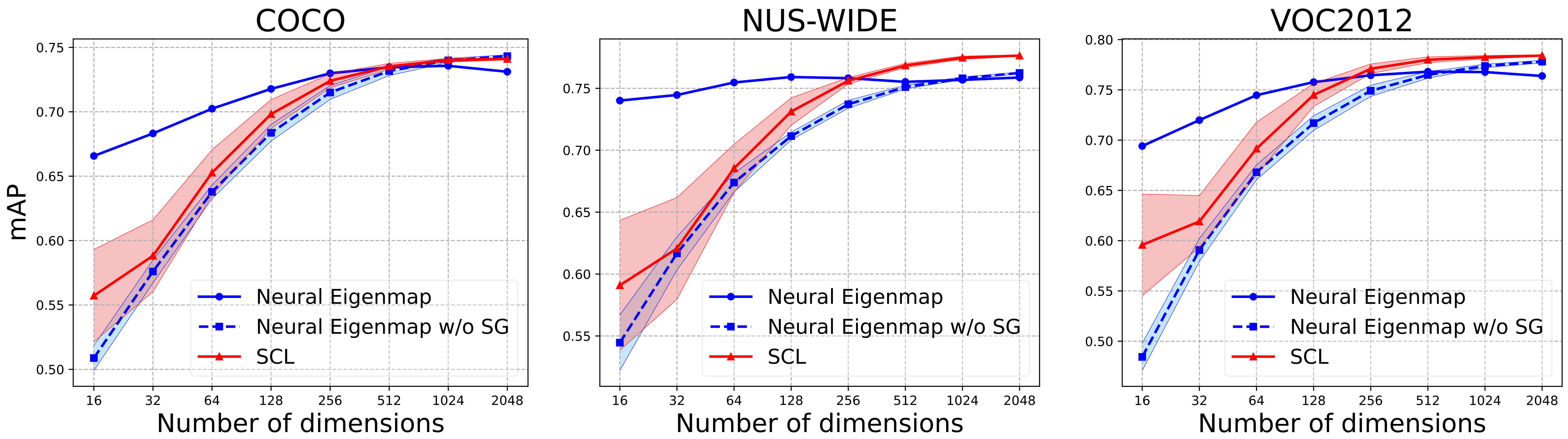
ImageNet Top-1 accuracies of linear classifiers trained on neural eigenfunction outputs (100 epoch results).

Deng*, Shi*, Zhang, Cui, Lu & Zhu. Neural Eigenfunctions Are Structured Representation Learners. *arXiv:2210.12637*, 2022.

Neural Eigenmaps

Deng*, S* et al., 2022

- Structured representations—features are ordered by importance
- Can be used as adaptive-length codes in image retrieval systems

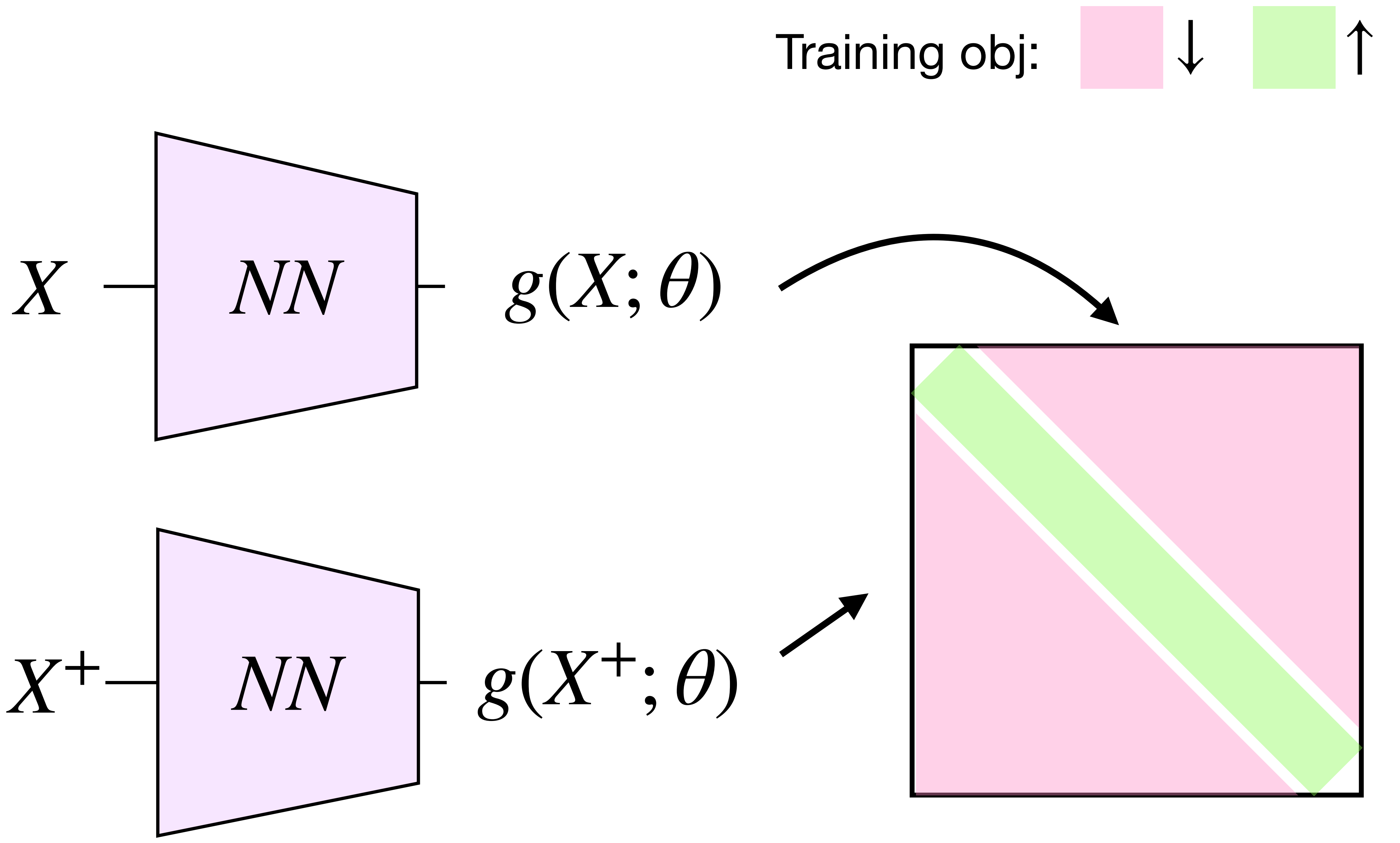


Maintaining similar retrieval performance as leading SSL methods after truncating up to 94% of the representation length

Deng*, Shi*, Zhang, Cui, Lu & Zhu. Neural Eigenfunctions Are Structured Representation Learners. *arXiv:2210.12637*, 2022.

Neural Eigenmaps: Algorithm

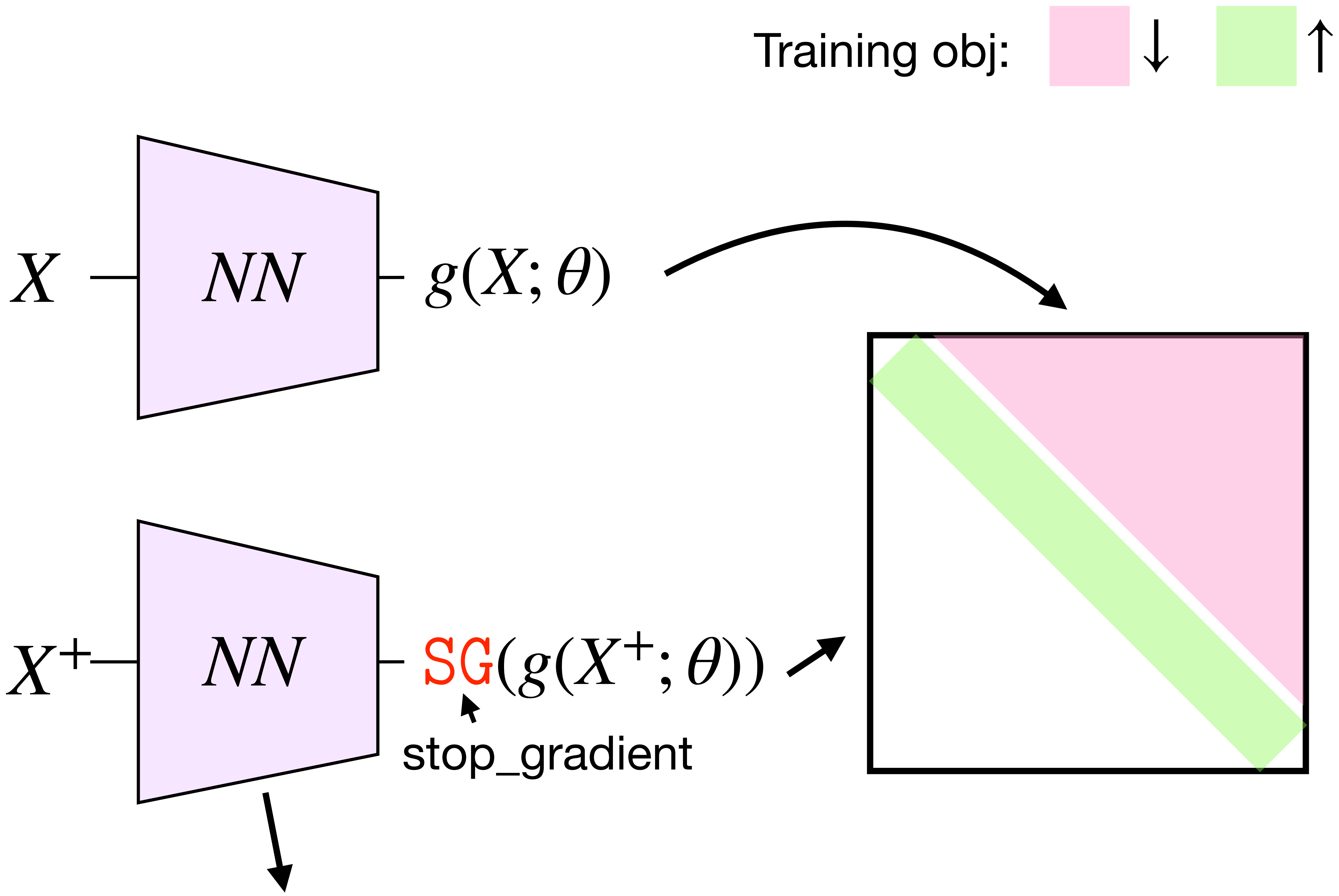
Self-Supervised Learning



X and X^+ are positive pairs

Empirical Covariance of $g(X;\theta)$ and $g(X^+;\theta)$

Neural Eigenmap (Ours)



converges to ordered eigenfunctions

Deng*, Shi*, Zhang, Cui, Lu & Zhu. Neural Eigenfunctions Are Structured Representation Learners. *arXiv:2210.12637*, 2022.

Takeaways

$$k(x, x')$$

$$+$$

$$\boxed{x_1, \ldots, x_n \sim p \text{ (data distribution)}} \longrightarrow \underset{\text{Eigenfunction}}{\psi_j(x)} \longrightarrow \underset{\text{Score}}{\nabla \log p(x)}$$

Score-based modelling

- Replacing nonparametric methods with a deep functional representation is fruitful.
- The underlying principle (Stein's method) can be generalized to discrete domains.

Open Questions

- To what extent can spectral methods explain cross-domain self-supervised learning (e.g., CLIP)?
- Will generative modelling and representation learning eventually converge to a single method?

Thanks!

Collaborators: Jun Zhu, Lester Mackey, Michalis K. Titsias, Shengyang Sun, Yang Song, Yuhao Zhou, Jessica Hwang, Chang Liu, Zhijie Deng

References

- Shi, Sun, & Zhu. A spectral approach to gradient estimation for implicit distributions. ICML 2018
- Titsias & Shi. Double control variates for gradient estimation in discrete latent-variable models. AISTATS 2022
- Shi, et al. Gradient estimation with discrete Stein operators. NeurIPS 2022
- Shi, Liu, & Mackey. Sampling with mirrored Stein operators. ICLR 2022
- 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
- Deng, Shi, & Zhu. NeuralEF: Deconstructing kernels by deep neural networks. ICML 2022
- Deng*, Shi*, Zhang, Cui, Lu, & Zhu. Neural Eigenfunctions Are Structured Representation Learners. arXiv:2210.12637, 2022