Deep Spectral Methods: Another Way to Unsupervised Learning

Zhijie Deng
Qing Yuan Research Institute
Shanghai Jiao Tong University
zhijied@sjtu.edu.cn

Unsupervised learning is critical for creating human-level intelligence

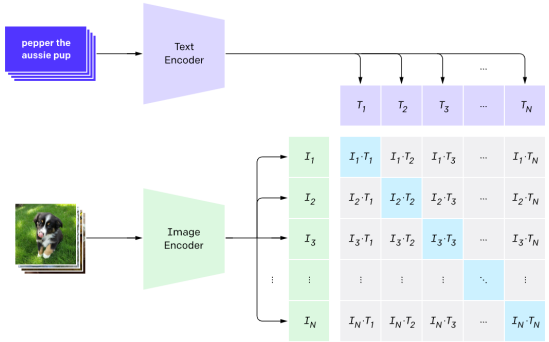
Cherry: reinforcement learning
(A few bits for some samples)

Cream: supervised learning
(10->10,000 bits per sample)

Cake: unsupervised learning
(Millions of bits per sample)

[Yann LeCun's Cake Analogy, NIPS '16]

The recent breakthroughs

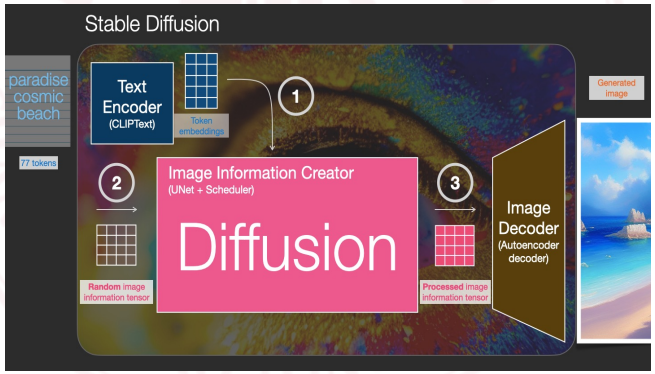


CLIP

[Image source: https://openai.com/research/clip]

ChatGPT/GPT-4

[Image source: https://www.sfgate.com/tech/article/chatgpt-openai-everyday-guide-17777804.php]



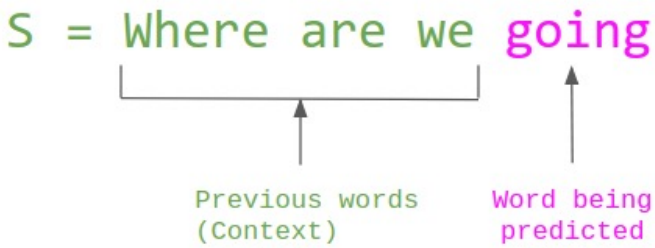
Stable Diffusion

[Image source: https://jalammar.github.io/images/stable-diffusion/stable-diffusion-diffusion-process.png]

Motivate the rapid progress in AIGC, AIGA, AIGX…

The learning goal has not converged

- Contrastive/non-contrastive learning (InfoMax)
- Language modeling (estimate densities):

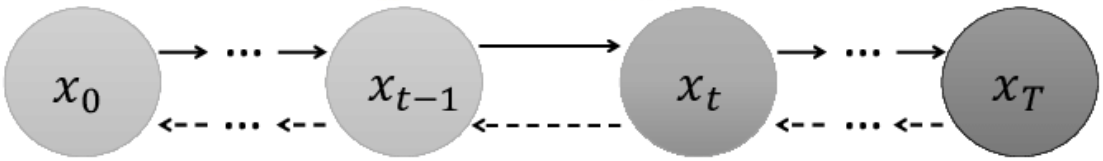


[image source: https://thegradient.pub/understanding-evaluation-metrics-for-language-models/]

```
P(S) = P(Where) x P(are | Where) x P(we | Where are) x P(going | Where are we)
```

- Score-based modeling (estimate scores, i.e., gradients of log density):

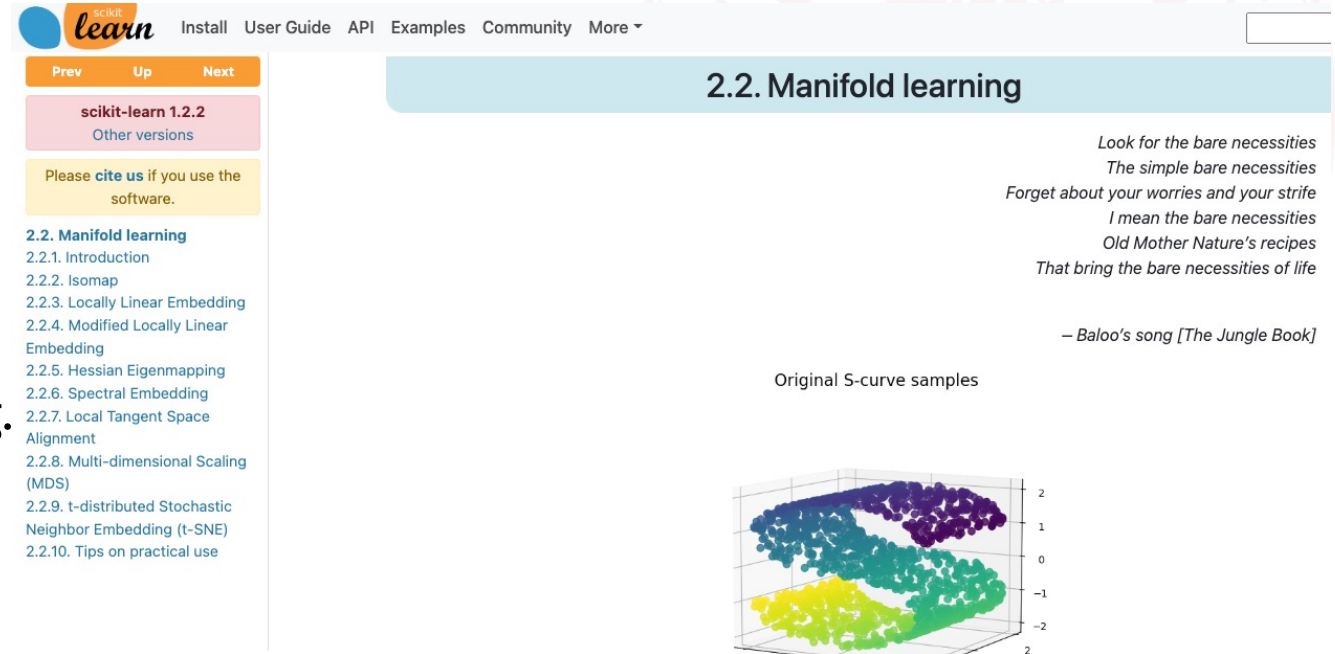
$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{\omega}_t$$



$$\mathrm{d}\mathbf{x}_t = \overbrace{\left[-\frac{1}{2}\beta(t)\mathbf{x}_t - \beta(t)\underbrace{\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t)}_{\text{“Score Function”}}\right]}^{\text{drift term}}\mathrm{d}t + \overbrace{\sqrt{\beta(t)}\,\mathrm{d}\bar{\boldsymbol{\omega}}_t}^{\text{diffusion term}}$$

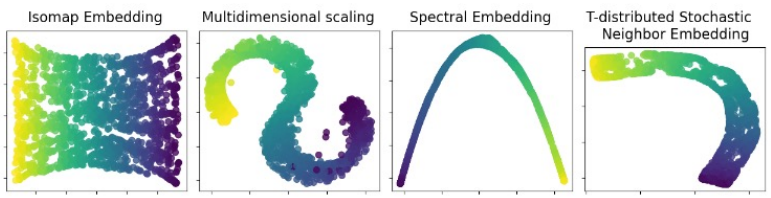
Another viable way – spectral methods (learning eigenfunctions ψ)

It is a long-standing approach in machine learning for unsupervised learning.

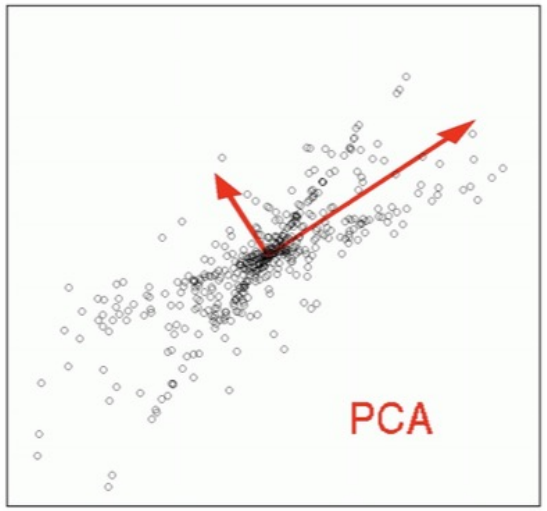


`sklearn.decomposition`.PCA

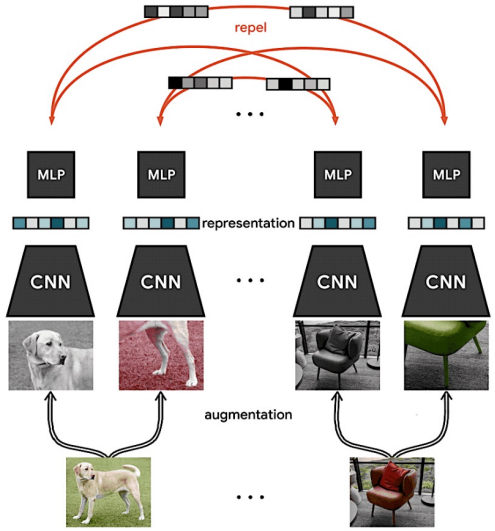
`sklearn.decomposition`.KernelPCA



Why learning eigenfunctions



v.s.



maximizing the variation of data representations

maximizing mutual information

Why learning eigenfunctions

Stein's Lemma (1972)

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

SSGE (Shi et al., ICML 2018)

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

density (score) eigenfunction

Spectral methods seem to capture more information than generative modelling

Eigenfunctions defined on the kernel integral operator

$(T_k f)(x) := \mathbb{E}_{x' \sim p}\left[k(x, x') f(x')\right]$

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

- Similar to the infinite-dim matrix eigenvalue problem:

$$Ku = \lambda u$$

Eigenfunctions defined on the kernel integral operator

$$(T_k f)(x) := \mathbb{E}_{x' \sim p}\left[k(x, x') f(x')\right]$$

$$k(x, x') \quad + \quad x_1, \ldots, x_n \sim p_{\text{(data distribution)}} \longrightarrow \psi_j(x)$$

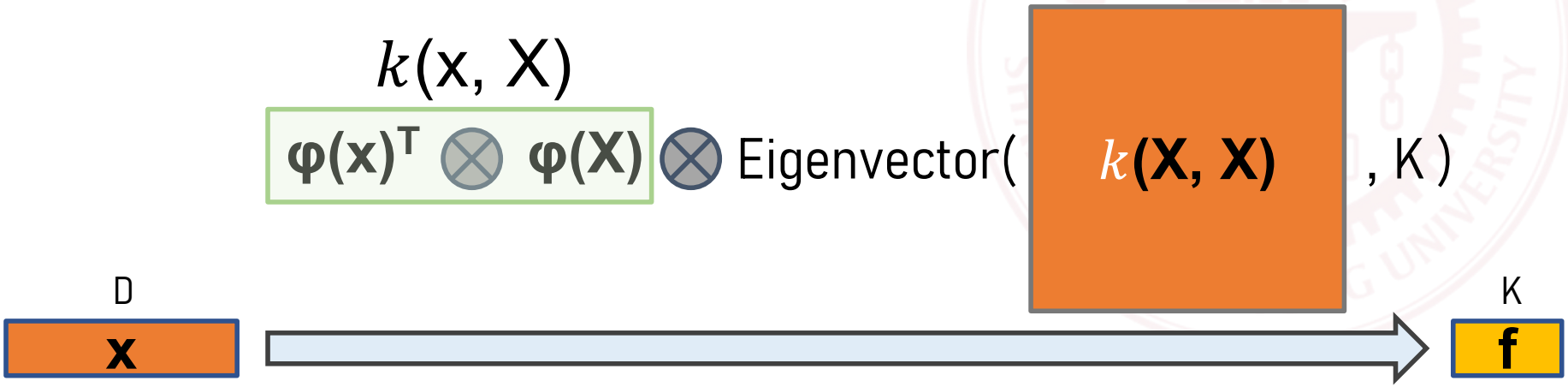
Eigenfunction

- It seems to be a good learning principle. Why less used today?

➢ Scaling is a problem for nonparametric methods

➢ Cannot leverage inductive bias such as equivariance

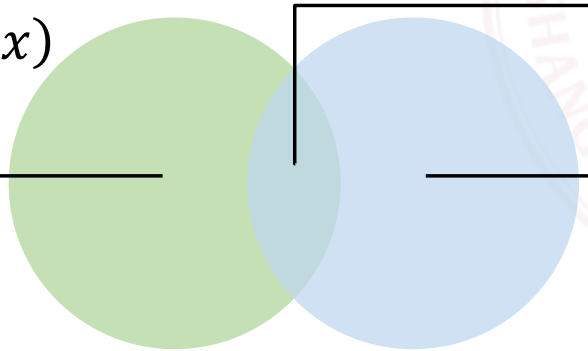
An example of the classic approach (Nystrom method)



$$\psi(x) = k(x, X)[\frac{v_1}{\sqrt{\mu_1}}, \dots, \frac{v_K}{\sqrt{\mu_K}}]$$

Spectral methods + deep learning

$$\mathbb{E}_{x' \sim p}\left[\kappa(x, x')\psi(x')\right] = \mu\psi(x)$$



Spectral methods:
learn eigenfunctions;
usually nonparametric

Deep learning:
expressive; parametric

An objective for learning neural eigenfunctions

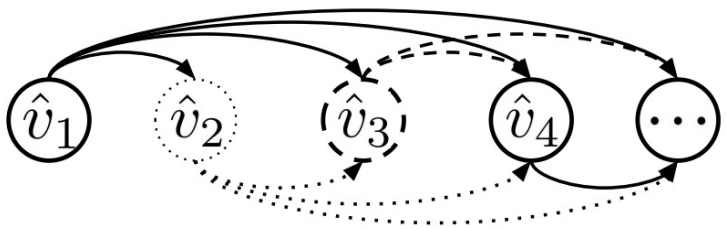
Deng, Shi & Zhu, ICML'22

L2 Batch normalization

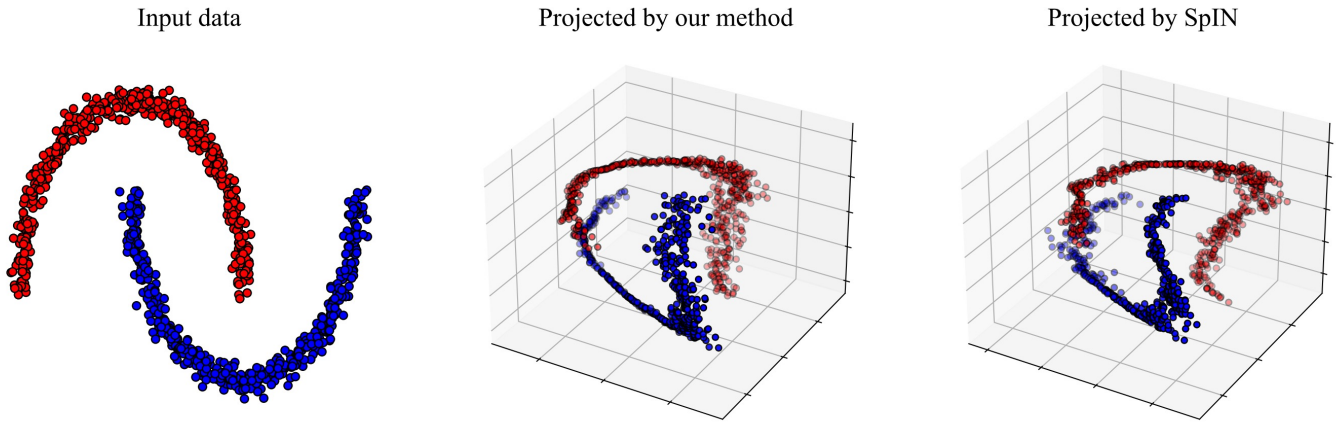
$$\max_{\psi_j} R_{jj} - \boxed{\sum_{i=1}^{j-1} \frac{R_{ij}^2}{R_{ii}}} \quad s.t. \quad \boxed{\mathbb{E}_{x' \sim p}\left[\psi_j(x')^2\right] = 1}, j = 1, \ldots, k$$

$$R_{ij} = \mathbb{E}_{x, x' \sim p}\left[\psi_i(x)\kappa(x, x')\psi_j(x')\right]$$

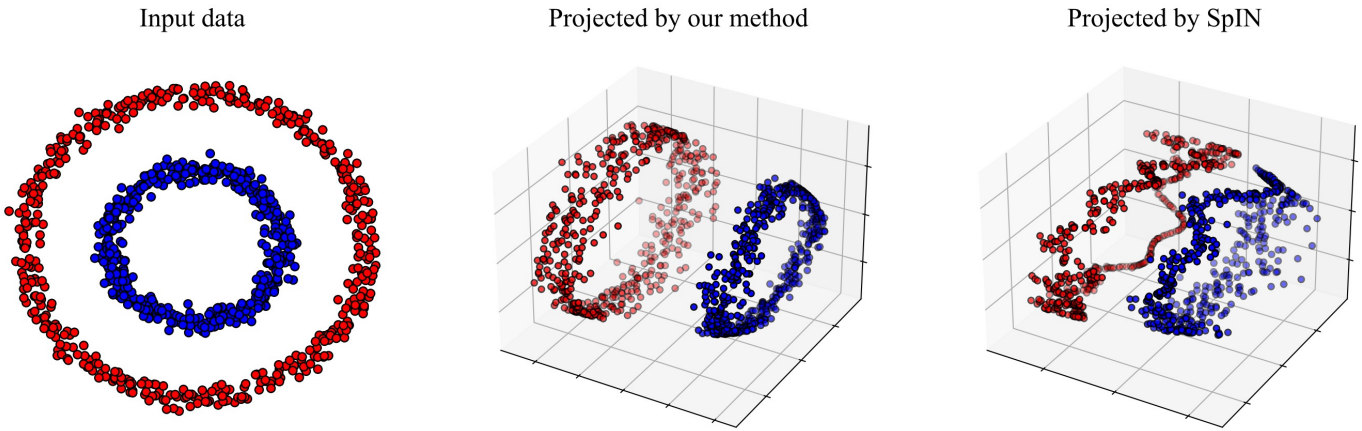
- Can be seen as a function-space generalization of EigenGame [Gemp et al., 2020] which works on PSD matrices



The neural eigenfunctions of kernels defined with random MLPs



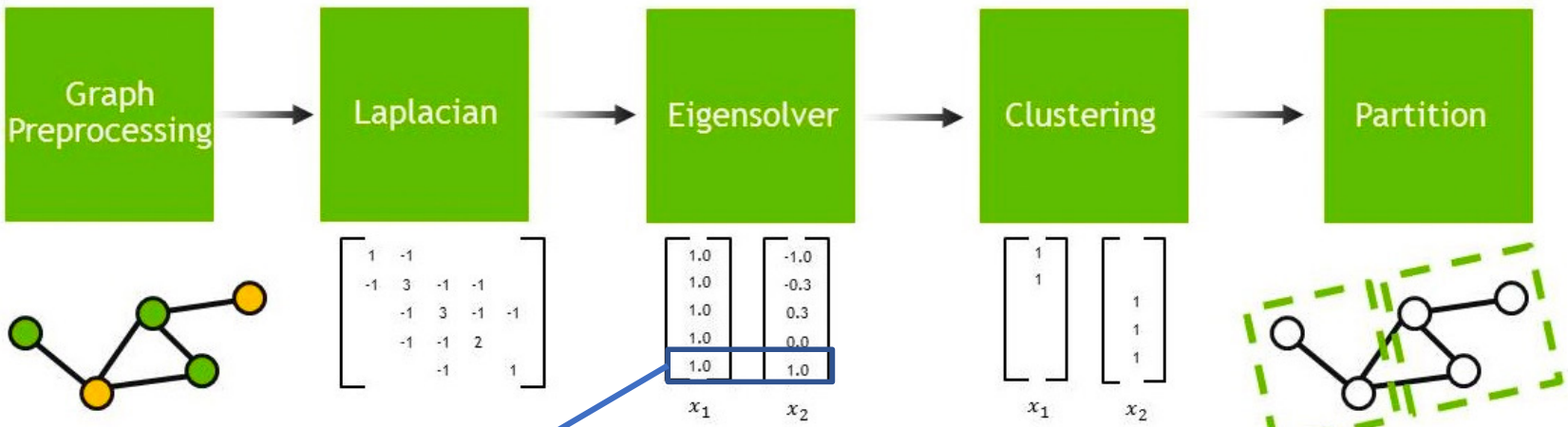
(a) "Two-moon" data



(b) "Circles" data

The spectral principle for representation learning

- Such a principle dates back to *spectral clustering* [Shi & Malik, 2000] and *Laplacian Eigenmaps* [Belkin & Niyogi, 2003]

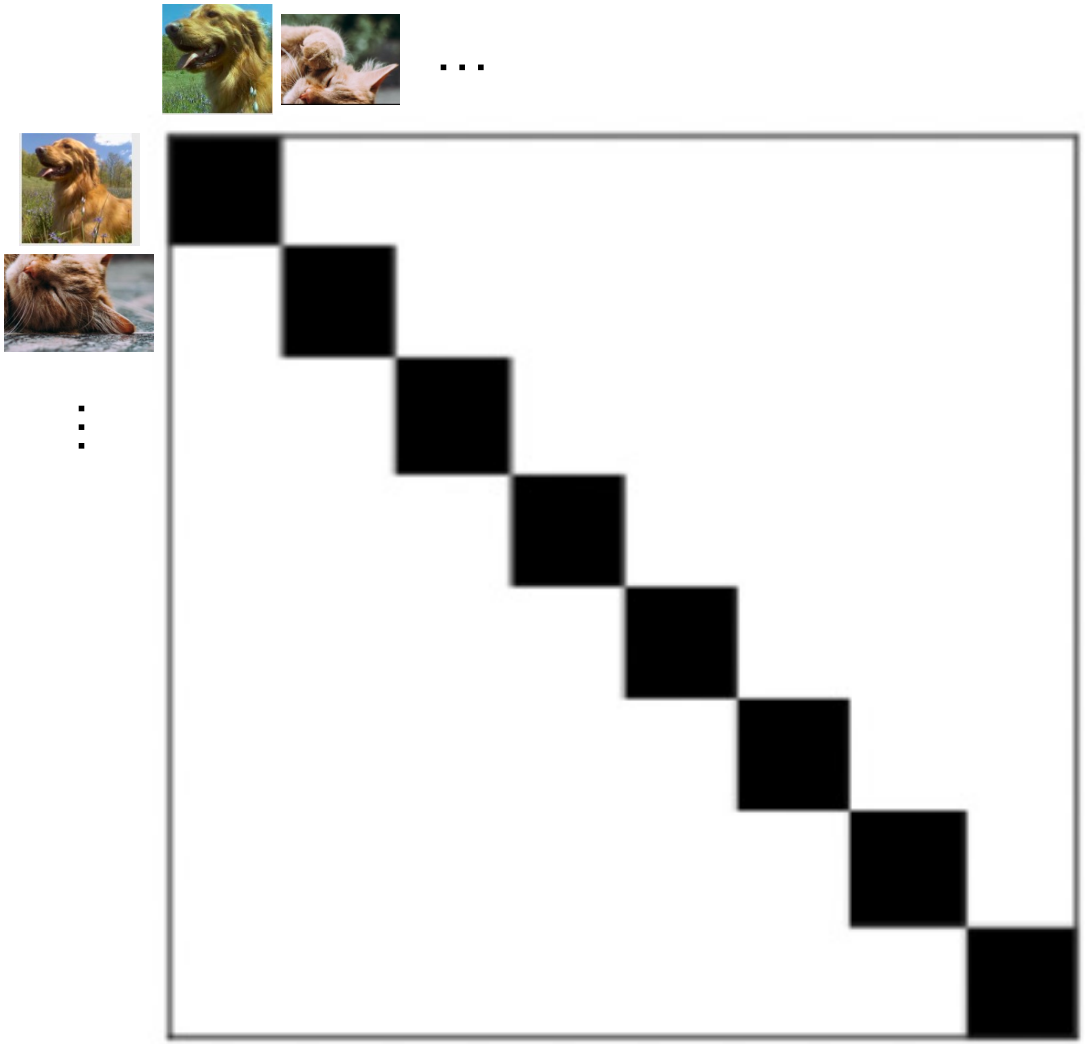


[Source: https://developer.nvidia.com/discover/cluster-analysis]

the *eigenmap* of data

- The outputs of principal eigenfunctions are representations that optimally preserve local neighborhoods on data manifolds (min-cut of a graph)

Key to generalizing this principle to domains of interest - the choice of the kernel



- The contrastive kernel
- $\kappa(x, x') = \frac{E_{p(\bar{x})}[p(x|\bar{x})p(x'|\bar{x})]}{p(x)p(x')}$

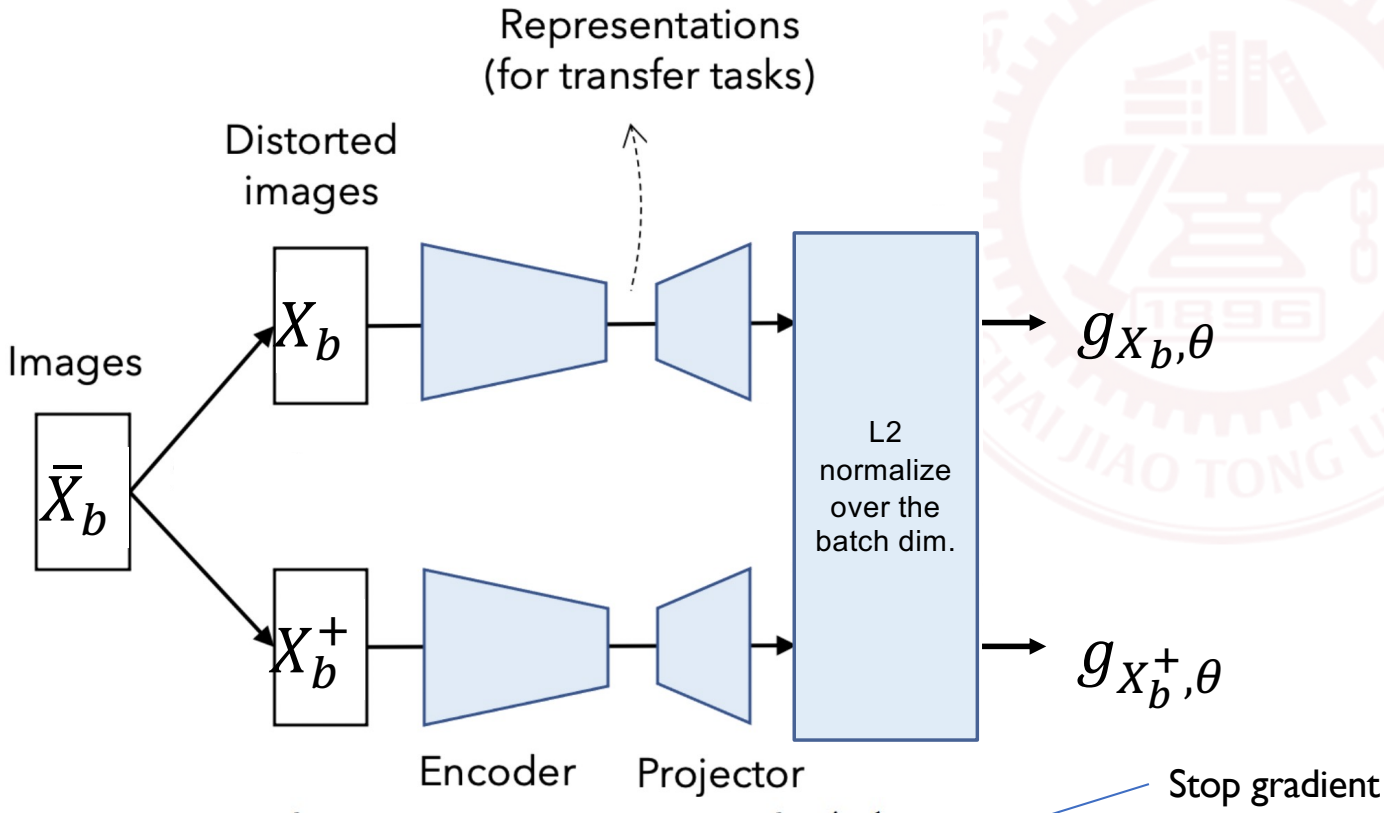
  $p(x|\bar{x})$: augmentation distribution

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

- The kernel can reflect semantic closeness

Eigenfunctions are strong self-supervised learners

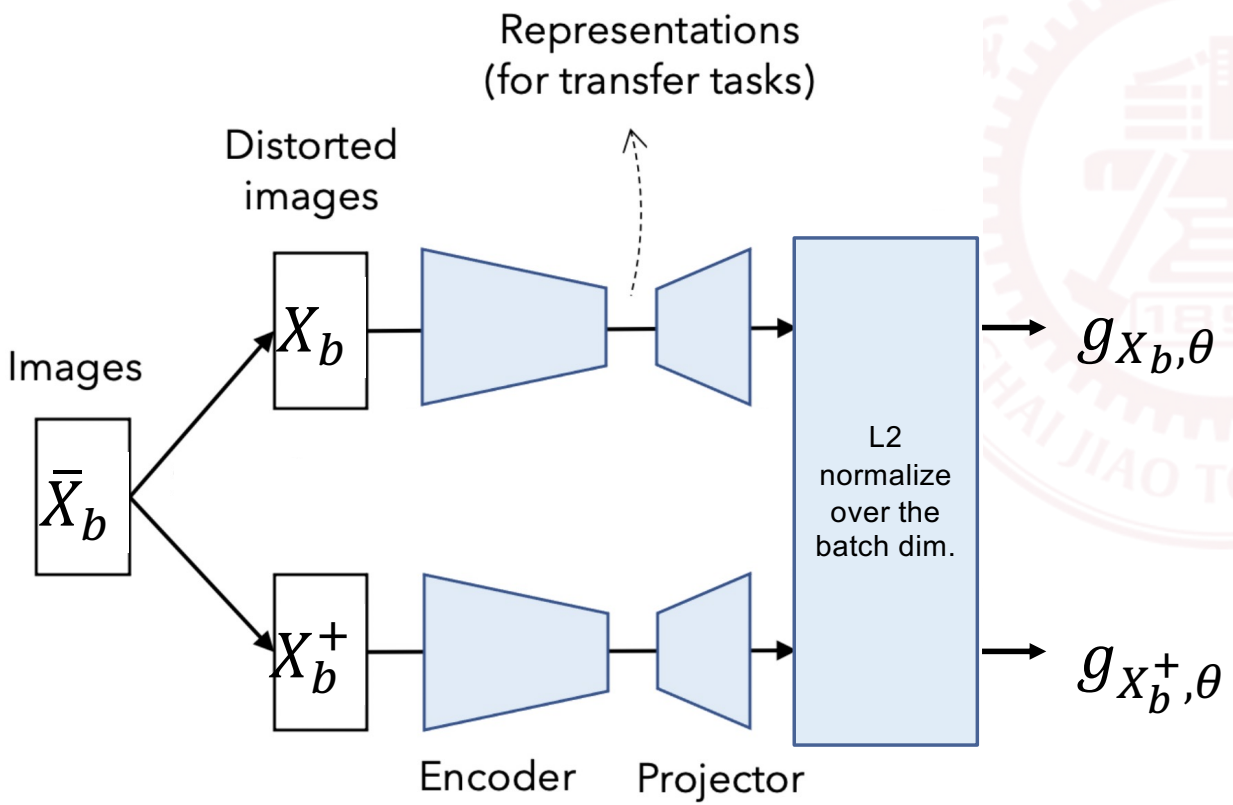
Deng*, Shi*, et al., 2022



$$\ell(\boldsymbol{\theta}) = \sum_{j=1}^{k} \left(\boldsymbol{g}_{\mathbf{X}_b,\boldsymbol{\theta}} \boldsymbol{g}_{\mathbf{X}_b^+,\boldsymbol{\theta}}^\top\right)_{j,j} - \alpha \sum_{j=1}^{k} \sum_{i=1}^{j-1} \left(\boxed{\widehat{\boldsymbol{g}_{\mathbf{X}_b,\boldsymbol{\theta}}}} \boldsymbol{g}_{\mathbf{X}_b^+,\boldsymbol{\theta}}^\top\right)_{i,j}^2$$

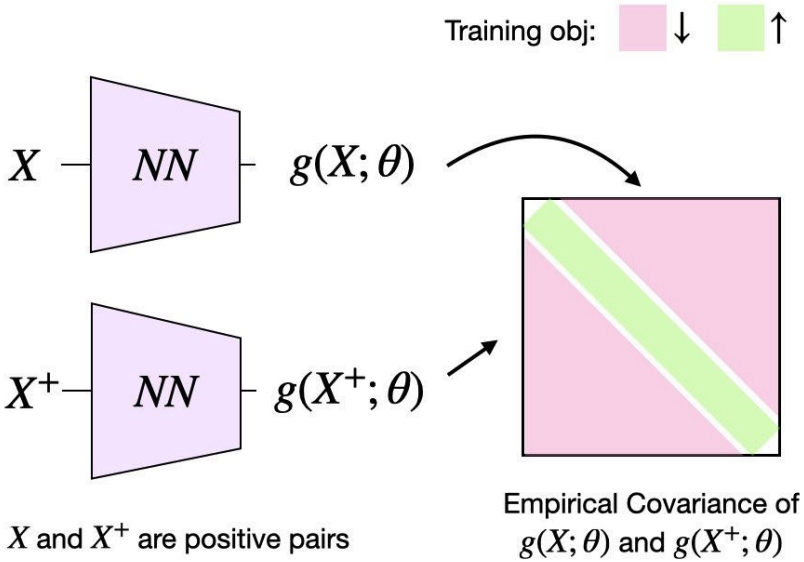
Deng*, Shi*, Zhang, Cui, Lu & Zhu. Neural Eigenfunctions Are Structured Representation Learners. *arXiv:2210.12637,* 2022.

One merit

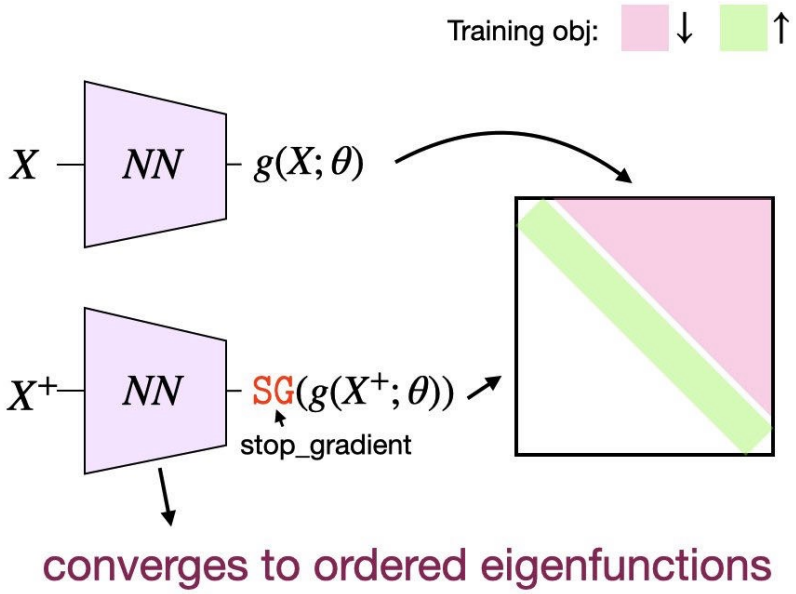


- The features are ordered by their relative importance due to the convergence to ordered eigenfunctions (principal eigenfuncs contain more critical info from kernel)
- The features are orthogonal to others in function space, so redundancy is minimized
- So we can adapt representation length according to cost-quality tradeoff

The comparison to Barlow Twins (Zbontar et al., 2021)

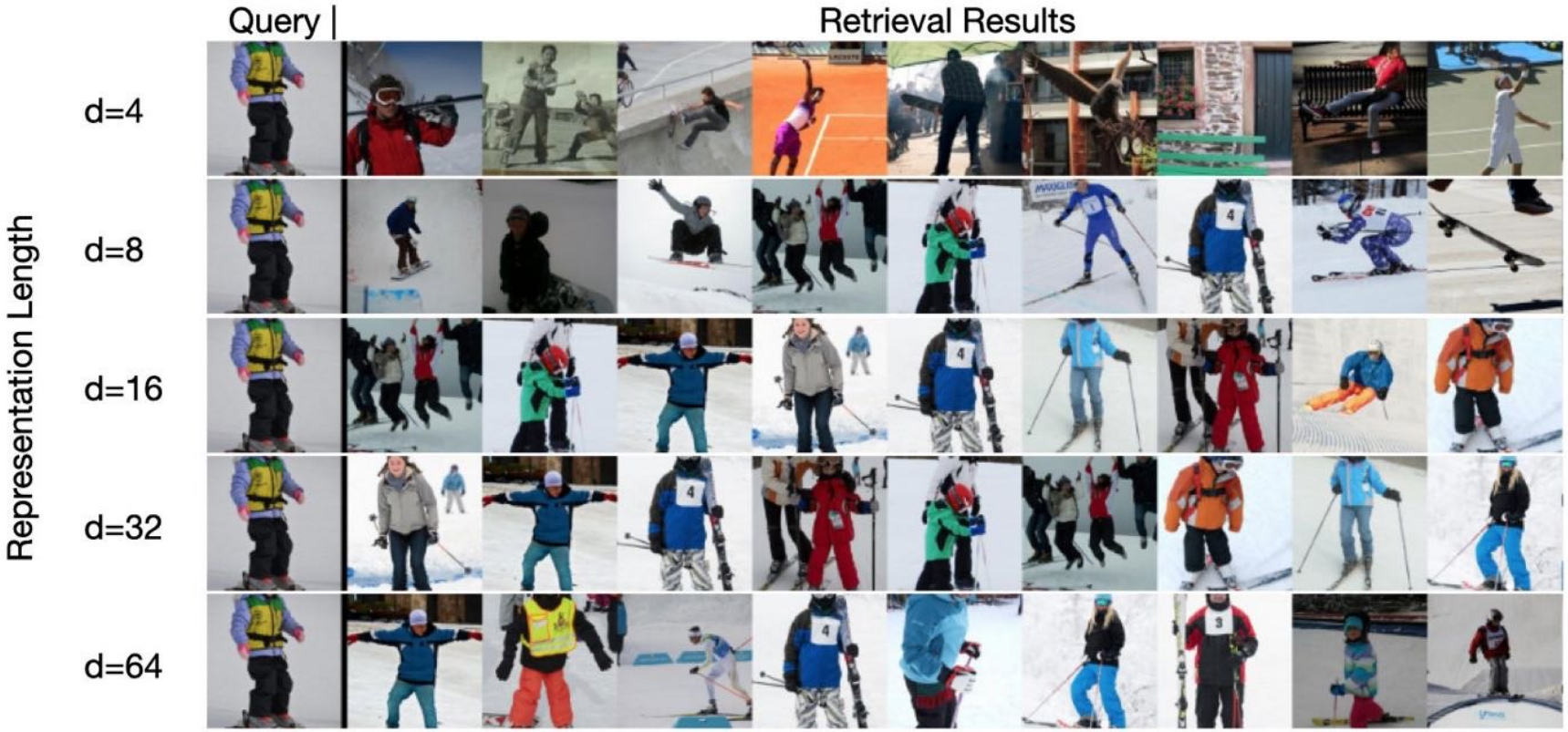


Barlow Twins



Ours

Unsupervised image retrieval at different levels of representation truncation



Unsupervised image retrieval at different levels of representation truncation

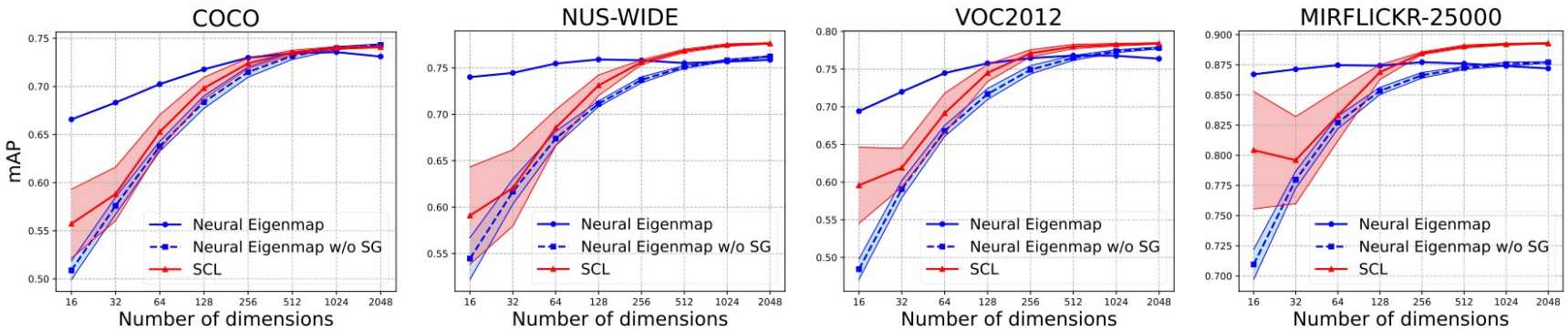


Figure 1: Retrieval mAP varies w.r.t. representation dimensionality.

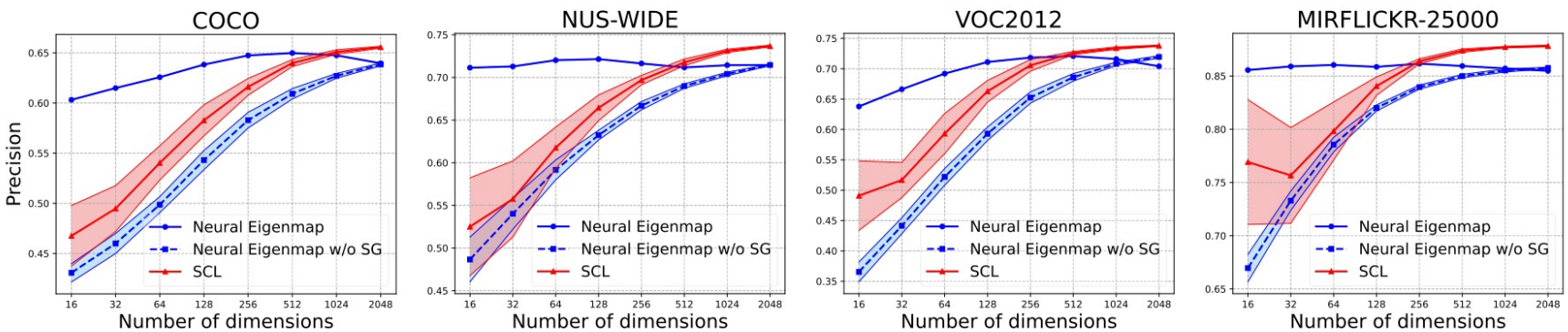
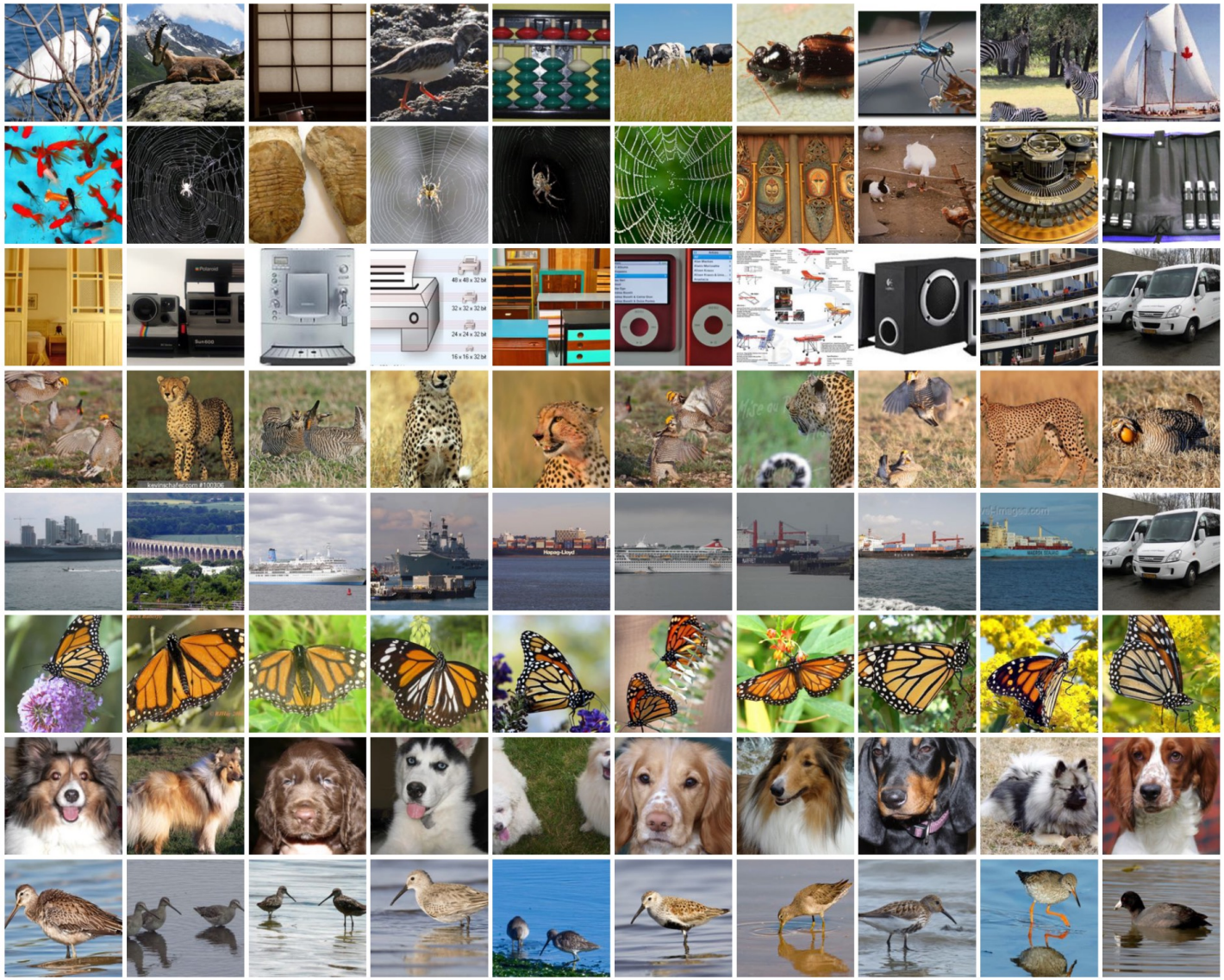


Figure 2: Retrieval precision varies w.r.t. representation dimensionality.

Neural Eigenmap requires up to **16× fewer** representation dimensions than the competitors to achieve similar retrieval performance

Images that excite the neural eigenfunctions most



ImageNet linear probe accuracy with ResNet-50 encoder

Table 1: Comparisons on ImageNet linear probe accuracy (%) with the ResNet-50 encoder pre-trained for *100 epochs*. The results of SimCLR, SwAV, MoCo v2, BYOL, and SimSiam are from (Chen & He, 2021). The result of SCL is from (HaoChen et al., 2021), and that of Barlow Twins is reproduced by ourselves. As shown, our method outperforms all baselines.

Method	batch size	top-1 accuracy
SimCLR	4096	66.5
SwAV	4096	66.5
MoCo v2	256	67.4
BYOL	4096	66.5
SimSiam	256	68.1
SCL	384	67.0
Barlow Twins	2048	66.2
Neural Eigenmap	2048	**68.4**

Comparison on ImageNet linear probe accuracy with various training epochs

Method	100 ep	200 ep	400 ep
SimSiam	68.1	70.0	70.8
Neural Eigenmap	68.4	70.3	71.5

Transfer learning on COCO detection and instance segmentation

Pre-training method	COCO detection			COCO instance seg.		
	AP_{50}	AP	AP_{75}	AP^{mask}_{50}	AP^{mask}	AP^{mask}_{75}
ImageNet supervised	58.2	38.2	41.2	54.7	33.3	35.2
SimCLR	57.7	37.9	40.9	54.6	33.3	35.3
MoCo v2	58.8	39.2	42.5	55.5	34.3	36.6
BYOL	57.8	37.9	40.9	54.3	33.2	35.0
SimSiam, base	57.5	37.9	40.9	54.2	33.2	35.2
SimSiam, optimal	59.3	39.2	42.1	56.0	34.4	36.7
Neural Eigenmap	**59.6**	**39.9**	**43.5**	**56.3**	**34.9**	**37.4**

Neural Eigenmaps for graph-structure data

- Learning eigenfunctions provides a unifying surrogate objective for representation learning

- The normalized adjacency matrix for a graph is

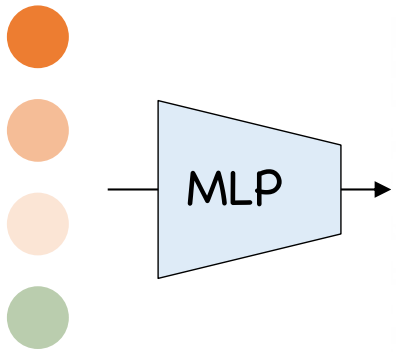
$$\bar{\mathbf{A}} := \mathbf{D}^{1/2}\mathbf{A}\mathbf{D}^{1/2}$$

- We propose to treat $\bar{A}$ as the gram matrix of $\dot{\kappa}(x, x)$ on X

- The kernel may not be positive semidefinite, so we make a fix to our theorem and show that when the kernel has at least $k - 1$ positive eigenvalues, we can still use that optimization problem to discover the k principal eigenfunctions.

Neural Eigenmaps for graph-structure data

Node features X_b



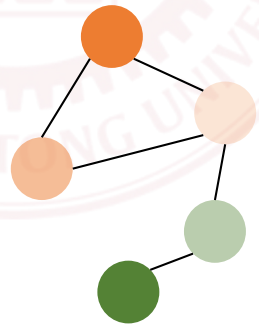
…

Neural Eigenmaps $g_{X_b,\theta}$

v_1	-4.3	0.3	0.7	0.01	0.02	…	-1.1
v_2	-1.0	0.01	0.8	-0.2	1.3	…	2.6
v_3	-4.1	-6.9	2.4	3.7	-1.2	…	7.3
v_4	3.2	1.0	0.1	0.4	-3.9	…	0.4

…

Embedded into



The learning objective $\ell(\boldsymbol{\theta}) = \sum_{j=1}^{k} \left(\boldsymbol{g}_{\mathbf{X}_b,\boldsymbol{\theta}} \bar{\mathbf{A}}_b \boldsymbol{g}_{\mathbf{X}_b,\boldsymbol{\theta}}^{\top}\right)_{j,j} - \alpha \sum_{j=1}^{k} \sum_{i=1}^{j-1} \left(\widehat{\boldsymbol{g}_{\mathbf{X}_b,\boldsymbol{\theta}}} \bar{\mathbf{A}}_b \boldsymbol{g}_{\mathbf{X}_b,\boldsymbol{\theta}}^{\top}\right)_{i,j}^{2}$

Neural Eigenmaps can also beat Laplacian Eigenmaps and GCNs!

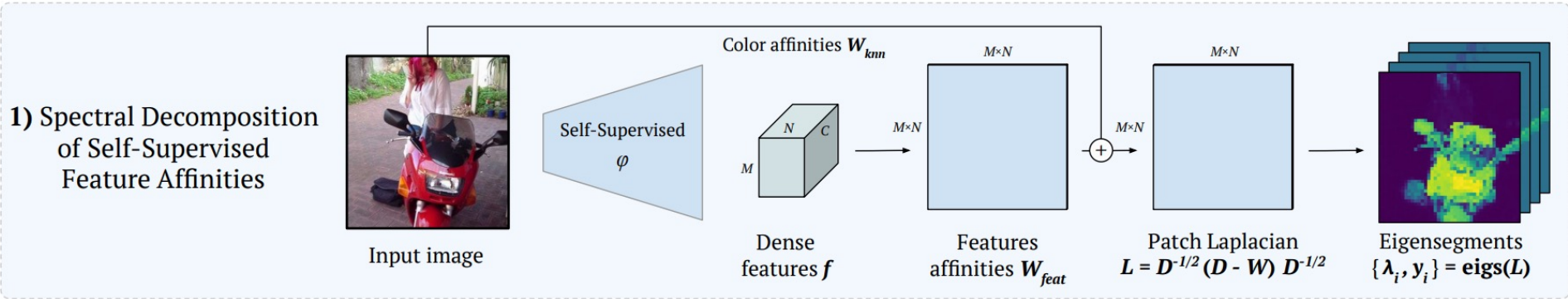
- We operate on OGBN-Products [Hu et al., 2020], one of the most large-scale node property prediction benchmarks, with **2, 449, 029** nodes and **61, 859, 140** edges

Method	100% training labels	10% training labels	1% training labels
Plain MLP	62.16 ± 0.15	57.44 ± 0.20	47.76 ± 0.62
Laplacian Eigenmap + MLP	64.21 ± 0.35	58.99 ± 0.20	49.94 ± 0.30
Node2vec + MLP	72.50 ± 0.46	68.72 ± 0.43	61.97 ± 0.44
GCN	75.72 ± 0.31	73.14 ± 0.34	67.61 ± 0.48
Neural Eigenmap	76.93 ± 0.04	74.48 ± 0.39	67.84 ± 0.79
Neural Eigenmap w/o `stop_grad`	$\mathbf{78.33} \pm 0.08$	$\mathbf{75.78} \pm 0.46$	$\mathbf{68.04} \pm 0.39$

- Our method is much faster than GCNs in the test phase because GCN needs to aggregate information from the graph while our methods doesn't

Spectral clustering for unsupervised semantic segmentation

Challenges



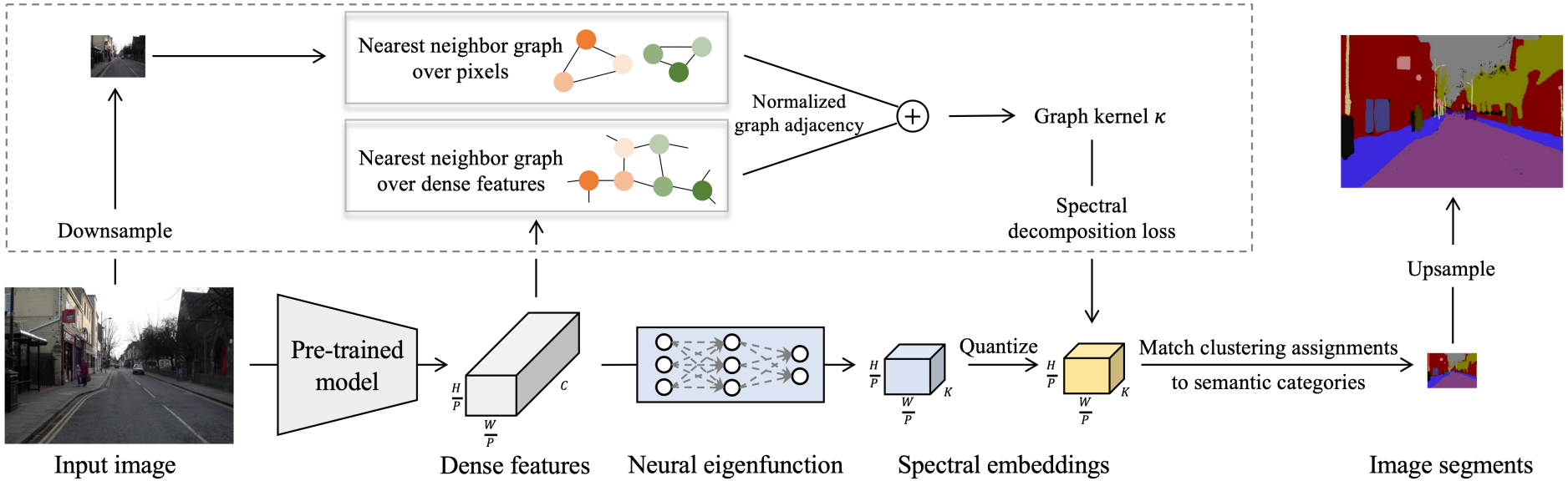
[Melas-Kyriazi et al., 2022]

- The scalability issue: computing the eigenvectors of the NP^2-by-NP^2 matrix over a large dataset is intractable (P^2 is the number of patches in a picture)

- The inference still involves matrix decomposition

Unsupervised semantic segmentation by learning eigenfunctions

Deng & Luo, ICCV 2023



Unsupervised semantic segmentation by learning eigenfunctions

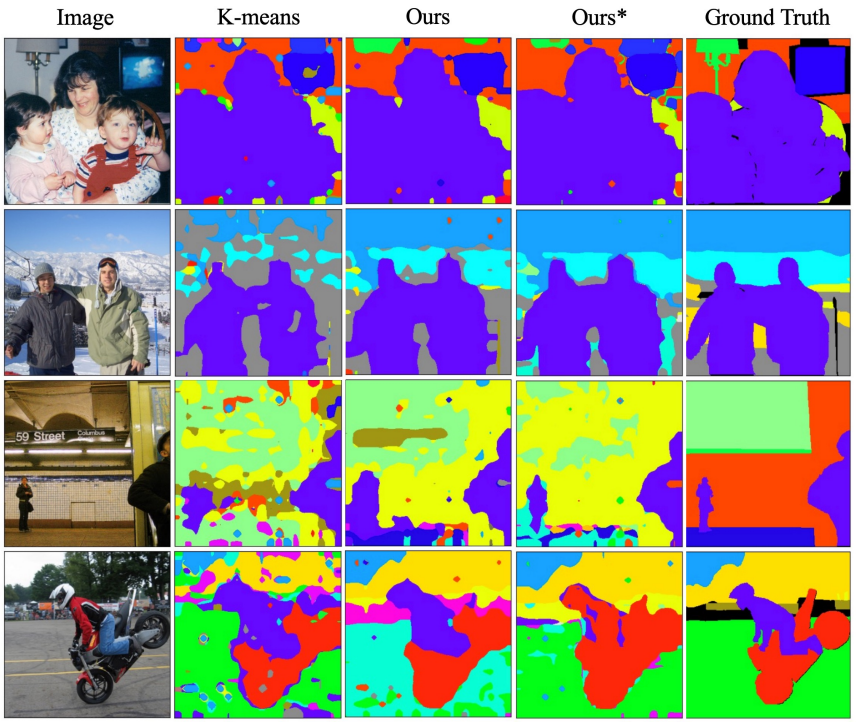


Figure 2. Visualization of the unsupervised semantic segmentation results on Pascal Context [31].

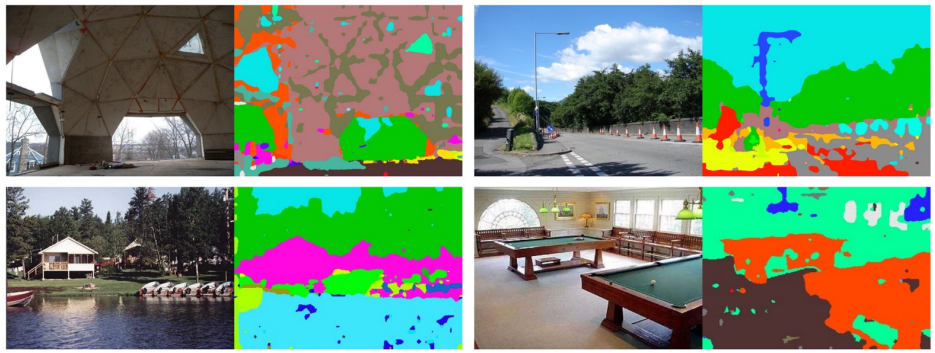
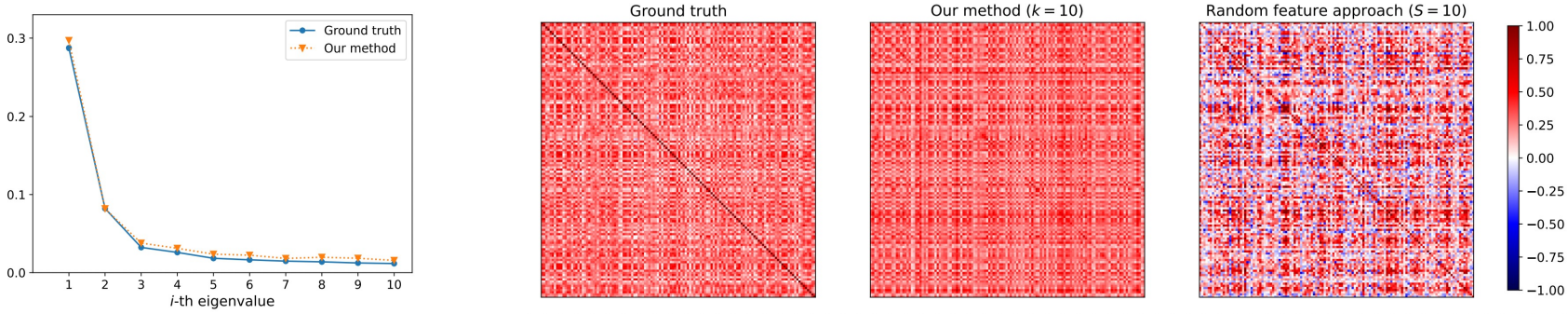


Figure 3. Visualization of the unsupervised semantic segmentation results of our method on ADE20K. In each pair, the left refers to the input image, and the right refers to the segmentation result. As shown, our method can yield reasonable pixel groups for images containing complex structures.

Scaling up Neural Tangent Kernels (NTKs)

Mercer's theorem: $\kappa(x, x') = \sum_{j \geq 1} \mu_j \psi_j(x) \psi_j(x')$



Approximating NTKs of ResNet-20

- NTKs are powerful kernels and important tools for understanding deep learning
- Scaling NTKs has been painful: 1K random features = 1K forward/backward passes
- Replace random features with neural eigenfunctions!

Accelerate Laplace approximation with approximated NTK

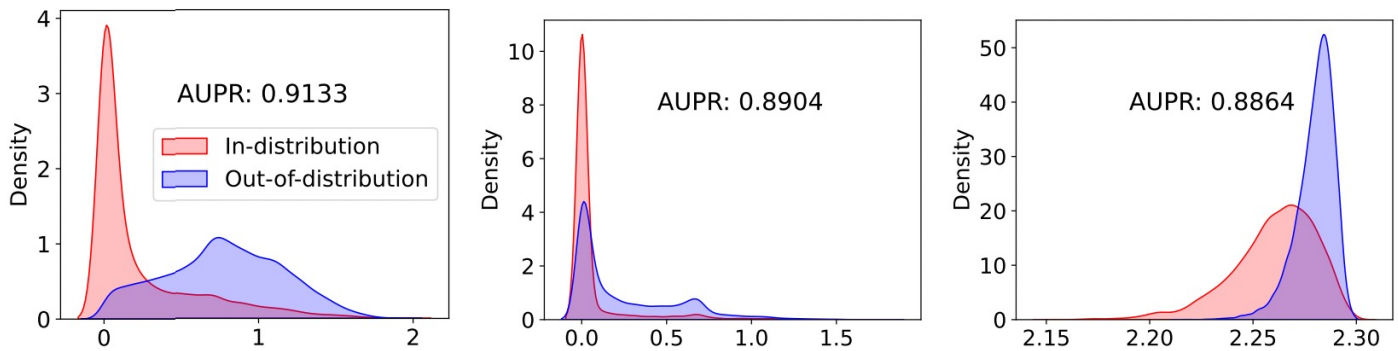
Deng, Zhou & Zhu, NeurIPS'22

- The functional predictive for linearized Laplace approximation:

$$\kappa_{\mathrm{LLA}}(\boldsymbol{x},\boldsymbol{x}') = \sigma_0^2\Big(\kappa_{\mathrm{NTK}}(\boldsymbol{x},\boldsymbol{x}') - \kappa_{\mathrm{NTK}}(\boldsymbol{x},\mathbf{X})[\boldsymbol{\Lambda}_{\mathbf{X},\mathbf{Y}}^{-1}/\sigma_0^2 + \kappa_{\mathrm{NTK}}(\mathbf{X},\mathbf{X})]^{-1}\kappa_{\mathrm{NTK}}(\mathbf{X},\boldsymbol{x}')\Big)$$

- Introduce NeuralEF to approximate the NTK:

$$\begin{aligned}\kappa_{\mathrm{LLA}}(\boldsymbol{x},\boldsymbol{x}') \approx& \sigma_0^2\Big(\varphi(\boldsymbol{x})\varphi(\boldsymbol{x}')^\top - \varphi(\boldsymbol{x})\boldsymbol{\varphi}_{\mathbf{X}}^\top\left[\boldsymbol{\Lambda}_{\mathbf{X},\mathbf{Y}}^{-1}/\sigma_0^2 + \boldsymbol{\varphi}_{\mathbf{X}}\boldsymbol{\varphi}_{\mathbf{X}}^\top\right]^{-1}\boldsymbol{\varphi}_{\mathbf{X}}\varphi(\boldsymbol{x}')^\top\Big)\\ =& \varphi(\boldsymbol{x})\underbrace{\Big[\sum_i \varphi(\boldsymbol{x}_i)^\top\Lambda(\boldsymbol{x}_i,\boldsymbol{y}_i)\varphi(\boldsymbol{x}_i) + \mathbf{I}_K/\sigma_0^2\Big]}_{\mathbf{G}}{}^{-1}\varphi(\boldsymbol{x}')^\top \triangleq \kappa_{\mathrm{ELLA}}(\boldsymbol{x},\boldsymbol{x}')\end{aligned}$$



(a) Ours (b) MAP (c) KFAC LLA

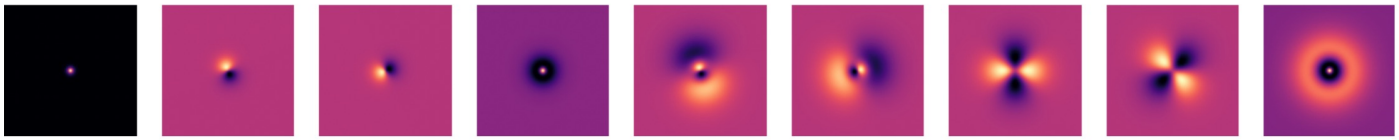
Deng, Zhou & Zhu. Accelerated Linearized Laplace Approximation for Bayesian Deep Learning. *NeurIPS* 2022.

Solve PDEs by eigendecomposition

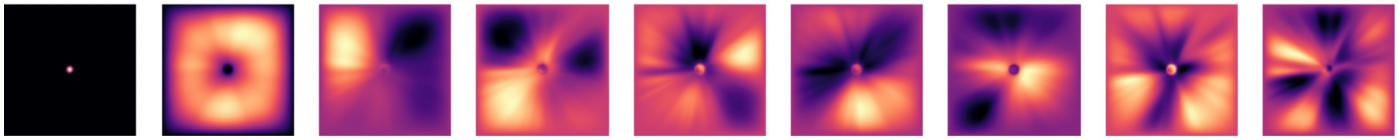
- The time-independent Schrodinger equation for a single particle with mass m in a potential field $\mathrm{V}(\mathrm{x})$ is a PDE of the form:

$$E\psi(\mathbf{x}) = \frac{-\hbar^2}{2m}\nabla^2\psi(\mathbf{x}) + V(\mathbf{x})\psi(\mathbf{x}) = \mathcal{H}[\psi](\mathbf{x})$$

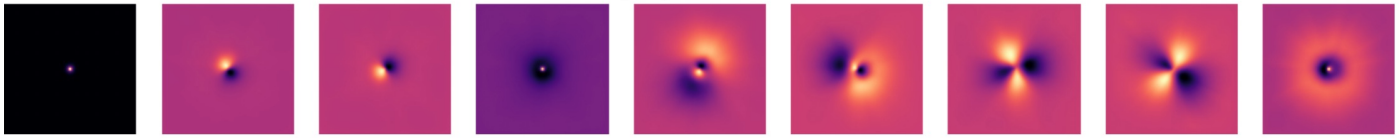
whose solutions describe the wavefunctions $\psi(x)$ with unique energy E



(a) Eigenvectors found by exact eigensolver on a grid



(b) Eigenfunctions found by SpIN without bias correction ($\beta = 1$)



(c) Eigenfunctions found by SpIN with $\beta = 0.01$ to correct for biased gradients

David Pfau et al., Spectral Inference Networks: Unifying Deep and Spectral Learning. ICLR'19

Learn the solving operator for PDEs

$$(T_k f)(x) := \mathbb{E}_{x' \sim p}\left[k(x, x') f(x')\right]$$

- The current approach: given T_k, estimate μ, ψ

- A new problem: given $T_k f_i = u_i, i = 1, \ldots, N$, estimate μ, ψ

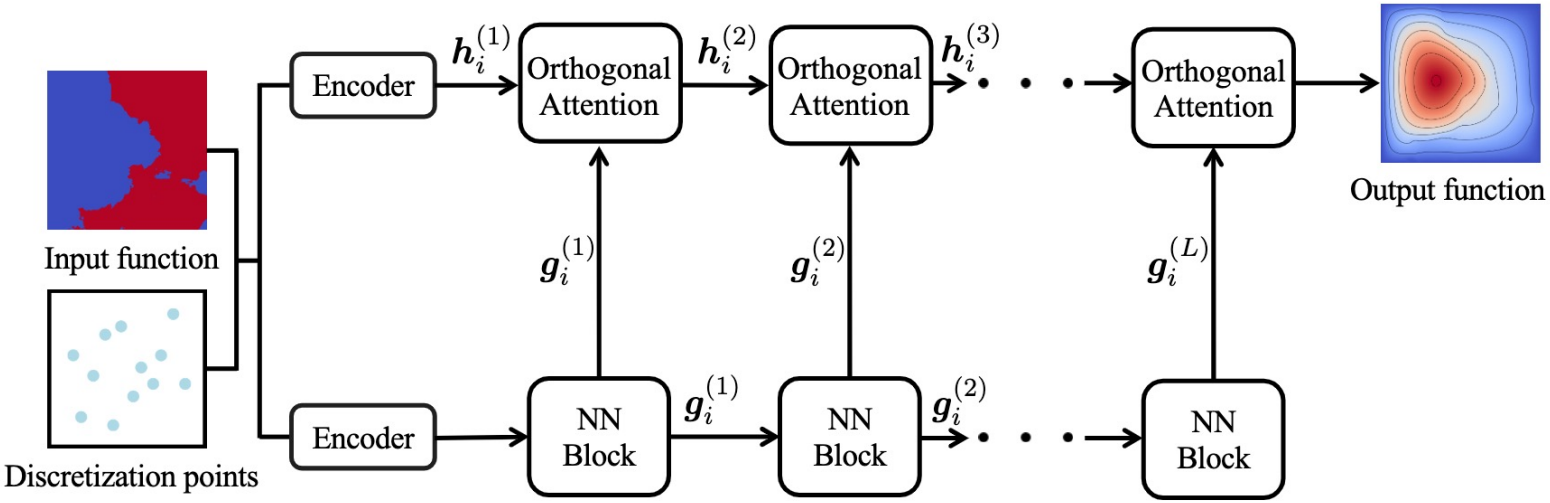
✓ Can recover the kernel integral operator from μ, ψ

✓ Corresponds to the Green's function method for solving PDE

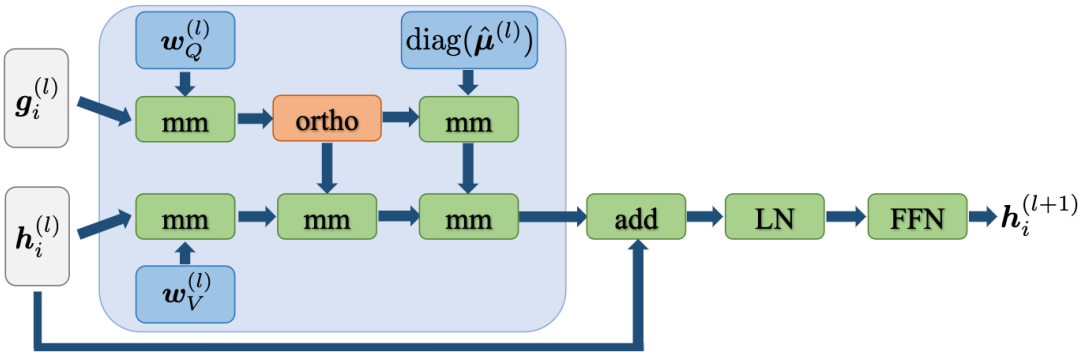
Learn the solving operator for PDEs

Xiao, Hao, Lin, Deng* & Su*, 2023

- Orthogonal neural operator



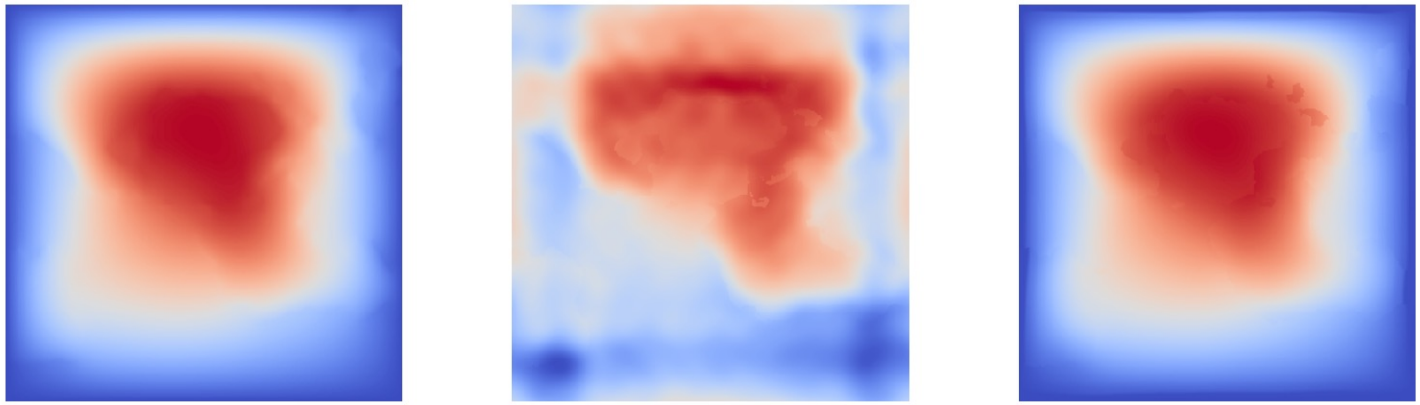
- Orthogonal attention



Learn the solving operator for PDEs

Xiao, Hao, Lin, Deng* & Su*, 2023

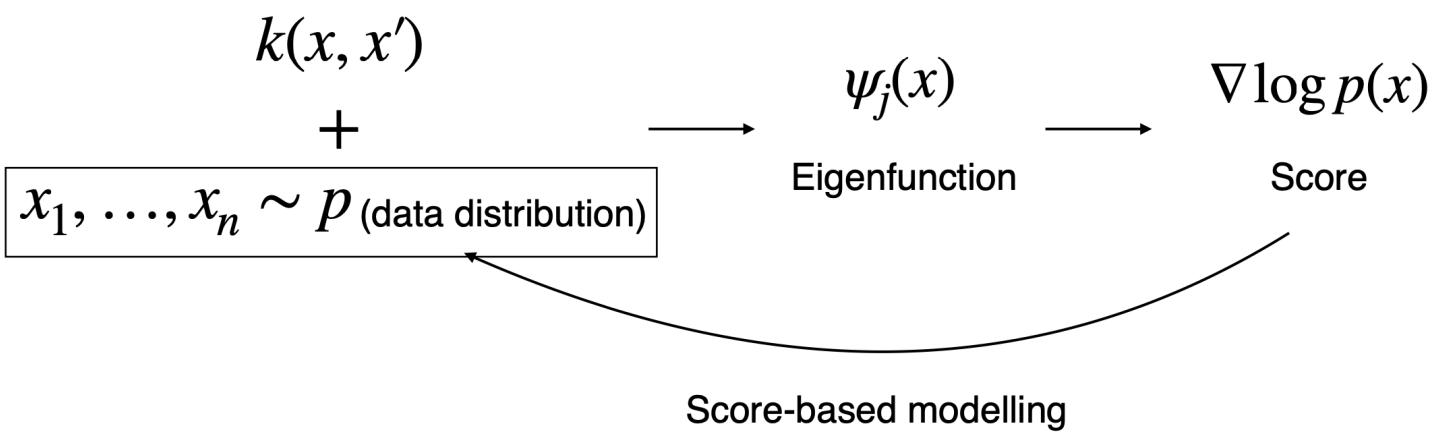
- Improved generalization ability



(a) (b) (c)

Figure 3: Zero-shot super-resolution results on Darcy. (a): Groundtruth. (b): Prediction of FNO. (c): Prediction of ONO. Trained on 43×43 data and evaluated on 421×421.

Future direction: the integration of neural eigenfunctions and score-based models



Will generative modelling and representation learning eventually converge to a single method?

Takeaways

- Spectral methods can lead to a framework of unsupervised learning

- Replacing nonparametric methods with a deep functional representation is fruitful.

Thanks!

References

NeuralEF: Deconstructing Kernels by Deep Neural Networks
Zhijie Deng, Jiaxin Shi, and Jun Zhu
International Conference on Machine Learning **(ICML)**, 2022

Accelerated Linearized Laplace Approximation for Bayesian Deep Learning
Zhijie Deng, Feng Zhou, and Jun Zhu
Advances in Neural Information Processing Systems **(NeurIPS)**, 2022

Neural Eigenfunctions Are Structured Representation Learners
Zhijie Deng*, Jiaxin Shi*, Hao Zhang, Peng Cui, Cewu Lu, and Jun Zhu

Learning Neural Eigenfunctions for Unsupervised Semantic Segmentation
Zhijie Deng and Yucen Luo
International Conference on Computer Vision **(ICCV)**, Paris, France, 2023

Improved Operator Learning by Orthogonal Attention
Zipeng Xiao, Zhongkai Hao, Bokai Lin, **Zhijie Deng***, and Hang Su*