NeuralSVD : Operator SVD with Neural Networks via Nested Low-Rank Approximation

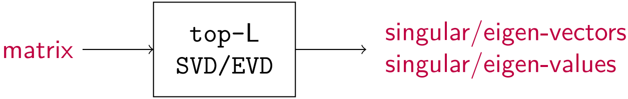
Jongha (Jon) Ryu, MIT EECS

ITA 2024

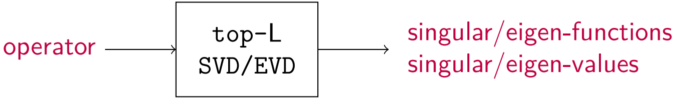
Joint work with

Xiangxiang Xu, Mellihcan Erol, Yuheng Bu,
Lizhong Zheng, and Gregory Wornell

Spectral Decomposition is Extremely Versatile



Spectral Decomposition is Extremely Versatile



Spectral Decomposition is Extremely Versatile

1. Representation learning (low-dimensional embedding of data)

operator $\longrightarrow$ `top-L SVD/EVD` $\longrightarrow$ singular/eigen-functions, singular/eigen-values

Spectral Decomposition is Extremely Versatile

1. Representation learning (low-dimensional embedding of data)
   PCA,

$$\underbrace{\hat{\Sigma} \triangleq \mathbb{E}_{\hat{p}(\mathbf{x})}[\mathbf{x}\mathbf{x}^{\mathsf{T}}]}_{\substack{\text{empirical covariance} \\ \in \mathbb{R}^{D\times D}}} \longrightarrow \boxed{\begin{array}{c}\texttt{top-L} \\ \texttt{EVD}\end{array}} \longrightarrow \sum_{i=1}^{L} \lambda_i \mathbf{u}_i \mathbf{u}_i^{\mathsf{T}}$$

Spectral Decomposition is Extremely Versatile

1. Representation learning (low-dimensional embedding of data)
   PCA, **Kernel PCA**,

$$\underbrace{k(\mathbf{x}, \mathbf{x}')}_{\text{kernel}} \longrightarrow \boxed{\begin{array}{c}\texttt{top-L}\\ \texttt{EVD}\end{array}} \longrightarrow \sum_{i=1}^{L} \lambda_i \phi_i(\mathbf{x}) \phi_i(\mathbf{x}')$$

Spectral Decomposition is Extremely Versatile

1. Representation learning (low-dimensional embedding of data)
   PCA, **Kernel PCA**,

$$\underbrace{K \triangleq \big[k(\mathbf{x}_i, \mathbf{x}_j)\big]}_{\substack{\text{kernel matrix}\\ \in \mathbb{R}^{N\times N}}} \longrightarrow \boxed{\begin{array}{c}\texttt{top-L}\\ \texttt{EVD}\end{array}} \longrightarrow \sum_{i=1}^{L} \hat{\lambda}_i \hat{\phi}_i \hat{\phi}_i^{\intercal}$$

Spectral Decomposition is Extremely Versatile

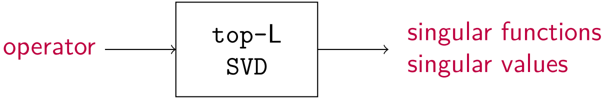
1. Representation learning (low-dimensional embedding of data)
   PCA, **Kernel PCA**, **Laplacian eigenmaps**, ...

$$\underbrace{L \triangleq I - D^{-1/2} A D^{-1/2}}_{\substack{\text{graph Laplacian} \\ \in \mathbb{R}^{N\times N}}} \longrightarrow \boxed{\begin{array}{c}\texttt{top-L} \\ \texttt{EVD}\end{array}} \longrightarrow \sum_{i=1}^{L} \hat{\lambda}_i \hat{\phi}_i \hat{\phi}_i^{\intercal}$$

Spectral Decomposition is Extremely Versatile

1. Representation learning (low-dimensional embedding of data)
   PCA, **Kernel PCA**, **Laplacian eigenmaps**, ...

2. Representation learning for two variables (coembedding)



Spectral Decomposition is Extremely Versatile

1. Representation learning (low-dimensional embedding of data)
   PCA, **Kernel PCA**, **Laplacian eigenmaps**, ...

2. Representation learning for two variables (coembedding)
   CCA,

$$\underbrace{\hat{\Sigma}_{\mathbf{x}}^{-1/2}\hat{\Sigma}_{\mathbf{xy}}\hat{\Sigma}_{\mathbf{y}}^{-1/2}}_{\substack{\text{cross correlation}\\ \in\mathbb{R}^{D_{\mathbf{x}}\times D_{\mathbf{y}}}}} \longrightarrow \boxed{\begin{array}{c}\texttt{top-L}\\ \texttt{SVD}\end{array}} \longrightarrow \sum_{i=1}^{L}\sigma_i \mathbf{u}_i \mathbf{v}_i^{\mathsf{T}}$$

Spectral Decomposition is Extremely Versatile

1. Representation learning (low-dimensional embedding of data)
   PCA, **Kernel PCA**, **Laplacian eigenmaps**, ...

2. Representation learning for two variables (coembedding)
   CCA, **HGR maximal correlation**, ...

$$\underbrace{\frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{x})p(\mathbf{y})}-1}_{\text{canonical dependence kernel}} \longrightarrow \boxed{\begin{array}{c}\texttt{top-L}\\ \texttt{EVD}\end{array}} \longrightarrow \sum_{i=1}^{L}\sigma_i\phi_i(\mathbf{x})\psi_i(\mathbf{y})$$

Spectral Decomposition is Extremely Versatile

1. Representation learning (low-dimensional embedding of data)
   PCA, **Kernel PCA**, **Laplacian eigenmaps**, ...

2. Representation learning for two variables (coembedding)
   CCA, **HGR maximal correlation**, ...

3. Solving PDEs (e.g., **quantum chemistry**)

$$\underbrace{\mathcal{H} \triangleq \mathcal{T} + \mathcal{V}}_{\text{Hamiltonian}} \longrightarrow \boxed{\begin{array}{c}\texttt{top-L}\\ \texttt{SVD/EVD}\end{array}} \longrightarrow \sum_{i=1}^{L} \lambda_i |\phi_i\rangle\langle\phi_i|$$

Spectral Decomposition is Extremely Versatile

1. Representation learning (low-dimensional embedding of data)
   PCA, **Kernel PCA**, **Laplacian eigenmaps**, ...
2. Representation learning for two variables (coembedding)
   CCA, **HGR maximal correlation**, ...
3. Solving PDEs (e.g., **quantum chemistry**)

$$\text{operator} \longrightarrow \boxed{\begin{array}{c}\texttt{top-L}\\ \texttt{SVD/EVD}\end{array}} \longrightarrow \begin{array}{l}\text{singular-/eigen-functions}\\ \text{singular-/eigen-values}\end{array}$$

Spectral Decomposition is Extremely Versatile

1. Representation learning (low-dimensional embedding of data)
   PCA, **Kernel PCA**, **Laplacian eigenmaps**, ...
2. Representation learning for two variables (coembedding)
   CCA, **HGR maximal correlation**, ...
3. Solving PDEs (e.g., **quantum chemistry**)

operator $\longrightarrow$ [top-L SVD/EVD] $\longrightarrow$ singular-/eigen-functions, singular-/eigen-values

Spectral Decomposition is Extremely Versatile

1. Representation learning (low-dimensional embedding of data)
   PCA, **Kernel PCA**, **Laplacian eigenmaps**, ...

2. Representation learning for two variables (coembedding)
   CCA, **HGR maximal correlation**, ...

3. Solving PDEs (e.g., **quantum chemistry**)

operator $\longrightarrow$ `top-L SVD/EVD` $\longrightarrow$ singular-/eigen-functions, singular-/eigen-values

- **The standard approach**: decompose LARGE matrix 🤔???

🤯 **Example**: With N electrons, $\mathbf{r} \in \mathbb{R}^{3N} \to O(\epsilon^{-3N})$ samples

Spectral Decomposition is Extremely Versatile

1. Representation learning (low-dimensional embedding of data)
   PCA, **Kernel PCA**, **Laplacian eigenmaps**, ...

2. Representation learning for two variables (coembedding)
   CCA, **HGR maximal correlation**, ...

3. Solving PDEs (e.g., **quantum chemistry**)

operator → `top-L SVD/EVD` → singular-/eigen-functions
singular-/eigen-values

- **The standard approach**: decompose LARGE matrix ???

Example: With N electrons, $\mathbf{r} \in \mathbb{R}^{3N} \to O(\epsilon^{-3N})$ samples

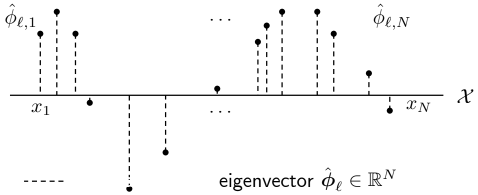
Is there an alternative to this matrix approach?

Nonparametric Approach

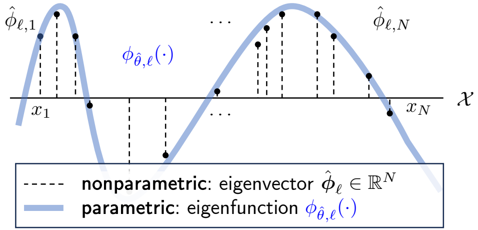
$\hat{\phi}_{\ell,1}$				$\cdots$	$\cdots$				$\hat{\phi}_{\ell,N}$

eigenvector $\hat{\boldsymbol{\phi}}_\ell \in \mathbb{R}^N$

Nonparametric Approach



~~Non~~parametric Approach



$$\phi(\cdot) \approx [\phi(x_1), \ldots, \phi(x_N)]^\intercal$$

Parametric Approach

Matrix SVD

$$T = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$$

where $\mathbf{u}_i^T \mathbf{u}_j = \mathbf{v}_i^T \mathbf{v}_j = \delta_{ij}$, $\sigma_1 \geq \sigma_2 \geq \ldots \geq 0$

Parametric Approach

Operator SVD

$$\mathcal{T} = \sum_{i=1}^{\infty} \sigma_i |\phi_i\rangle\langle\psi_i|$$

where $\langle\phi_i|\phi_j\rangle = \langle\psi_i|\psi_j\rangle = \delta_{ij}$, $\sigma_1 \geq \sigma_2 \geq \ldots \geq 0$

Operator SVD

$$\mathcal{T} = \sum_{i=1}^{\infty} \sigma_i |\phi_i\rangle\langle\psi_i|$$

where $\langle\phi_i|\phi_j\rangle = \langle\psi_i|\psi_j\rangle = \delta_{ij}$, $\sigma_1 \geq \sigma_2 \geq \ldots \geq 0$

- Hilbert space $\mathcal{F} := \mathscr{L}^2_\mu(\mathcal{X}) := \{f\colon \mathcal{X} \to \mathbb{R} \mid \|f\|^2 < \infty\}$ with inner product

$$\langle f_1|f_2\rangle := \int_{\mathcal{X}} f_1(x) f_2(x) \mu(dx) \approx \frac{1}{N} \sum_{n=1}^{N} f_1(x_n) f_2(x_n)$$

Parametric Approach

Operator SVD

$$\mathcal{T} = \sum_{i=1}^{\infty} \sigma_i |\phi_i\rangle\langle\psi_i|$$

where $\langle\phi_i|\phi_j\rangle = \langle\psi_i|\psi_j\rangle = \delta_{ij}$, $\sigma_1 \geq \sigma_2 \geq \ldots \geq 0$

- Hilbert space $\mathcal{F} := \mathscr{L}^2_\mu(\mathcal{X}) := \{f\colon \mathcal{X} \to \mathbb{R} \mid \|f\|^2 < \infty\}$ with inner product

$$\langle f_1|f_2\rangle := \int_{\mathcal{X}} f_1(x) f_2(x) \mu(dx) \approx \frac{1}{N} \sum_{n=1}^{N} f_1(x_n) f_2(x_n)$$

- When $\mathcal{T}$ is symmetric PD, SVD = EVD

Parametric Approach

Operator SVD

$$\mathcal{T} = \sum_{i=1}^{\infty} \sigma_i |\phi_i\rangle\langle\psi_i|$$

where $\langle\phi_i|\phi_j\rangle = \langle\psi_i|\psi_j\rangle = \delta_{ij}$, $\sigma_1 \geq \sigma_2 \geq \ldots \geq 0$

- Hilbert space $\mathcal{F} := \mathscr{L}^2_\mu(\mathcal{X}) := \{f\colon \mathcal{X} \to \mathbb{R} \mid \|f\|^2 < \infty\}$ with inner product

$$\langle f_1|f_2\rangle := \int_{\mathcal{X}} f_1(x)f_2(x)\mu(dx) \approx \frac{1}{N}\sum_{n=1}^{N} f_1(x_n)f_2(x_n)$$

- When $\mathcal{T}$ is symmetric PD, SVD = EVD
- Integral kernel operator: $(\mathcal{K}\phi)(y) := \int k(x, y)\phi(x)\mu(dx)$

Parametric Approach

Operator SVD

$$\mathcal{T} = \sum_{i=1}^{\infty} \sigma_i |\phi_i\rangle\langle\psi_i|$$

where $\langle\phi_i|\phi_j\rangle = \langle\psi_i|\psi_j\rangle = \delta_{ij}$, $\sigma_1 \geq \sigma_2 \geq \ldots \geq 0$

- To train parametric eigen- (singular-) functions (parameterized by neural networks), solve an **optimization problem** that characterizes the top-L EVD/SVD

Parametric Approach

Operator SVD

$$\mathcal{T} = \sum_{i=1}^{\infty} \sigma_i |\phi_i\rangle\langle\psi_i|$$

where $\langle\phi_i|\phi_j\rangle = \langle\psi_i|\psi_j\rangle = \delta_{ij}$, $\sigma_1 \geq \sigma_2 \geq \ldots \geq 0$

- To train parametric eigen- (singular-) functions (parameterized by neural networks), solve an **optimization problem** that characterizes the top-L EVD/SVD

🙄 Most (if not all) existing methods are based on Rayleigh quotient maximization

Parametric Approach

Operator SVD

$$\mathcal{T} = \sum_{i=1}^{\infty} \sigma_i |\phi_i\rangle\langle\psi_i|$$

where $\langle\phi_i|\phi_j\rangle = \langle\psi_i|\psi_j\rangle = \delta_{ij}$, $\sigma_1 \geq \sigma_2 \geq \ldots \geq 0$

- To train parametric eigen- (singular-) functions (parameterized by neural networks), solve an **optimization problem** that characterizes the top-L EVD/SVD

🙄 Most (if not all) existing methods are based on Rayleigh quotient maximization

😎 We propose an optimization framework based on **nested low-rank approximation**!

Low-Rank Approximation (LoRA)

Theorem (Eckart–Young, 1936)

$$(\mathbf{f}_{1:L}^\star, \mathbf{g}_{1:L}^\star) \in \mathop{\arg\min}_{\substack{\mathbf{f}_1,\ldots,\mathbf{f}_L \in \mathbb{R}^M,\\ \mathbf{g}_1,\ldots,\mathbf{g}_L \in \mathbb{R}^N}} \Big\| \mathsf{T} - \sum_{i=1}^{L} \mathbf{f}_i \mathbf{g}_i^\intercal \Big\|_F^2$$

$$\textit{If } \mathsf{T} = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^\intercal, \quad \textit{then} \quad \sum_{i=1}^{L} \mathbf{f}_i^\star (\mathbf{g}_i^\star)^\intercal = \sum_{i=1}^{L} \sigma_i \mathbf{u}_i \mathbf{v}_i^\intercal$$

Low-Rank Approximation (LoRA)

Theorem (Schmidt, 1907)

$$(\mathbf{f}^\star_{1:L}, \mathbf{g}^\star_{1:L}) \in \operatorname*{arg\,min}_{\substack{f_1,\ldots,f_L \in \mathcal{F},\\ g_1,\ldots,g_L \in \mathcal{G}}} \left\| \mathcal{T} - \sum_{i=1}^{L} |f_i\rangle\langle g_i| \right\|_{HS}^2$$

$$\textit{If } \mathcal{T} = \sum_{i=1}^{\infty} \sigma_i |\phi_i\rangle\langle\psi_i|, \ \textit{then} \ \sum_{i=1}^{L} |f_i^\star\rangle\langle g_i^\star| = \sum_{i=1}^{L} \sigma_i |\phi_i\rangle\langle\psi_i|$$

Low-Rank Approximation (LoRA)

Theorem (Schmidt, 1907)

$$(\mathbf{f}^{\star}_{1:L}, \mathbf{g}^{\star}_{1:L}) \in \mathop{\arg\min}_{\substack{f_1,\ldots,f_L \in \mathcal{F},\\ g_1,\ldots,g_L \in \mathcal{G}}} \Big\| \mathcal{T} - \sum_{i=1}^{L} |f_i\rangle\langle g_i| \Big\|_{\mathrm{HS}}^2$$

$$\textit{If } \mathcal{T} = \sum_{i=1}^{\infty} \sigma_i |\phi_i\rangle\langle\psi_i|, \ \textit{then } \sum_{i=1}^{L} |f_i^{\star}\rangle\langle g_i^{\star}| = \sum_{i=1}^{L} \sigma_i |\phi_i\rangle\langle\psi_i|$$

$$\boxed{\mathcal{L}_{\mathrm{LoRA}}(\mathbf{f}_{1:L}, \mathbf{g}_{1:L}) := \Big\| \mathcal{T} - \sum_{i=1}^{L} |f_i\rangle\langle g_i| \Big\|_{\mathrm{HS}}^2 - \|\mathcal{T}\|_{\mathrm{HS}}^2}$$

Low-Rank Approximation (LoRA)

Theorem (Schmidt, 1907)

$$(\mathbf{f}^\star_{1:L}, \mathbf{g}^\star_{1:L}) \in \mathop{\arg\min}_{\substack{f_1,\ldots,f_L \in \mathcal{F},\\ g_1,\ldots,g_L \in \mathcal{G}}} \left\| \mathcal{T} - \sum_{i=1}^{L} |f_i\rangle\langle g_i| \right\|_{\mathrm{HS}}^2$$

$$\textit{If } \mathcal{T} = \sum_{i=1}^{\infty} \sigma_i |\phi_i\rangle\langle\psi_i|, \ \textit{then} \ \sum_{i=1}^{L} |f_i^\star\rangle\langle g_i^\star| = \sum_{i=1}^{L} \sigma_i |\phi_i\rangle\langle\psi_i|$$

$$\boxed{\mathcal{L}_{\mathrm{LoRA}}(\mathbf{f}_{1:L}, \mathbf{g}_{1:L}) := -2\sum_{i=1}^{L} \langle g_i | \mathcal{T} f_i\rangle + \sum_{i=1}^{L}\sum_{i'=1}^{L} \langle f_i | f_{i'}\rangle\langle g_i | g_{i'}\rangle}$$

Low-Rank Approximation (LoRA)

Theorem (Schmidt, 1907)

$$(\mathbf{f}^\star_{1:L}, \mathbf{g}^\star_{1:L}) \in \operatorname*{arg\,min}_{\substack{f_1,\ldots,f_L \in \mathcal{F},\\ g_1,\ldots,g_L \in \mathcal{G}}} \Big\| \mathcal{T} - \sum_{i=1}^{L} |f_i\rangle\langle g_i| \Big\|_{HS}^2$$

$$\textit{If } \mathcal{T} = \sum_{i=1}^{\infty} \sigma_i |\phi_i\rangle\langle\psi_i|, \ \textit{then} \ \sum_{i=1}^{L} |f_i^\star\rangle\langle g_i^\star| = \sum_{i=1}^{L} \sigma_i |\phi_i\rangle\langle\psi_i|$$

$$\boxed{\mathcal{L}_{\mathrm{LoRA}}(\mathbf{f}_{1:L}, \mathbf{g}_{1:L}) := -2\sum_{i=1}^{L} \langle g_i | \mathcal{T} f_i\rangle + \sum_{i=1}^{L}\sum_{i'=1}^{L} \langle f_i | f_{i'}\rangle\langle g_i | g_{i'}\rangle}$$

☺️ Unconstrained optimization with computable objective!

Low-Rank Approximation (LoRA)

Theorem (Schmidt, 1907)

$$(\mathbf{f}^\star_{1:L}, \mathbf{g}^\star_{1:L}) \in \operatorname*{arg\,min}_{\substack{f_1,\ldots,f_L\in\mathcal{F},\\ g_1,\ldots,g_L\in\mathcal{G}}} \left\| \mathcal{T} - \sum_{i=1}^{L} |f_i\rangle\langle g_i| \right\|_{\mathrm{HS}}^2$$

If $\mathcal{T} = \sum_{i=1}^{\infty} \sigma_i |\phi_i\rangle\langle\psi_i|$, *then* $\sum_{i=1}^{L} |f_i^\star\rangle\langle g_i^\star| = \sum_{i=1}^{L} \sigma_i |\phi_i\rangle\langle\psi_i|$

$$\boxed{\mathcal{L}_{\mathrm{LoRA}}(\mathbf{f}_{1:L}, \mathbf{g}_{1:L}) := -2\sum_{i=1}^{L}\langle g_i|\mathcal{T} f_i\rangle + \sum_{i=1}^{L}\sum_{i'=1}^{L}\langle f_i|f_{i'}\rangle\langle g_i|g_{i'}\rangle}$$

- ☺️ Unconstrained optimization with computable objective!
- 🤨 But, the optimal solution only captures the top-L subspaces (i.e., not ordered)

Nesting for Symmetry Breaking

- **High-level idea**: minimize $\mathcal{L}_{\mathsf{LoRA}}(\mathbf{f}_{1:i}, \mathbf{g}_{1:i})$ for $i = 1, \ldots, L$

Nesting for Symmetry Breaking

- **High-level idea**: minimize $\mathcal{L}_{\mathrm{LoRA}}(\mathbf{f}_{1:i}, \mathbf{g}_{1:i})$ for $i = 1, \ldots, L$
- **Why does this work?**

Nesting for Symmetry Breaking

- **High-level idea**: minimize $\mathcal{L}_{\mathsf{LoRA}}(\mathbf{f}_{1:i}, \mathbf{g}_{1:i})$ for $i = 1, \ldots, L$
- **Why does this work?**

$$
\begin{aligned}
(f_1^\star, g_1^\star) &\in \arg\min_{f_1, g_1} \mathcal{L}_{\mathsf{LoRA}}(f_1, g_1) \\
(\mathbf{f}_{1:2}^\star, \mathbf{g}_{1:2}^\star) &\in \arg\min_{\mathbf{f}_{1:2}, \mathbf{g}_{1:2}} \mathcal{L}_{\mathsf{LoRA}}(\mathbf{f}_{1:2}, \mathbf{g}_{1:2}) \\
&\vdots \\
(\mathbf{f}_{1:L}^\star, \mathbf{g}_{1:L}^\star) &\in \arg\min_{\mathbf{f}_{1:L}, \mathbf{g}_{1:L}} \mathcal{L}_{\mathsf{LoRA}}(\mathbf{f}_{1:L}, \mathbf{g}_{1:L})
\end{aligned}
$$

Nesting for Symmetry Breaking

- **High-level idea**: minimize $\mathcal{L}_{\mathsf{LoRA}}(\mathbf{f}_{1:i}, \mathbf{g}_{1:i})$ for $i = 1, \ldots, L$
- **Why does this work?**

$$
\begin{aligned}
|f_1^\star\rangle\langle g_1^\star| &= \sigma_1 |\phi_1\rangle\langle\psi_1| \\
(\mathbf{f}_{1:2}^\star, \mathbf{g}_{1:2}^\star) &\in \arg\min_{\mathbf{f}_{1:2}, \mathbf{g}_{1:2}} \mathcal{L}_{\mathsf{LoRA}}(\mathbf{f}_{1:2}, \mathbf{g}_{1:2}) \\
&\vdots \\
(\mathbf{f}_{1:L}^\star, \mathbf{g}_{1:L}^\star) &\in \arg\min_{\mathbf{f}_{1:L}, \mathbf{g}_{1:L}} \mathcal{L}_{\mathsf{LoRA}}(\mathbf{f}_{1:L}, \mathbf{g}_{1:L})
\end{aligned}
$$

Nesting for Symmetry Breaking

- **High-level idea**: minimize $\mathcal{L}_{\mathsf{LoRA}}(\mathbf{f}_{1:i}, \mathbf{g}_{1:i})$ for $i = 1, \ldots, L$
- **Why does this work?**

$$
\begin{aligned}
|f_1^\star\rangle\langle g_1^\star| &= \sigma_1|\phi_1\rangle\langle\psi_1| \\
|f_1^\star\rangle\langle g_1^\star| + |f_2^\star\rangle\langle g_2^\star| &= \sigma_1|\phi_1\rangle\langle\psi_1| + \sigma_2|\phi_2\rangle\langle\psi_2| \\
&\vdots \\
(\mathbf{f}_{1:L}^\star, \mathbf{g}_{1:L}^\star) &\in \arg\min_{\mathbf{f}_{1:L}, \mathbf{g}_{1:L}} \mathcal{L}_{\mathsf{LoRA}}(\mathbf{f}_{1:L}, \mathbf{g}_{1:L})
\end{aligned}
$$

Nesting for Symmetry Breaking

- **High-level idea**: minimize $\mathcal{L}_{\mathsf{LoRA}}(\mathbf{f}_{1:i}, \mathbf{g}_{1:i})$ for $i = 1, \ldots, L$
- **Why does this work?**

$$|f_1^\star\rangle\langle g_1^\star| = \sigma_1|\phi_1\rangle\langle\psi_1|$$

$$|f_1^\star\rangle\langle g_1^\star| + |f_2^\star\rangle\langle g_2^\star| = \sigma_1|\phi_1\rangle\langle\psi_1| + \sigma_2|\phi_2\rangle\langle\psi_2|$$

$$\vdots$$

$$|f_1^\star\rangle\langle g_1^\star| + \ldots + |f_L^\star\rangle\langle g_L^\star| = \sigma_1|\phi_1\rangle\langle\psi_1| + \ldots + \sigma_L|\phi_L\rangle\langle\psi_L|$$

Nesting for Symmetry Breaking

- **High-level idea**: minimize $\mathcal{L}_{\mathrm{LoRA}}(\mathbf{f}_{1:i}, \mathbf{g}_{1:i})$ for $i = 1, \ldots, L$
- **Why does this work?**

$$
\begin{aligned}
|f_1^\star\rangle\langle g_1^\star| &= \sigma_1|\phi_1\rangle\langle\psi_1| \\
|f_1^\star\rangle\langle g_1^\star| + |f_2^\star\rangle\langle g_2^\star| &= \sigma_1|\phi_1\rangle\langle\psi_1| + \sigma_2|\phi_2\rangle\langle\psi_2| \\
&\vdots \\
|f_1^\star\rangle\langle g_1^\star| + \ldots + |f_L^\star\rangle\langle g_L^\star| &= \sigma_1|\phi_1\rangle\langle\psi_1| + \ldots + \sigma_L|\phi_L\rangle\langle\psi_L|
\end{aligned}
$$

$\Longrightarrow$ $|f_i^\star\rangle\langle g_i^\star| = \sigma_i|\phi_i\rangle\langle\psi_i|$ for all $i \in [L]$ $\square$

Nesting for Symmetry Breaking

- **High-level idea**: minimize $\mathcal{L}_{\mathsf{LoRA}}(\mathbf{f}_{1:i}, \mathbf{g}_{1:i})$ for $i = 1, \ldots, L$
- **Why does this work?**

$$
\begin{aligned}
|f_1^\star\rangle\langle g_1^\star| &= \sigma_1|\phi_1\rangle\langle\psi_1| \\
|f_1^\star\rangle\langle g_1^\star| + |f_2^\star\rangle\langle g_2^\star| &= \sigma_1|\phi_1\rangle\langle\psi_1| + \sigma_2|\phi_2\rangle\langle\psi_2| \\
&\vdots \\
|f_1^\star\rangle\langle g_1^\star| + \ldots + |f_L^\star\rangle\langle g_L^\star| &= \sigma_1|\phi_1\rangle\langle\psi_1| + \ldots + \sigma_L|\phi_L\rangle\langle\psi_L|
\end{aligned}
$$

$\Longrightarrow |f_i^\star\rangle\langle g_i^\star| = \sigma_i|\phi_i\rangle\langle\psi_i|$ for all $i \in [L]$ $\square$

- How can we **implement** this idea?

Sequential **Nesting**

- **Idea**: for each $i \in [L]$,

$$\text{update } (f_i, g_i) \text{ as if } (\mathbf{f}_{1:i-1}, \mathbf{g}_{1:i-1}) \text{ were perfectly matched}$$

- **Implementation**: for each $i \in [L]$,

$$\text{update } (f_i^{(t)}, g_i^{(t)}) \text{ using gradient } \partial_{(f_i, g_i)} \mathcal{L}_{\mathsf{LoRA}}(\mathbf{f}_{1:i}^{(t)}, \mathbf{g}_{1:i}^{(t)})$$

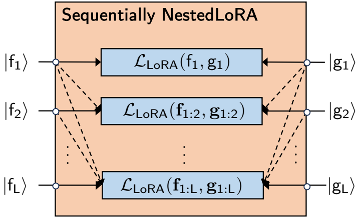
Sequential **Nesting**

- **Idea**: for each $i \in [L]$,

$$\text{update } (f_i, g_i) \text{ as if } (\mathbf{f}_{1:i-1}, \mathbf{g}_{1:i-1}) \text{ were perfectly matched}$$

- **Implementation**: for each $i \in [L]$,

$$\text{update } (f_i^{(t)}, g_i^{(t)}) \text{ using gradient } \partial_{(f_i, g_i)} \mathcal{L}_{\mathsf{LoRA}}(\mathbf{f}_{1:i}^{(t)}, \mathbf{g}_{1:i}^{(t)})$$



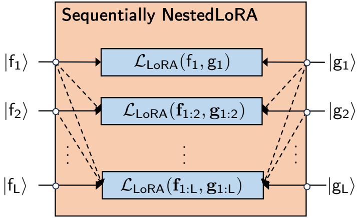
Sequential **Nesting**

- **Idea**: for each $i \in [L]$,

$$\text{update } (f_i, g_i) \text{ as if } (\mathbf{f}_{1:i-1}, \mathbf{g}_{1:i-1}) \text{ were perfectly matched}$$

- **Implementation**: for each $i \in [L]$,

$$\text{update } (f_i^{(t)}, g_i^{(t)}) \text{ using gradient } \partial_{(f_i, g_i)} \mathcal{L}_{\mathrm{LoRA}}(\mathbf{f}_{1:i}^{(t)}, \mathbf{g}_{1:i}^{(t)})$$



- Works as expected if (f_i, g_i) and (f_j, g_j) do not share parameters for any $i \neq j$

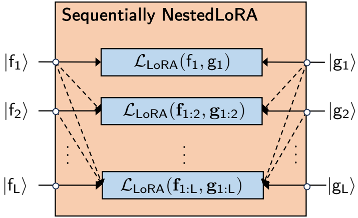
Sequential **Nesting**

- **Idea**: for each $i \in [L]$,

$$\text{update } (f_i, g_i) \text{ as if } (\mathbf{f}_{1:i-1}, \mathbf{g}_{1:i-1}) \text{ were perfectly matched}$$

- **Implementation**: for each $i \in [L]$,

$$\text{update } (f_i^{(t)}, g_i^{(t)}) \text{ using gradient } \partial_{(f_i, g_i)} \mathcal{L}_{\mathsf{LoRA}}(\mathbf{f}_{1:i}^{(t)}, \mathbf{g}_{1:i}^{(t)})$$



- Works as expected if (f_i, g_i) and (f_j, g_j) do not share parameters for any $i \neq j$
- 🤨 When $L \gg 1$, disjoint parameterization might be not feasible

Joint **Nesting**

🤔 Can there exist an optimization procedure for shared parameterization?

Joint **Nesting**

🤔 Can there exist an optimization procedure for shared parameterization?

- **Idea**: minimize a single objective $\mathcal{L}_{\mathsf{jnt}}(\mathbf{f}_{1:L}, \mathbf{g}_{1:L}; \mathbf{w}) := \sum_{i=1}^{L} w_i \mathcal{L}_{\mathsf{LoRA}}(\mathbf{f}_{1:i}, \mathbf{g}_{1:i})$

Joint **Nesting**

🤔 Can there exist an optimization procedure for shared parameterization?

- **Idea**: minimize a single objective $\mathcal{L}_{\mathsf{jnt}}(\mathbf{f}_{1:L}, \mathbf{g}_{1:L}; \mathbf{w}) := \sum_{i=1}^{L} w_i \mathcal{L}_{\mathsf{LoRA}}(\mathbf{f}_{1:i}, \mathbf{g}_{1:i})$

Joint **Nesting**

🤔 Can there exist an optimization procedure for shared parameterization?

- **Idea**: minimize a single objective $\mathcal{L}_{\mathrm{jnt}}(\mathbf{f}_{1:L}, \mathbf{g}_{1:L}; \mathbf{w}) := \sum_{i=1}^{L} w_i \mathcal{L}_{\mathrm{LoRA}}(\mathbf{f}_{1:i}, \mathbf{g}_{1:i})$
- **Implementation**:

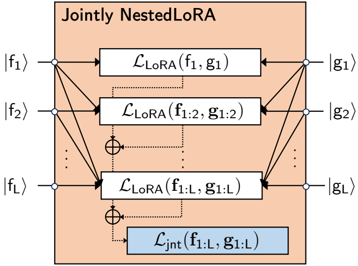
$$\text{update } (\mathbf{f}_{1:L}^{(t)}, \mathbf{g}_{1:L}^{(t)}) \text{ using gradient } \partial_{(\mathbf{f}_{1:L}, \mathbf{g}_{1:L})} \mathcal{L}_{\mathrm{jnt}}(\mathbf{f}_{1:L}^{(t)}, \mathbf{g}_{1:L}^{(t)}; \mathbf{w}))$$

Joint **Nesting**

🤔 Can there exist an optimization procedure for shared parameterization?

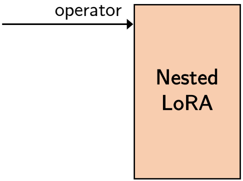
- **Idea**: minimize a single objective $\mathcal{L}_{\mathrm{jnt}}(\mathbf{f}_{1:L}, \mathbf{g}_{1:L}; \mathbf{w}) := \sum_{i=1}^{L} w_i \mathcal{L}_{\mathrm{LoRA}}(\mathbf{f}_{1:i}, \mathbf{g}_{1:i})$
- **Implementation**:

$$\text{update } (\mathbf{f}_{1:L}^{(t)}, \mathbf{g}_{1:L}^{(t)}) \text{ using gradient } \partial_{(\mathbf{f}_{1:L}, \mathbf{g}_{1:L})} \mathcal{L}_{\mathrm{jnt}}(\mathbf{f}_{1:L}^{(t)}, \mathbf{g}_{1:L}^{(t)}; \mathbf{w}))$$

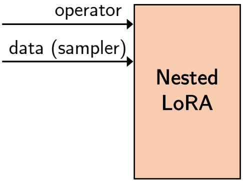


NeuralSVD = NestedLoRA + Neural Networks

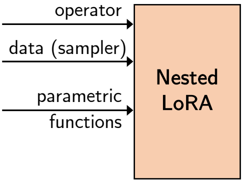
NeuralSVD = NestedLoRA + Neural Networks



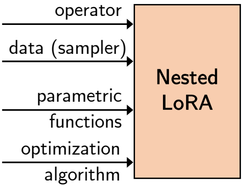
NeuralSVD = NestedLoRA + Neural Networks



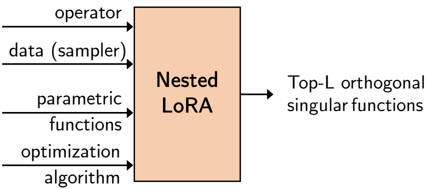
NeuralSVD = NestedLoRA + Neural Networks



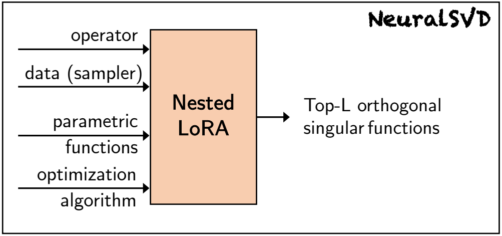
NeuralSVD = NestedLoRA + Neural Networks



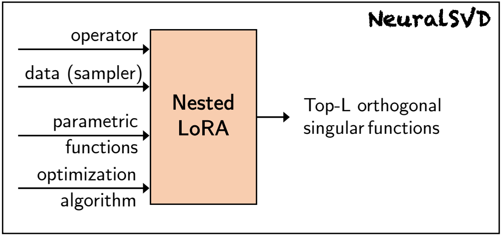
NeuralSVD = NestedLoRA + Neural Networks



NeuralSVD = NestedLoRA + Neural Networks

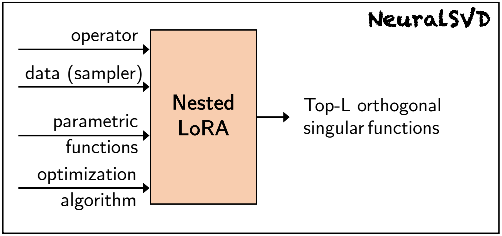


NeuralSVD = NestedLoRA + Neural Networks



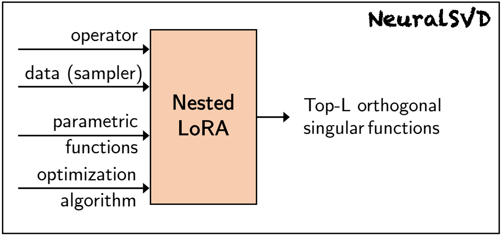
- Disjoint parameterization → sequential nesting
- Shared parameterization → joint nesting

NeuralSVD = NestedLoRA + Neural Networks



- Disjoint parameterization → sequential nesting
- Shared parameterization → joint nesting
  - Xu and Zheng (2023) proposed joint nesting for $k(x, y) = \frac{p(x,y)}{p(x)p(y)} - 1$ (canonical dependence kernel)

NeuralSVD $=$ NestedLoRA $+$ Neural Networks



- Disjoint parameterization $\to$ sequential nesting
- Shared parameterization $\to$ joint nesting
  - Xu and Zheng (2023) proposed joint nesting for $k(x, y) = \frac{p(x,y)}{p(x)p(y)} - 1$ (canonical dependence kernel)
- **Practical considerations**: NN architecture / optimization algorithm

Experiment 1. Schrödinger Equation (2D Hydrogen atom)

- Time-independent Schrödinger equation: for Hamiltonian $\mathcal{H} := -\nabla^2 + \mathcal{V}$,

$$\mathcal{H}|\psi\rangle = \mathsf{E}|\psi\rangle$$

- Single-electron potential $\mathcal{V}(\mathbf{x}) = -\frac{1}{\|\mathbf{x}\|_2}$

Experiment 1. Schrödinger Equation (2D Hydrogen atom)

- Time-independent Schrödinger equation: for Hamiltonian $\mathcal{H} := -\nabla^2 + \mathcal{V}$,

$$\mathcal{H}|\psi\rangle = \mathsf{E}|\psi\rangle$$

- Single-electron potential $\mathcal{V}(\mathbf{x}) = -\frac{1}{\|\mathbf{x}\|_2}$
- Closed-form solution $(d = 2)$

  Each eigenstate is parameterized by a pair of integers (n, l) for $n \geq 0$ and $-n \leq l \leq n$

  Eigenvalues are $\lambda_{n,l} := \frac{1}{4}(n + \frac{1}{2})^{-2}$

Experiment 1. Schrödinger Equation (2D Hydrogen atom)

- Time-independent Schrödinger equation: for Hamiltonian $\mathcal{H} := -\nabla^2 + \mathcal{V}$,

$$\mathcal{H}|\psi\rangle = \mathsf{E}|\psi\rangle$$

- Single-electron potential $\mathcal{V}(\mathbf{x}) = -\frac{1}{\|\mathbf{x}\|_2}$
- Closed-form solution $(d = 2)$

  Each eigenstate is parameterized by a pair of integers (n, l) for $n \geq 0$ and $-n \leq l \leq n$

  Eigenvalues are $\lambda_{n,l} := \frac{1}{4}(n + \frac{1}{2})^{-2}$
- We decompose the **negative Hamiltonian** (ground-state first)

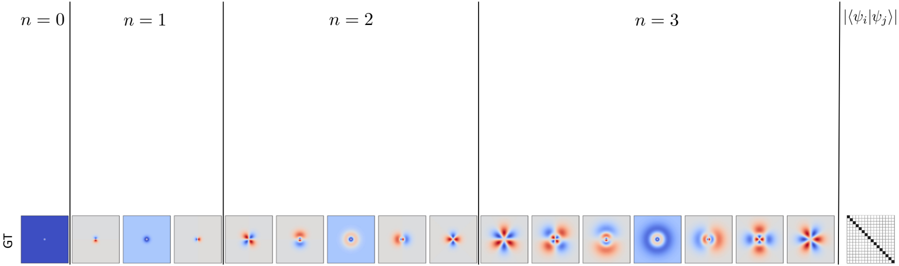


Figure: Learned eigenfunctions from SpIN, NeuralEF, and NeuralSVD.

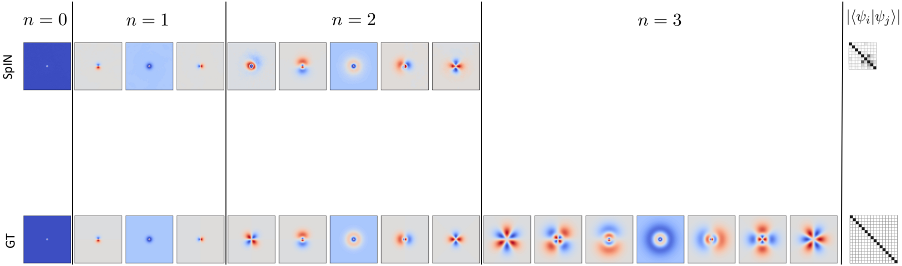


Figure: Learned eigenfunctions from SpIN, NeuralEF, and NeuralSVD.

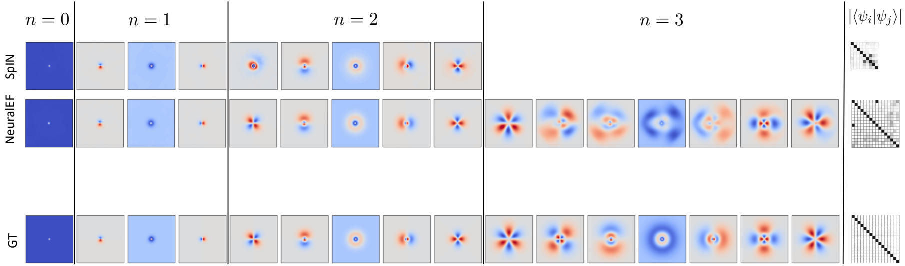


Figure: Learned eigenfunctions from SpIN, NeuralEF, and NeuralSVD.

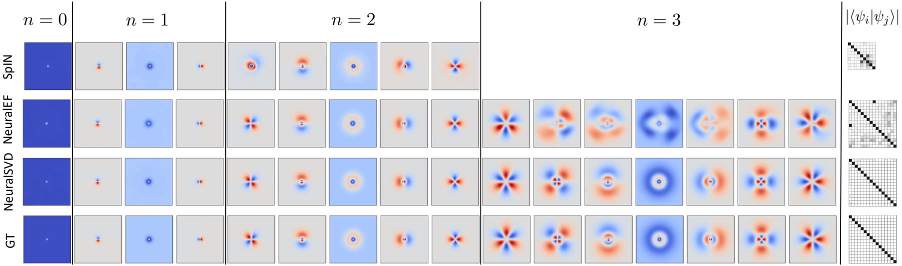


Figure: Learned eigenfunctions from SpIN, NeuralEF, and NeuralSVD.

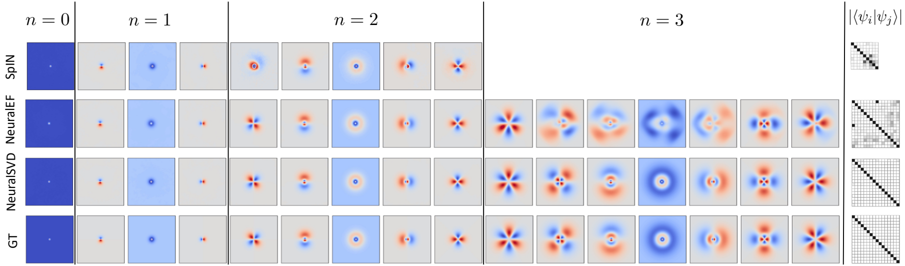


Figure: Learned eigenfunctions from SpIN, NeuralEF, and NeuralSVD.

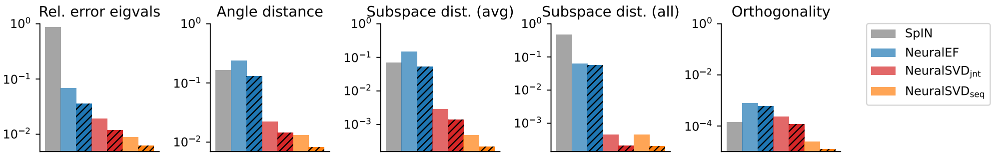
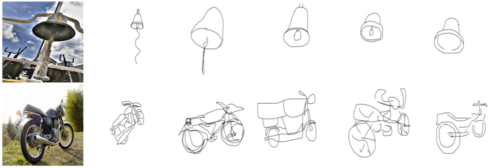
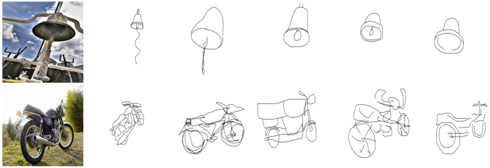


Figure: Summary of quantitative evaluations for solving TISE of 2D hydrogen atom. Non-hatched, light-colored bars represent batch size of 128, while hatched bars indicate batch size of 512.

Experiment 2. Cross-Domain Retrieval with CDK

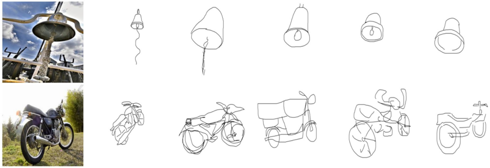


- **Goal**: given a human sketch ($\mathbf{x}$), retrieve photos ($\mathbf{y}$)



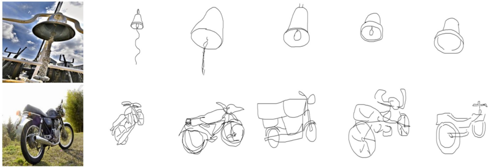
- **Goal**: given a human sketch ($\mathbf{x}$), retrieve photos ($\mathbf{y}$)
- **Our method**: maximum likelihood retrieval with canonical dependence kernel (CDK)

Experiment 2. Cross-Domain Retrieval with CDK



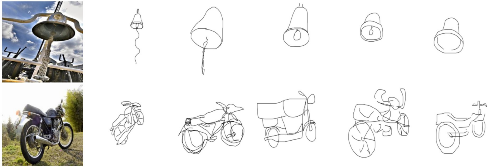
- **Goal**: given a human sketch ($\mathbf{x}$), retrieve photos ($\mathbf{y}$)
- **Our method**: maximum likelihood retrieval with canonical dependence kernel (CDK)
  - Define $p(\mathbf{x}, \mathbf{y}) = E_{p(c)}[p(\mathbf{x}|c)p(\mathbf{y}|c)]$

Experiment 2. Cross-Domain Retrieval with CDK



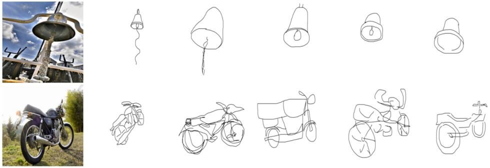
- **Goal**: given a human sketch ($\mathbf{x}$), retrieve photos ($\mathbf{y}$)
- **Our method**: maximum likelihood retrieval with canonical dependence kernel (CDK)
  - Define $p(\mathbf{x}, \mathbf{y}) = \mathrm{E}_{p(c)}[p(\mathbf{x}|c)p(\mathbf{y}|c)]$
  - (Training) Decompose CDK $\dfrac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x})p(\mathbf{y})} - 1 \approx \mathbf{f}_{1:L}(\mathbf{x})^{\intercal}\mathbf{g}_{1:L}(\mathbf{y})$

Experiment 2. Cross-Domain Retrieval with CDK



- **Goal**: given a human sketch ($\mathbf{x}$), retrieve photos ($\mathbf{y}$)
- **Our method**: maximum likelihood retrieval with canonical dependence kernel (CDK)
  - Define $p(\mathbf{x}, \mathbf{y}) = E_{p(c)}[p(\mathbf{x}|c)p(\mathbf{y}|c)]$
  - (Training) Decompose CDK $\dfrac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x})p(\mathbf{y})} - 1 \approx \mathbf{f}_{1:L}(\mathbf{x})^\intercal \mathbf{g}_{1:L}(\mathbf{y})$
  - (Inference) Given $\mathbf{x}$, retrieve $\arg\max_{\mathbf{y}} p(\mathbf{x}|\mathbf{y}) \approx \arg\max_{\mathbf{y}} \mathbf{f}_{1:L}(\mathbf{x})^\intercal \mathbf{g}_{1:L}(\mathbf{y})$

Experiment 2. Cross-Domain Retrieval with CDK



- **Goal**: given a human sketch ($\mathbf{x}$), retrieve photos ($\mathbf{y}$)
- **Our method**: maximum likelihood retrieval with canonical dependence kernel (CDK)
  - Define $p(\mathbf{x}, \mathbf{y}) = E_{p(c)}[p(\mathbf{x}|c)p(\mathbf{y}|c)]$
  - (Training) Decompose CDK $\frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x})p(\mathbf{y})} - 1 \approx \mathbf{f}_{1:L}(\mathbf{x})^\intercal \mathbf{g}_{1:L}(\mathbf{y})$
  - (Inference) Given $\mathbf{x}$, retrieve $\arg\max_{\mathbf{y}} p(\mathbf{x}|\mathbf{y}) \approx \arg\max_{\mathbf{y}} \mathbf{f}_{1:L}(\mathbf{x})^\intercal \mathbf{g}_{1:L}(\mathbf{y})$

☺️ **Structured representation**: coordinates are ordered in the order of importance

Table: Evaluation of the ZS-SBIR task with the Sketchy Extended dataset [4].

Model	Gen. model	Ext. knowledge	P@100	mAP	Split
LCALE [3]	$*$	Word embed.	0.583	0.476	1
IIAE [2]	$*$		0.659	0.573	1
NeuralSVD			**0.670** $\pm$0.010	**0.581** $\pm$0.008	1
			0.724 $\pm$0.008	0.641 $\pm$0.008	2

Table: Evaluation of the ZS-SBIR task with the Sketchy Extended dataset [4].

Model	Gen. model	Ext. knowledge	P@100	mAP	Split
LCALE [3]	∗	Word embed.	0.583	0.476	1
IIAE [2]	∗		0.659	0.573	1
NeuralSVD			**0.670** ±0.010	**0.581** ±0.008	1
			0.724 ±0.008	0.641 ±0.008	2

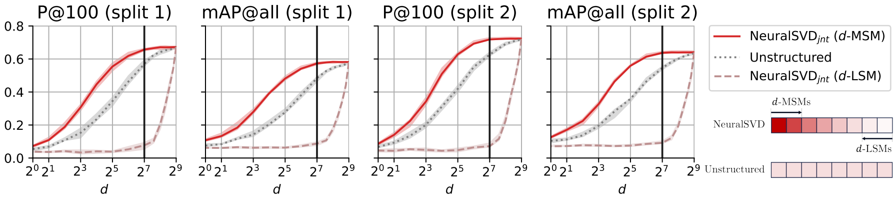
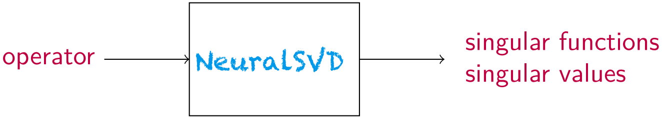


Figure: The mAP performance of NeuralSVD on ZS-SBIR task, when varying dimensions.

Postscript



Postscript

operator $\longrightarrow$ NeuralSVD $\longrightarrow$ singular functions
singular values

- NeuralSVD $\rightarrow$ principled & scalable solutions for innumerous applications

Postscript

operator $\longrightarrow$ NeuralSVD $\longrightarrow$ singular functions
singular values

- NeuralSVD $\rightarrow$ principled & scalable solutions for innumerous applications
- **Nested low-rank approximation** (Nested LoRA)
  - ✓ Unconstrained optimization
  - ✓ Unbiased gradient estimates
  - ✓ No additional regularization required

Postscript

operator $\longrightarrow$ NeuralSVD $\longrightarrow$ singular functions
singular values

- NeuralSVD $\rightarrow$ principled & scalable solutions for innumerous applications
- **Nested low-rank approximation** (Nested LoRA)

☺️ Way a lot more to come!

Postscript

operator $\longrightarrow$ NeuralSVD $\longrightarrow$ singular functions
singular values

- NeuralSVD $\rightarrow$ principled & scalable solutions for innumerous applications
- **Nested low-rank approximation** (Nested LoRA)

😊 Way a lot more to come!

- 🔬 **AI for science (physical simulations)**: differential operators
- 🧠 **Representation learning**: CDK, neural kernels (e.g., neural Fisher kernel, NTK), ...
- 🕸️ **Graph information processing**: graph Laplacians
- 📈 **Dynamical systems / time-series modeling**: Koopman operator
- ...

Postscript

operator → NeuralSVD → singular functions, singular values

- NeuralSVD → principled & scalable solutions for innumerous applications
- **Nested low-rank approximation** (Nested LoRA)

😊 Way a lot more to come!

- 🔬 **AI for science (physical simulations)**: differential operators
- 🧠 **Representation learning**: CDK, neural kernels (e.g., neural Fisher kernel, NTK), ...
- 🕸 **Graph information processing**: graph Laplacians
- 📈 **Dynamical systems / time-series modeling**: Koopman operator
- ...

😉 Check out our preprint and implementation: jongharyu.github.io

Unified Implementation: **Gradient Masking**

- A unified gradient expression:

$$|\,\partial_{f_i}\mathcal{L}\rangle = 2\Big\{-m_i|\mathcal{T}^*g_i\rangle + \sum_{i'=1}^{L} M_{ii'}|f_{i'}\rangle\langle g_{i'}|g_i\rangle\Big\}$$

- For sequential nesting:

$$\mathbf{m} \leftarrow \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \quad M \leftarrow \begin{bmatrix} 1 & 1 & \dots & 1 \\ 0 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

- For joint nesting:

$$\mathbf{m} \leftarrow \begin{bmatrix} w_1 + w_2 + \ldots + w_L \\ w_2 + \ldots + w_L \\ \vdots \\ w_L \end{bmatrix}, \quad M \leftarrow \begin{bmatrix} w_1 & w_2 & \dots & w_L \\ w_2 & w_2 & \dots & w_L \\ \vdots & \vdots & \ddots & \vdots \\ w_L & w_L & \dots & w_L \end{bmatrix}$$

[1] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, September 1936. ISSN 0033-3123, 1860-0980. doi: 10.1007/BF02288367.

[2] HyeongJoo Hwang, Geon-Hyeong Kim, Seunghoon Hong, and Kee-Eung Kim. Variational interaction information maximization for cross-domain disentanglement. In *Adv. Neural Inf. Proc. Syst.*, volume 33, 2020.

[3] Kaiyi Lin, Xing Xu, Lianli Gao, Zheng Wang, and Heng Tao Shen. Learning cross-aligned latent embeddings for zero-shot cross-modal retrieval. In *Proc. AAAI Conf. Artif. Int.*, volume 34, pages 11515–11522, 2020.

[4] Li Liu, Fumin Shen, Yuming Shen, Xianglong Liu, and Ling Shao. Deep sketch hashing: Fast free-hand sketch-based image retrieval. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 2862–2871, 2017.

[5] Erhard Schmidt. Zur Theorie der linearen und nichtlinearen Integralgleichungen. *Math. Ann.*, 63(4):433–476, December 1907. ISSN 0025-5831, 1432-1807. doi: 10.1007/BF01449770.

References II

[6] Xiangxiang Xu and Lizhong Zheng. A geometric framework for neural feature learning. *arXiv preprint arXiv:2309.10140*, 2023.