Operator SVD with Neural Networks via Nested Low-Rank Approximation

Jongha (Jon) Ryu[1], Xiangxiang Xu[1], H. S. Melihcan Erol[1], Yuheng Bu[2], Lizhong Zheng[1], Gregory W. Wornell[1]

[1]Department of EECS, MIT [2]Department of ECE, University of Florida

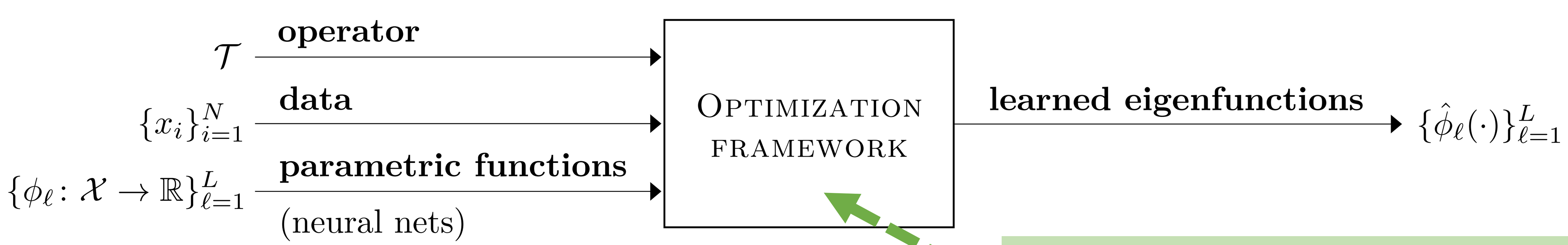
Various engineering / scientific problems can be reduced to "**Eigenvalue Problem (EVP)**"

A canonical example is the time-independent Schrodinger equation: $\mathcal{H}|\psi\rangle = \lambda|\psi\rangle$

A standard approach quantizes the problem and solves a matrix EVP → **NOT SCALABLE!**

The "parametric" approach

which has become popular recently in quantum chemistry



We propose a **new** optimization framework which can lead to a top-L **ordered** orthogonal eigenbasis without much hassle!

why parametric?

Criterion	**Nonparametric** (matrix)	**Parametric** (operator)
Training	computational complexity	optimization complexity
Testing	Nyström formula (computational complexity, storage complexity, statistical complexity)	neural networks (approximation error, optimization error)
For PDEs	discretization	mesh-free

why challenging?

Most existing approaches are based on the Rayleigh quotient maximization $\lambda_1 = \max_{\hat{\psi}_1\colon \|\hat{\psi}_1\|_2=1} \langle\hat{\psi}_1|\mathcal{T}\hat{\psi}_1\rangle$ (for top-1 mode)

and a variant of the trace maximization $\max_{\hat{\psi}_1,\ldots,\hat{\psi}_L} \sum_{i=1}^L \langle\hat{\psi}_i|\mathcal{T}\hat{\psi}_i\rangle$ s.t. $\langle\hat{\psi}_i|\hat{\psi}_j\rangle = \delta_{ij}$ → hard to deal with!

Finding an ordered eigenbasis also requires an additional care

Our approach: **Nested Low-rank Approximation**

Intuition

$$|f_1^\star\rangle\langle f_1^\star| = \lambda_1|\psi_1\rangle\langle\psi_1|$$
$$|f_1^\star\rangle\langle f_1^\star| + |f_2^\star\rangle\langle f_2^\star| = \lambda_1|\psi_1\rangle\langle\psi_1| + \lambda_2|\psi_2\rangle\langle\psi_2|$$
$$\vdots$$
$$|f_1^\star\rangle\langle f_1^\star| + \ldots + |f_L^\star\rangle\langle f_L^\star| = \lambda_1|\psi_1\rangle\langle\psi_1| + \ldots + \lambda_L|\psi_L\rangle\langle\psi_L|$$

ingredient 1. low-rank approximation (LoRA)

Schmidt theorem (1907) (a.k.a. Eckart-Young theorem (1936))

If $f_1^\star,\ldots,f_L^\star \in \arg\min_{f_1,\ldots,f_L} \left\|\mathcal{T} - \sum_{\ell=1}^L |f_\ell\rangle\langle f_\ell|\right\|_{\mathsf{HS}}^2$, then $(f_1^\star,\ldots,f_L^\star)$ is a "scaled" top-L orthonormal eigenbasis up to a rotation

(or equivalently, $\sum_{\ell=1}^L |f_\ell^\star\rangle\langle f_\ell^\star| = \sum_{\ell=1}^L \lambda_\ell|\psi_\ell\rangle\langle\psi_\ell|$)

$$\Longrightarrow \mathcal{L}(\mathbf{f}_{1:L}) \triangleq \left\|\mathcal{T} - \sum_{\ell=1}^L |f_\ell\rangle\langle f_\ell|\right\|_{\mathsf{HS}}^2 - \|\mathcal{T}\|_{\mathsf{HS}}^2 = -2\sum_{\ell=1}^L \langle f_\ell|\mathcal{T}f_\ell\rangle + \sum_{\ell=1}^L\sum_{\ell'=1}^L \langle f_\ell|f_{\ell'}\rangle^2$$

Remark: an unconstrained optimization problem!

ingredient 2. nesting

Idea: solve $\min_{\mathbf{f}_{1:\ell}} \mathcal{L}(\mathbf{f}_{1:\ell})$ for each $\ell = 1,\ldots,L$

Theorem (joint nesting)

Define $\mathcal{L}_{\mathsf{nested}}(\mathbf{f};\mathbf{w}) \triangleq \sum_{\ell=1}^L w_\ell \mathcal{L}(\mathbf{f}_{1:\ell})$ for any positive weights.

If $\mathbf{f}_{1:L}^\dagger \in \arg\min_{\mathbf{f}_{1:L}} \mathcal{L}_{\mathsf{nested}}(\mathbf{f};\mathbf{w})$, then $|f_\ell^\dagger\rangle = \sqrt{\lambda_\ell}|\psi_\ell\rangle$.

Remark 1: can recover the **ordered** orthogonal eigenbasis!

Remark 2: gradient can be estimated **without bias**

Comparison to existing methods

NeuralSVD = NestedLoRA + NNs

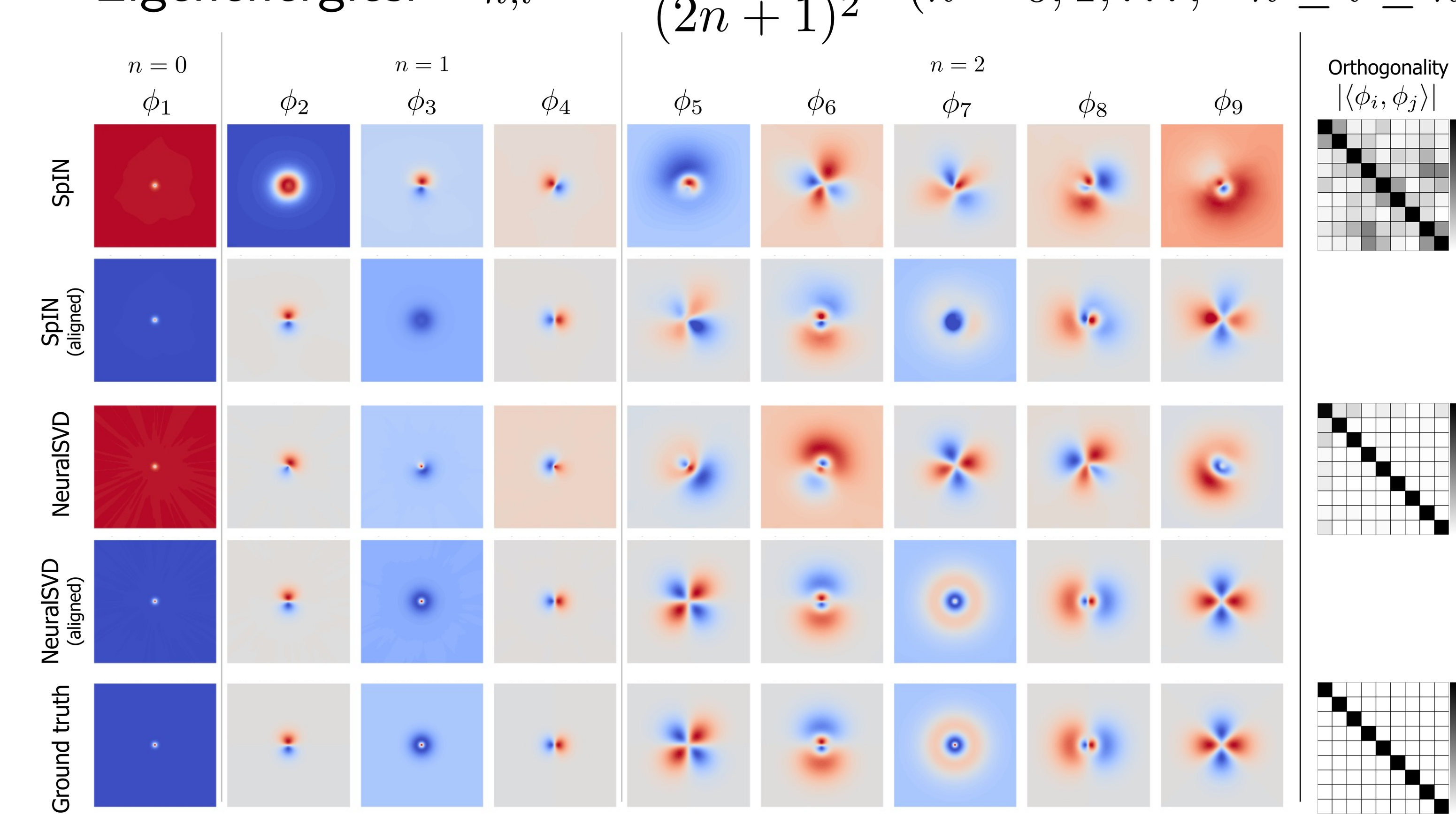
	SpIN	**NeuralEF**	**NeuralSVD**
Goal	EVD	EVD	SVD
Unbiased gradient estimates	✓	✗	✓
To handle orthogonality constraints	(per-step) Cholesky decomposition	function normalization	-
To remove bias in gradient estimates	bi-level optimization	large batch size	-

Remarks and future directions

- Our approach can naturally perform **SVD**!
- There is yet another (better) version of nesting! (see full paper)
- Also applicable to (some) **non-compact** operators! (see full paper)
- **Various other applications**
  - other PDEs (see full paper)
  - machine learning: correlation analysis / embedding
    - canonical dependence kernel $k(x,y) = \frac{p(x,y)}{p(x)p(y)}$ (see full paper)
    - graph Laplacians
  - control: Koopman operators

Simple demonstration: 2D hydrogen atom

- Hamiltonian: $\mathcal{H} = -\nabla^2 - \frac{1}{\|\mathbf{x}\|_2}$
- Eigenenergies: $E_{n,l} = -\frac{1}{(2n+1)^2}$ $(n = 0,1,\ldots,-n \le l \le n)$



full paper + code

https://bit.ly/49ORn3z