Operator SVD with Neural Networks via Nested Low-Rank Approximation

J. Jon Ryu[1] **Xiangxiang Xu**[1] **H. S. Melihcan Erol**[1] **Yuheng Bu**[2]
Lizhong Zheng[1] **Gregory W. Wornell**[1]
[1]Dept. of EECS, MIT [2]Dept. of ECE, University of Florida
{jongha,xuxx,hsmerol,lizhong,gww}@mit.edu
buyuheng@ufl.edu

Abstract

This paper proposes an optimization-based method to learn the singular value decomposition (SVD) of a compact operator with ordered singular functions. The proposed objective function is based on Schmidt's low-rank approximation theorem (1907) that characterizes a truncated SVD as a solution minimizing the mean squared error, accompanied with a technique called *nesting* to learn the ordered structure. When the optimization space is parameterized by neural networks, we refer to the proposed method as *NeuralSVD*. The implementation does not require sophisticated optimization tricks unlike existing approaches.

1 Introduction

Spectral decomposition techniques, including singular value decomposition (SVD) and eigenvalue decomposition (EVD), are crucial tools in machine learning and data science for handling large datasets and reducing their dimensionality while preserving prominent structures; see, e.g., [11, 4]. These form the foundation of various low-dimensional embedding algorithms [18, 21, 15, 19, 13, 2, 3, 5] and correlation analysis algorithms [12, 23] and are widely used in image and signal processing [1, 22, 24, 20, 16], natural language processing [9, 7], among other fields. Beyond the learning applications, solving eigenvalue problems is often a crucial step in solving partial differential equations from physical sciences, such as Schrödinger's equations in quantum chemistry.

For large-scale, high-dimensional data, however, the memory, computational, and statistical complexity of spectral decomposition poses a significant challenge in practice. While there exist streaming-type algorithms that may alleviate the issues with large-scale data, the standard approach suffers the curse of dimensionality, which is not easy to deal within the matrix decomposition framework. A promising alternative is to approximate the singular- or eigen-functions using parametric function approximators, assuming that there exists an abstract operator that induces a target matrix to decompose. Given the exceptional ability of neural networks to generalize with complex data, such as convolutional neural networks for images and transformers for natural language, one can anticipate better extrapolation performance than in the matrix approach. By encoding the spectral information to a single function, this framework can also significantly reduce the test-time complexity.

The idea of using neural networks to approximately perform spectral decomposition of an operator was explored in the machine learning community by Pfau et al. [14] and Deng et al. [6] who proposed a framework to systematically recover the ordered eigenfunctions. There exists a separate line of (long) history in computational physics to numerically solve Schrödinger equations via *neural-network ansatzes* (trial wavefunctions), but most, if not all, of the existing methods do not have a systematic way to learn the top-L eigenfunctions.

Links to the full paper and code can be found at the first author's website: https://jongharyu.github.io.

Machine Learning and the Physical Sciences Workshop, NeurIPS 2023.

In this paper, we propose a new optimization framework that can train neural networks to approximate the eigen- (or singular-) functions of a compact operator, which can address the issues of the prior works. We characterize the ordered singular functions as the unique global optimizer of a single optimization problem, based on Schmidt's low-rank approximation theorem (1907) and a trick called *nesting* to enforce the structure. We demonstrate the power of our framework for solving a simple Schrödinger equation among many other applications.

2 Motivation and Overview

As an alternative solution to the aforementioned problems with the matrix approach, we advocate the parametric, optimization-based approach, which aims to train parametric functions (which are often neural networks) to fit the desired singular- or eigen- functions. We start from an abstract setting, where our problem of interest is already reduced to finding the EVD or SVD of a linear operator. As our main framework of the low-rank approximation naturally characterizes SVD, we start by describing SVD and show how we can perform EVD within the same framework as a special case.

Operator SVD We consider two separable Hilbert spaces $\mathcal{F}$ and $\mathcal{G}$ and a linear operator $\mathcal{T}\colon \mathcal{F} \to \mathcal{G}$. For most applications, the Hilbert spaces $\mathcal{F}$ and $\mathcal{G}$ are L^2 spaces. In learning problems, $\mathcal{T}$ is typically given as an integral kernel operator induced by a kernel function, accompanied with data distributions. In solving PDEs, $\mathcal{T}$ is often a differential operator that governs a physical system of interest, where the underlying measure is the Lebesgue measure over a domain.

It is well known that for a compact operator $\mathcal{T}$, there exist orthonormal bases $(\phi_i)_{i\geq 1}$ and $(\psi_i)_{i\geq 1}$ with a sequence of non-increasing, non-negative real numbers $(\sigma_i)_{i\geq 1}$ such that $(\mathcal{T}\phi_i)(y) = \sigma_i\psi_i(y)$, $(\mathcal{T}^*\psi_i)(x) = \sigma_i\phi_i(x)$, $i = 1, 2, \ldots$. The function pairs (ϕ_i, ψ_i) are called (left- and right-, resp.) singular functions corresponding to the singular value σ_i. Hence, the compact operator $\mathcal{T}$ can be written as $\mathcal{T} = \sum_{i=1}^{\infty} \sigma_i |\psi_i\rangle\langle\phi_i|$, for $\sigma_1 \geq \sigma_2 \geq \ldots \geq 0$, which we call the SVD of $\mathcal{T}$.

EVD as a special case of SVD In several applications, the operator is self-adjoint (i.e., $\mathcal{T}^* = \mathcal{T}$ with $\mathcal{F} = \mathcal{G}$), and sometimes even positive definite. By the spectral theorem, a compact self-adjoint operator has the EVD of the form $\mathcal{T} = \sum_{i=1}^{\infty} \lambda_i |\phi_i\rangle\langle\phi_i|$. In this case, the singular values of the operator are the absolute values of its eigenvalues, and for each i, the i-th left- and right- singular functions are either identical (if $\lambda_i \geq 0$) or only different by the sign (if $\lambda_i < 0$). Hence, in particular, we can find its EVD by SVD in the case of a positive-definite (PD) operator.

Prior Works on Spectral Decomposition with Neural Networks As alluded to earlier, Spectral Inference Networks (SpIN) [14] and Neural Eigenfunctions (NeuralEF) [6] were proposed to train neural networks to approximate EVD of a compact, self-adjoint, PD operator $\mathcal{T}\colon \mathcal{F} \to \mathcal{F}$. They are both based on the following standard characterization of EVD, aiming to learn ℓ-th eigenfunction assuming that the top $(\ell - 1)$ eigenfunctions are well learned:

$$(P_\ell) \quad \begin{aligned} &\underset{\tilde{\phi}_\ell \in \mathcal{F}}{\text{maximize}} \;\; \langle \tilde{\phi}_\ell | \mathcal{T} \tilde{\phi}_\ell \rangle \\ &\text{subject to} \;\; \langle \tilde{\phi}_\ell | \tilde{\phi}_i \rangle = \delta_{\ell i} \quad \forall 1 \leq i \leq \ell. \end{aligned}$$

The difficulty in this formulation, however, lies in dealing with the hard orthogonality constraints. SpIN and NeuralEF propose different variants of this formulation with tailored optimization techniques, but they suffer crucial issues in practice; see Table 1 for a high-level comparison.

3 SVD via Nested Low-Rank Approximation

We propose a new optimization-based algorithm for SVD with neural networks, based on a century-old theory from functional analysis (Schmidt's approximation theorem) combined with a recently proposed technique (the "nesting trick") in the literature. The resulting framework is easier to understand compared to the prior methods, and admits a straightforward implementation.

Table 1: Comparison with SpIN [14] and NeuralEF [6].

	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	-

Low-Rank Approximation Let $\mathbf{f}_{1:\ell}(x) := [f_1(x), \dots, f_\ell(x)]^\intercal$. For the top-$L$ SVD of a given operator $\mathcal{T}$, our main objective function is based on the low-rank approximation (LoRA):

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

Here, $\|\mathcal{T}\|_{\mathsf{HS}}^2$ denotes the Hilbert–Schmidt norm of an operator $\mathcal{T}$, which can be understood as the operator version of the Frobenius norm for matrices. By Schmidt's low-rank approximation theorem [17], $(\mathbf{f}^\star, \mathbf{g}^\star)$ corresponds to the rank-L approximation of $\mathcal{T}$ in the following sense.

Theorem 1 (Informal)**.** *Define* $(\mathbf{f}^\star, \mathbf{g}^\star) := \arg\min_{f_\ell \in \mathcal{F}, g_\ell \in \mathcal{G},\ \ell \in [L]} \mathcal{L}_{\mathsf{LoRA}}(\mathbf{f}_{1:L}, \mathbf{g}_{1:L})$. *Then,* $\sum_{\ell=1}^{L} |g_\ell^\star\rangle\langle f_\ell^\star| = \sum_{\ell=1}^{L} \sigma_\ell |\psi_\ell\rangle\langle\phi_\ell|$.

Note that $(\mathsf{P}\mathbf{f}^\star, \mathsf{P}\mathbf{g}^\star)$ for any orthogonal matrix $\mathsf{P} \in \mathbb{R}^{L\times L}$ is also a global minimizer. This implies that this global minimzier only characterizes the top-L singular *subspaces*, and we thus require an additional trick to find the ordered singular functions and the singular values.

Nesting The key observation to break the symmetry in the objective $\mathcal{L}_{\mathsf{LoRA}}(\mathbf{f}_{1:L}, \mathbf{g}_{1:L})$ is that the ordered singular values and functions $\{(\sigma_\ell, \phi_\ell, \psi_\ell)\}_{\ell=1}^L$ can be characterized as the global minimizer of a single objective function, by taking a weighted sum of $\{\mathcal{L}_{\mathsf{LoRA}}(\mathbf{f}_{1:\ell}, \mathbf{g}_{1:\ell})\}_{\ell=1}^L$ with positive weights; see Theorem 2 below. That is,

$$\underset{\mathbf{f}_{1:L}, \mathbf{g}_{1:L}}{\text{minimize}}\, \mathcal{L}_{\mathsf{NestedLoRA}}(\mathbf{f}_{1:L}, \mathbf{g}_{1:L}),$$

where we define, for any positive weights $\mathbf{w} = (w_1, \dots, w_L) \in \mathbb{R}_{>0}^L$,

$$\begin{aligned}\mathcal{L}_{\mathsf{NestedLoRA}}(\mathbf{f}_{1:L}, \mathbf{g}_{1:L}) &:= \sum_{\ell=1}^{L} w_\ell \mathcal{L}_{\mathsf{LoRA}}(\mathbf{f}_{1:\ell}, \mathbf{g}_{1:\ell}) \\ &= -2\sum_{\ell=1}^{L} \mathsf{m}_\ell \langle g_\ell | \mathcal{T} f_\ell \rangle + \sum_{\ell=1}^{L}\sum_{\ell'=1}^{L} \mathsf{M}_{\ell\ell'} \langle f_\ell | f_\ell \rangle \langle g_\ell | g_\ell \rangle.\end{aligned}$$

Here, we define the vector mask as $\mathsf{m}_\ell := \sum_{i=\ell}^{L} w_i$ and the matrix mask as $\mathsf{M}_{\ell\ell'} = \mathsf{m}_{\max\{\ell,\ell'\}}$.

Theorem 2 (Informal)**.** *Let* $(\mathbf{f}_{1:L}^\dagger, \mathbf{g}_{1:L}^\dagger) := \arg\min_{f_\ell \in \mathcal{F}, g_\ell \in \mathcal{G}, \ell \in [L]} \mathcal{L}_{\mathsf{NestedLoRA}}(\mathbf{f}_{1:L}, \mathbf{g}_{1:L}; \mathbf{w})$. *Then, for each* ℓ, $|g_\ell^\dagger\rangle\langle f_\ell^\dagger| = \sigma_\ell |\psi_\ell\rangle\langle\phi_\ell|$.

In the case of degenerate singular values, a minimizer still recovers the subspace spanned by the singular functions sharing the same singular value. While this characterization remains true for any positive weights, we empirically found that the uniform weight $\mathbf{w} = (\frac{1}{L}, \dots, \frac{1}{L})$ works well. We remark in passing that the idea of nesting was first used in [25] to find the SVD of a special kernel, where the focus was on establishing a theoretical framework for structured representations.

Training Neural Networks with Minibatch Samples In practice, we train a neural network with minibatch training. When combined with neural networks, we call the entire framework *NeuralSVD*. Since the objective is in the form of summation of inner products, which are in turn integrals (or expectations), it allows a straightforward unbiased gradient estimator, we can use any off-the-shelf stochastic optimization method with minibatch to solve the optimization problem. Given a minibatch of size B, we can compute the minibatch objective by matrix operations, and the complexity is $O(B^2L + BL^2)$.

4 Experiment: Solving Time-Independent Schrödinger Equation

We demonstrate the potential power of our method in solving PDEs for scientific computing. In particular, following SpIN [14], we consider finding the first 9 eigenstates and energies of a hydrogen atom confined over a 2D plane by solving the time-independent Schrödinger equation $H\psi(\mathbf{x}) = E\psi(\mathbf{x})$. By ignoring constants, we can simplify it to the eigenvalue problem $(\nabla^2 + \frac{1}{\|\mathbf{x}\|_2})\psi(\mathbf{x}) = \lambda\psi(\mathbf{x})$ for $\mathbf{x} \in \mathbb{R}^2$. Here, the leading eigenvalues λ and eigenstates correspond to the ground states and energies E of the actual system due to the sign flipping. In this particular example, the closed-form solution is well-known [26]. Each eigenstate is parameterized by a pair of integers (n, ℓ) for $n \geq 0$ and $-n \leq \ell \leq n$, where the (negative) eigenenergy is $\lambda_{n,\ell} := \frac{1}{4}(n + \frac{1}{2})^{-2}$. Note that for each $n \geq 0$, there exist $2n + 1$ degenerate states that have the same energy.

Training Setup We adopted the training setup of [14] with some variations. We chose a sampling distribution $p(x)$ supported over a bounded box $[-D, D]^2$ for D sufficiently large; $D = 50$ was used as same as [14]. We note that this choice was apparently made based on prior knowledge that the first 9 eigenfunctions almost vanish outside the box $[-50, 50]^2$. To estimate higher modes that may span over a larger region, one would need to enlarge the box. Following [14], we further multiplied the factor $\prod_{i=1}^{d}(\sqrt{2D^2 - x_i^2} - D)$ to the network output, so that the output vanishes at the boundary of the box. We specifically used a two-dimensional Gaussian distribution $\mathcal{N}(0, 16^2\mathsf{I}_2)$ truncated over $[-50, 50]^2$ as a sampling distribution, to emphasize the importance of the approximation around the origin than the boundary of the box.

For both SpIN and NeuralSVD, we used 9 disjoint three-layer MLP with 128 hidden units to learn the first $L = 9$ eigenfunctions, and trained for 5×10^5 iterations with batch size 128. For the nonlinear activation function, we used the softplus activation $f(x) = \log(1 + e^x)$ following the implementation of [14]. For SpIN, we used the RMSProp optimizer [8] with learning rate 10^{-4}. For NeuralSVD, we used the RMSProp optimizer with learning rate 10^{-3} and the cosine learning rate schedule [10]. During the evaluation, we applied the exponential moving average (over the model parameters) with a decay rate of 0.995 to NeuralSVD for smoother results.

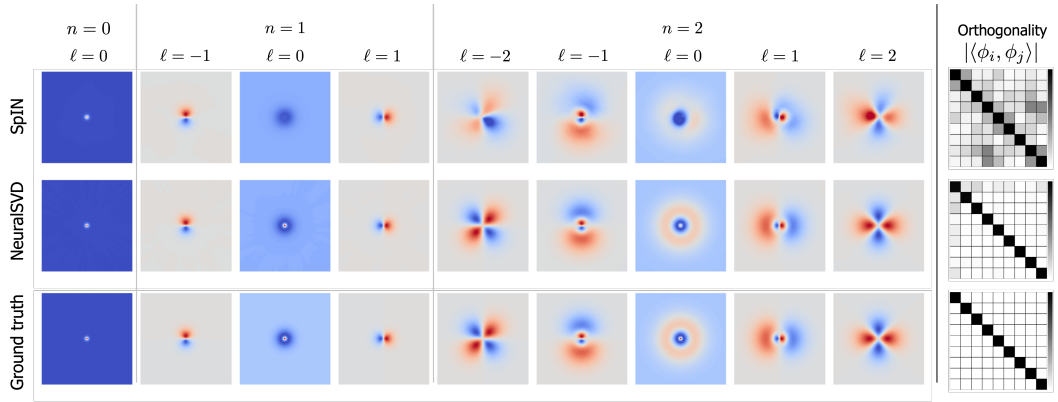

Figure 1: (Left) Visualization of the first 9 eigenfunctions $\phi_1, \ldots, \phi_9$ of the 2D hydrogen atom. The first and second rows present the learned eigenfunctions by SpIN and NeuralSVD, respectively. The learned functions are aligned by an orthogonal transformation within each degenerate subspace to compare with the ground truths in the third row. (Right) Visualization of the orthogonality of the learned eigenfunctions.

Results Fig. 1 shows the learned eigenfunctions from SpIN (with decay parameter $\beta = 0.01$ as suggested) and NeuralSVD. For comparison, we present the true eigenfunctions with a choice of canonical directions to plot the degenerate subspaces (third row). Note that SpIN does not match even after the rotation in several modes, e.g., $(n, \ell) = (1, 0), (2, -2), (2, 0), (2, 1)$. Further, the learned functions (before rotation) are not orthogonal as visualized in the right panel. In contrast, NeuralSVD can reliably match the correct eigenfunctions, with almost perfect orthogonality.

References

[1] Harry Andrews and C Patterson. Singular value decompositions and digital image processing. *IEEE Trans. Acoust. Speech Signal Process.*, 24(1):26–53, 1976.

[2] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6):1373–1396, 2003.

[3] Yoshua Bengio, Pascal Vincent, Jean-François Paiement, Olivier Delalleau, Marie Ouimet, and Nicolas Le Roux. *Spectral clustering and kernel PCA are learning eigenfunctions*, volume 1239. CIRANO, 2003.

[4] Avrim Blum, John Hopcroft, and Ravindran Kannan. *Foundations of data science*. Cambridge University Press, 2020.

[5] Michael AA Cox and Trevor F Cox. Multidimensional scaling. In *Handbook of data visualization*, pages 315–347. Springer, 2008.

[6] Zhijie Deng, Jiaxin Shi, and Jun Zhu. NeuralEF: Deconstructing kernels by deep neural networks. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proc. Int. Conf. Mach. Learn.*, volume 162 of *Proceedings of Machine Learning Research*, pages 4976–4992. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/deng22b.html`.

[7] Yoav Goldberg and Omer Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[8] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. , 2012.

[9] Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse Process.*, 25(2-3):259–284, 1998.

[10] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[11] Ivan Markovsky. *Low rank approximation: algorithms, implementation, applications*, volume 906. Springer, 2012.

[12] Tomer Michaeli, Weiran Wang, and Karen Livescu. Nonparametric canonical correlation analysis. In *Proc. Int. Conf. Mach. Learn.*, pages 1967–1976, 2016.

[13] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Adv. Neural Inf. Proc. Syst.*, volume 14, pages 849–856, 2001.

[14] David Pfau, Stig Petersen, Ashish Agarwal, David GT Barrett, and Kimberly L Stachenfeld. Spectral inference networks: Unifying deep and spectral learning. In *Int. Conf. Learn. Repr.*, 2018.

[15] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[16] Meyer Scetbon, Michael Elad, and Peyman Milanfar. Deep k-svd denoising. *IEEE Trans. Image Proc.*, 30:5944–5955, 2021.

[17] 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.

[18] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10(5):1299–1319, 1998.

[19] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.

[20] Henning Sprekeler. On the relation of slow feature analysis and Laplacian eigenmaps. *Neural Comput.*, 23(12):3287–3302, 2011.

[21] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[22] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *J. Cogn. Neurosci.*, 3(1):71–86, 1991.

[23] Lichen Wang, Jiaxiang Wu, Shao-Lun Huang, Lizhong Zheng, Xiangxiang Xu, Lin Zhang, and Junzhou Huang. An Efficient Approach to Informative Feature Extraction from Multimodal Data. In *Proc. AAAI Conf. Artif. Int.*, volume 33, pages 5281–5288, July 2019. doi: 10.1609/aaai.v33i01.33015281.

[24] Laurenz Wiskott and Terrence J Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Comput.*, 14(4):715–770, 2002.

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

[26] X L Yang, S H Guo, F T Chan, K W Wong, and W Y Ching. Analytic solution of a two-dimensional hydrogen atom. I. Nonrelativistic theory. *Phys. Rev. A*, 43(3):1186–1196, February 1991. ISSN 1050-2947. doi: 10.1103/physreva.43.1186.