Revisiting **Orbital Minimization Method** for **Neural Operator Decomposition**

Jongha (Jon) Ryu
Samuel Zhou
Gregory W. Wornell

MIT EECS

NEURAL INFORMATION PROCESSING SYSTEMS 2025

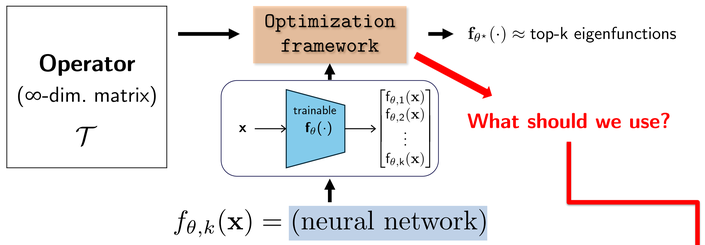
Problem: Operator Spectral Decomposition

Given a target positive semidefinite operator
find the top-K eigenvalues/functions

$$\mathcal{T} = \sum_{i=1}^{\infty} \lambda_i |\psi_i\rangle\langle\psi_i| \quad \xrightarrow{?} \quad \begin{matrix}(\lambda_i, |\psi_i\rangle) \\ \mathcal{T}|\psi_i\rangle = \lambda_i|\psi_i\rangle \\ i = 1, \ldots, K\end{matrix}$$

Applications (operator)
- Quantum chemistry (Hamiltonian)
- Representation learning (conditional expectation operator)
- Graph learning (graph Laplacians)
- ...

Parametric Spectral Decomposition

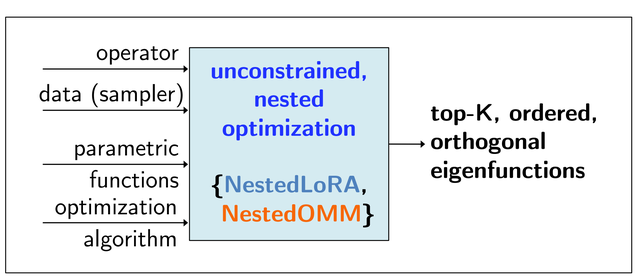


Classical approaches use very limited form of trial functions

- Finite element methods $f_{\theta,k}(\mathbf{x}) = \sum_{i=1}^{N} v_{k,i}\delta(\mathbf{x} - \mathbf{x}_i)$
- Galerkin type methods $f_{\theta,k}(\mathbf{x}) = \mathbf{a}_k^\intercal \zeta_{1:M}(\mathbf{x})$

Using neural networks can be **much more scalable** with enough inductive bias (proven to be powerful in *quantum chemistry*)

Our Approach: Nested Optimization with Unconstrained Variational Principle



- **Compact operators:** NestedLoRA (Ryu et al., ICML 2024)
- **PSD operators:** NestedOMM (this paper!)

What's Orbital Minimization Method (OMM)?

$$\mathsf{A} = \sum_{i=1}^{d} \lambda_i \mathbf{w}_i \mathbf{w}_i^\intercal, \qquad \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d \geq 0$$

$$\max_{\mathsf{V}\in\mathbb{R}^{d\times k}:\, \mathsf{V}^\intercal\mathsf{V}=\mathsf{I}_k} \mathrm{tr}(\mathsf{V}^\intercal\mathsf{A}\mathsf{V}) \qquad \text{standard Rayleigh quotient maximization}$$

$$\max_{\mathsf{V}\in\mathbb{R}^{d\times k}} \mathrm{tr}\left((\mathsf{V}^\intercal\mathsf{V})^{-1}\mathsf{V}^\intercal\mathsf{A}\mathsf{V}\right) \qquad \text{unconstrained version}$$

$$\min_{\mathsf{V}\in\mathbb{R}^{d\times k}} \mathcal{L}_{\mathrm{omm}}(\mathsf{V}), \quad \text{where } \mathcal{L}_{\mathrm{omm}}(\mathsf{V}) \triangleq -\mathrm{tr}((2\mathsf{I}_k - \mathsf{V}^\intercal\mathsf{V})\mathsf{V}^\intercal\mathsf{A}\mathsf{V})$$

All characterize the top-k eigensubspace as a global optima
(OMM was proposed by Mauri et al., (1993) in density functional theory)

Q1. Where does it come from?

(Original: ad-hoc Neumann series expansion)
We give a **simple yet intuitive** derivation:

$$\mathcal{L}_{\mathrm{omm}}(\mathsf{V}) = \mathrm{tr}((\mathsf{I}_d - \mathsf{V}\mathsf{V}^\intercal)^2\mathsf{A}) - \mathrm{tr}(\mathsf{A})$$
$$\mathcal{L}^{(p)}_{\mathrm{omm}}(\mathsf{V}) \triangleq \mathrm{tr}((\mathsf{I}_d - \mathsf{V}\mathsf{V}^\intercal)^{2p}\mathsf{A}) - \mathrm{tr}(\mathsf{A})$$

Q2. How can we show the consistency?

We provide an **elementary linear-algebraic proof**

Key observation: OMM only depends on the PSD matrix $\mathsf{V}\mathsf{V}^\intercal$

NestedOMM: OMM with Nesting

We can apply the nesting technique (Ryu et al., ICML 2024) to learn **ordered** eigenvectors → NestedOMM

- **Joint nesting:** update $\mathsf{V}_{1:k}$ by $\sum_{i=1}^{k} w_i \mathcal{L}_{\mathrm{omm}}(\mathsf{V}_{1:i})$
- **Sequential nesting:** update $\mathbf{v}_i$ using gradient of $\mathcal{L}_{\mathrm{omm}}(\mathsf{V}_{1:i})$, for each i

Special Case: Streaming PCA

PCA: $\mathsf{A} = \mathbb{E}_{p(\mathbf{x})}[\mathbf{x}\mathbf{x}^\mathrm{T}]$ / streaming PCA: $\mathsf{A}_t = \frac{1}{B}\sum_{b=1}^{B} \mathbf{x}_b^{(t)} \left(\mathbf{x}_b^{(t)}\right)^\mathrm{T}$

- OMM with sequential nesting (+ gradient descent)

$$\mathbf{v}_i^{(t+1)} \leftarrow \mathbf{v}_i^{(t)} + \eta_t \left\{ \left(\mathsf{I} - \mathsf{V}_{1:i}^{(t)}(\mathsf{V}_{1:i}^{(t)})^\intercal\right)\mathsf{A}_t\mathbf{v}_i^{(t)} + \mathsf{A}_t\left(\mathsf{I} - \mathsf{V}_{1:i}^{(t)}(\mathsf{V}_{1:i}^{(t)})^\intercal\right)\mathbf{v}_i^{(t)} \right\}$$

- cf. Generalized Hebbian algorithm (a.k.a. Sanger's algorithm)

$$\mathbf{v}_i^{(t+1)} \leftarrow \mathbf{v}_i^{(t)} + \eta_t \left(\mathsf{I} - \mathsf{V}_{1:i}^{(t)}(\mathsf{V}_{1:i}^{(t)})^\intercal\right)\mathsf{A}_t\mathbf{v}_i^{(t)}$$

Comparison with Low-Rank Approximation (LoRA)

$$\mathcal{L}_{\mathrm{omm}}(\mathsf{V}) = \mathrm{tr}((\mathsf{I}_d - \mathsf{V}\mathsf{V}^\intercal)^2\mathsf{A}) - \mathrm{tr}(\mathsf{A})$$
$$\mathsf{V}^\star(\mathsf{V}^\star)^\intercal = \mathsf{W}_{1:k}\mathsf{W}_{1:k}^\intercal$$

Only applicable for EVD of PSD matrices
Can outperform LoRA when A is sparse

$$\mathcal{L}_{\mathrm{lora}}(\mathsf{V}) = \|\mathsf{A} - \mathsf{V}\mathsf{V}^\intercal\|_\mathsf{F}^2 - \|\mathsf{A}\|_\mathsf{F}^2$$
$$\mathsf{V}^\star(\mathsf{V}^\star)^\intercal = \mathsf{W}_{1:k}\Lambda_{1:k}\mathsf{W}_{1:k}^\intercal$$

Applicable for SVD of any matrices

From Matrix to Operator

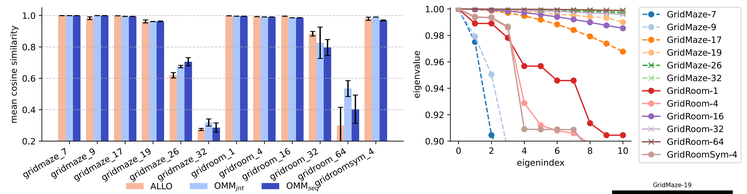
$$\mathcal{T}\colon L^2_\rho(\mathcal{X}) \to L^2_\rho(\mathcal{X}) \qquad L^2_\rho(\mathcal{X}) \triangleq \left\{ f\colon \mathcal{X}\to\mathbb{R} \;\middle|\; \int_\mathcal{X} f(\mathbf{x})^2\rho(d\mathbf{x}) < \infty \right\}$$
$$\langle f_0, f_1\rangle_\rho \triangleq \int_\mathcal{X} f_0(\mathbf{x})f_1(\mathbf{x})\rho(d\mathbf{x})$$

REPLACE

overlap matrix	$\mathsf{V}^\intercal\mathsf{V}$	w/	$\mathsf{M}_\rho[\mathbf{f}] \triangleq \int \mathbf{f}(\mathbf{x})\mathbf{f}(\mathbf{x})^\intercal\rho(d\mathbf{x})$
projected matrix	$\mathsf{V}^\intercal\mathsf{A}\mathsf{V}$		$\mathsf{M}_\rho[\mathbf{f},\mathcal{T}\mathbf{f}] \triangleq \int \mathbf{f}(\mathbf{x})(\mathcal{T}\mathbf{f})(\mathbf{x})^\intercal\rho(d\mathbf{x})$

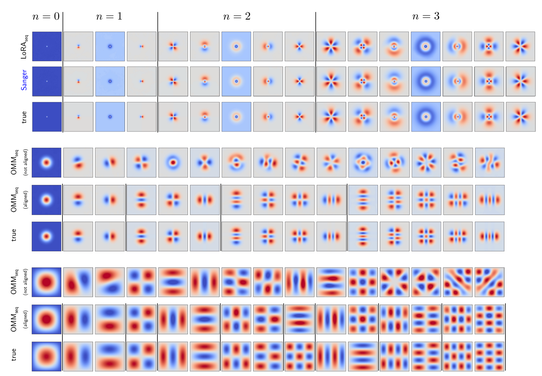
Experiments

(1) Representation learning for reinforcement learning



Compared to ALLO, OMM does not need any hyperparameter tuning and robust to small spectral gap!

(2) Solving time-independent Schrödinger equation



(1) Bounded spectrum
2d-confined hydrogen atom
(2) Unbounded spectrum can be handled by the **inverse-operator trick**!
(2-1) 2d harmonic oscillator
(2-2) 2d infinite-well

(3) Linear-algebraic self-supervised representation learning

Method	With projector		DirectCLR (top-64 dim.)	
	Top-1	Top-5	Top-1	Top-5
OMM ($p=1$)	60.02	87.13	59.83	86.65
OMM$_{\mathrm{jnt}}$ ($p=1$)	61.30	85.22	59.92	85.13
OMM$_{\mathrm{seq}}$ ($p=1$)	59.91	87.02	52.96	81.22
OMM ($p=2$)	63.92	89.08	61.27	87.07
OMM ($p=1$) + OMM ($p=2$)	64.77	89.18	63.99	88.88
SimCLR [3]	66.50	89.28	N/A	N/A

Higher-order OMM helps learn better representation
Though not SOTA, motivates further investigation!