# Online PCA in Converging Self-consistent Field Equations

Xihan Li[1], Xiang Chen[2], Rasul Tutunov[3], Haitham Bou Ammar[3], Lei Wang[4], Jun Wang[1]

[1]University College London

[2]Huawei Noah's Ark Lab

[3]Huawei R&D U.K.

[4]Institute of Physics, Chinese Academy of Sciences

Speaker: Xihan Li

NeurIPS 2023

# Problem Setting

Eigen Decomposition

An $n \times n$ matrix → $A$

A scalar (eigenvalue) → $\lambda$

$$Av = \lambda v$$

Eigen decomposition: given matrix $A$, find a vector $v$ (eigenvector) and a scalar $\lambda$ (eigenvalue) to satisfy the above equation

E.g., given $A = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$, we can do standard eigen decomposition to get a vector $v = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and a scalar $\lambda = 3$ so that

$$\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 3 \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

# Problem Setting

What will happen if matrix $A$ is not directly given, but $A$ is a given function of $v$?

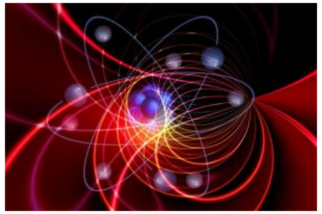
$F$ is given

$$A \doteq F(v)$$

An $n \times n$ matrix

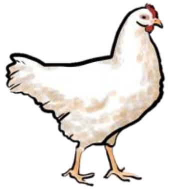
A scalar (eigenvalue)

$$\boxed{F(v)}v = \lambda v$$



Self-consistent Field Equation
(Important in Quantum Physics!)
$H|\Psi\rangle = E|\Psi\rangle$

Eigen decomposition cannot be directly applied anymore!



To obtain $v$, eigen decomposition needs $A \rightarrow A$ comes from $F(v) \rightarrow$ we need to obtain $v$

# Traditional Methods

Self-consistent Field method (fixed point iteration) for solving $F(v)v = \lambda v$

Assign an initial $v_0 \xrightarrow{F_0 = F(v_0)} F_0 \xrightarrow[\text{Eigen decomposition}]{F_0 v_1 = \lambda v_1} v_1 \longrightarrow F_1 \longrightarrow v_2 \longrightarrow$ … (until convergence)

Problem: easily fails to converge (**infinite oscillation** between two or more states)

$$\cdots \longrightarrow \underbrace{v' \longrightarrow F' \longrightarrow v'' \longrightarrow F'' \longrightarrow v'}_{\text{Repeat infinitely}} \longrightarrow \cdots$$

Two current main research directions:

1. Generate a better initial solution $v_0$
2. Mix $F_t$ with those in previous iterations $F_{t-1}, F_{t-2}, \ldots$ to stabilize the iteration

**We propose a third direction with the aid of machine learning techniques**

# Motivation

We find a connection between two very different problems in different fields

- Self-consistent Field Equation

An $n \times n$ matrix

A scalar (eigenvalue)

$$\boxed{F(v)}v = \lambda v$$

Infinite oscillation

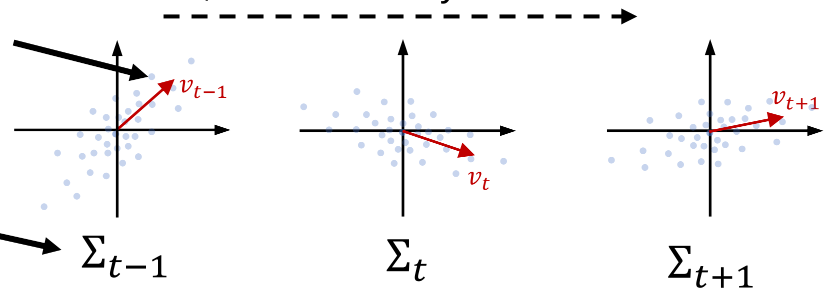
Task: finding $v$
Features:
1. Involves eigen decomposition
2. $F(v)$, the matrix to be decomposed, is not determined during the decomposition

- Online PCA

Time-variant data distribution in an online, non-stationary environment

Top principal component

Covariance matrix of the data distribution

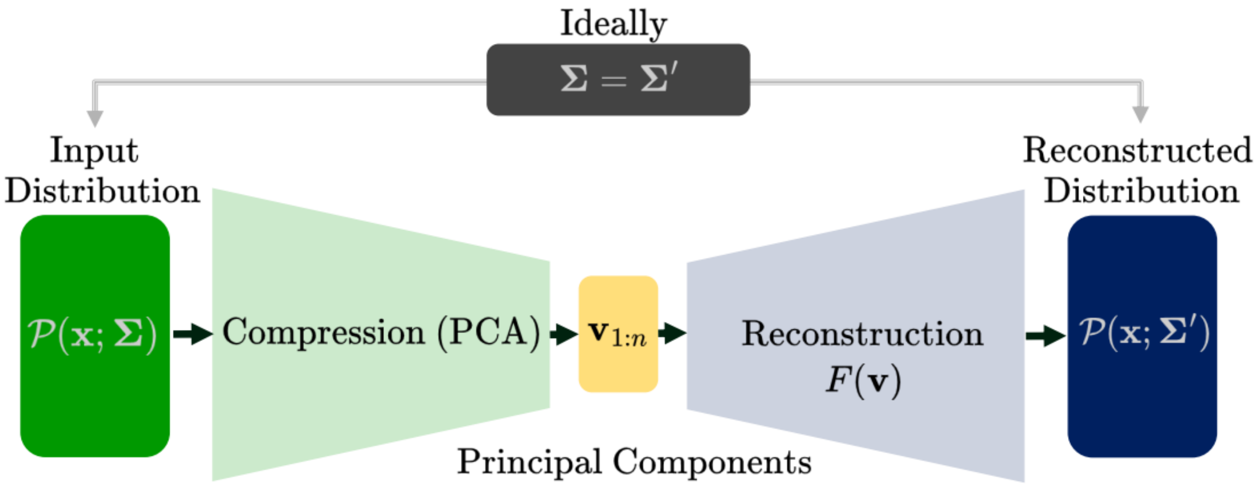



Incrementally Updated

Task: finding $v$
Features:
1. Involves eigen decomposition
2. $\Sigma_t$, the matrix to be decomposed, is not determined during the decomposition

**Can we use Online PCA to resolve the infinite oscillation issue of SCF equation solving?**

# Our Method




- $F(v)v = \lambda v$ is to say, if we have a matrix $\Sigma$, then
    1. Decompose $\Sigma$ to get its top eigenvector $v$
    2. Compute a new matrix $\Sigma' = F(v)$

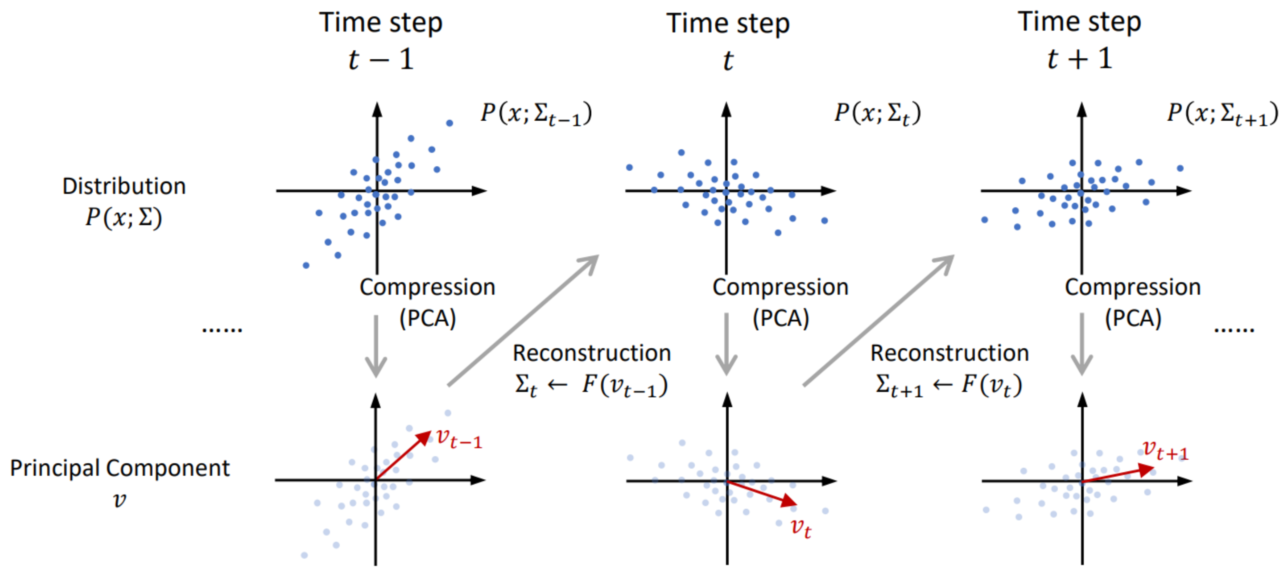
Then we will have $\Sigma' = \Sigma$

New interpretation:

"compress" $\Sigma$ with PCA to have $v$

"reconstruct" $\Sigma$ from $v$ with $F(\cdot)$

# Our Method



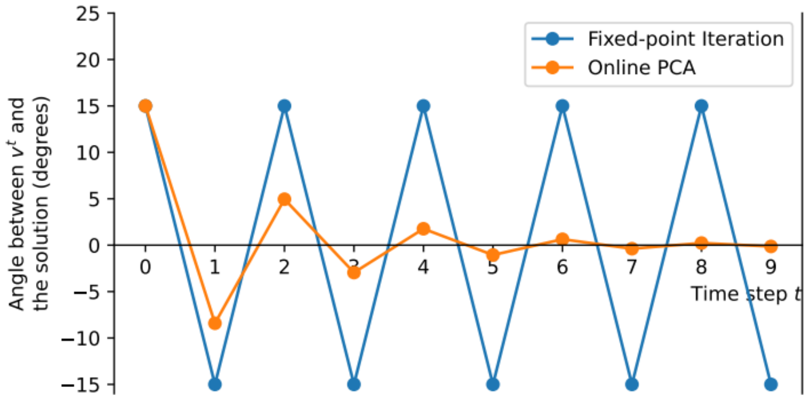

Then, the fixed-point iteration $v_0 \to F_0 \to v_1 \to F_1 \to \cdots$ can be regarded as

Compress (PCA) → reconstruct → compress (PCA) → reconstruct → …

We are continuously running PCA in a non-stationary environment!

Now we can apply Online PCA to **update $v$ incrementally** to avoid **infinite oscillation**

# Our Method



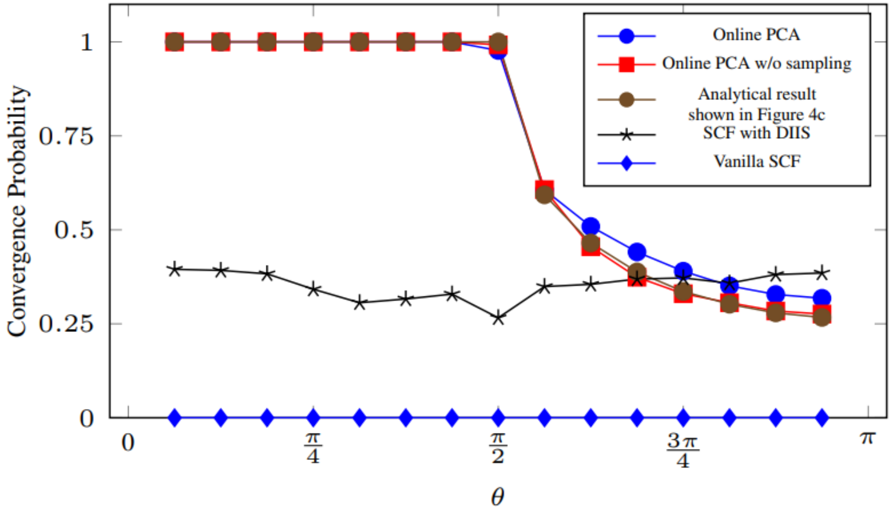

Then, the fixed-point iteration $v_0 \rightarrow F_0 \rightarrow v_1 \rightarrow F_1 \rightarrow \cdots$ can be regarded as

Compress (PCA) → reconstruct → compress (PCA) → reconstruct → …

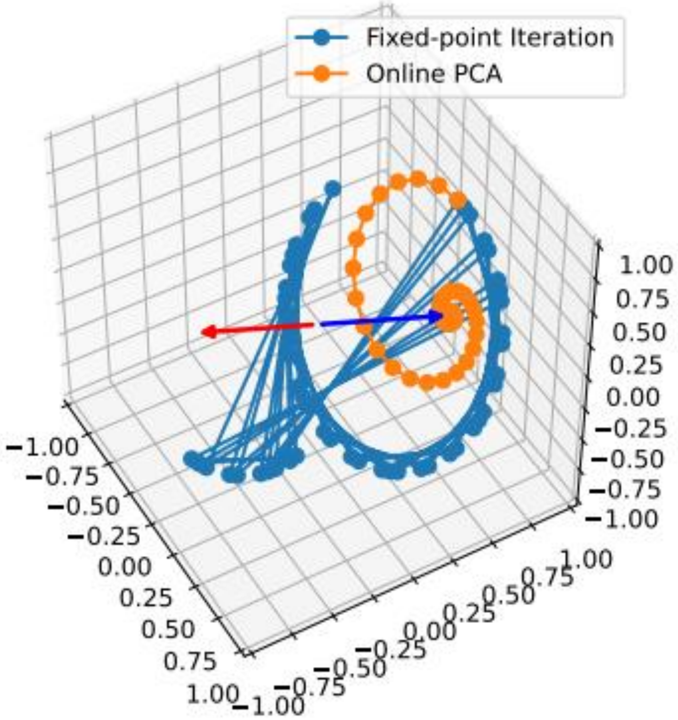
We are continuously running PCA in a non-stationary environment!

Now we can apply Online PCA to **update $v$ incrementally** to avoid **infinite oscillation**

# Experiment



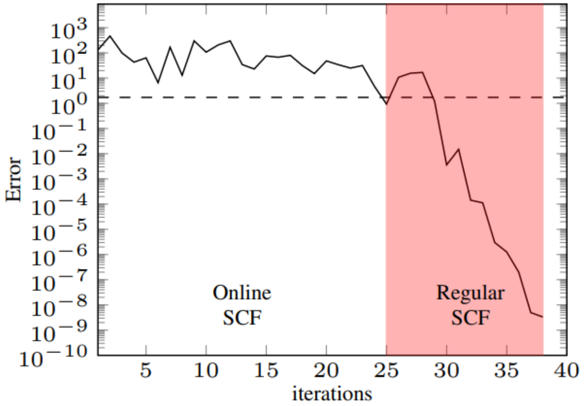

(a) Convergence ratio

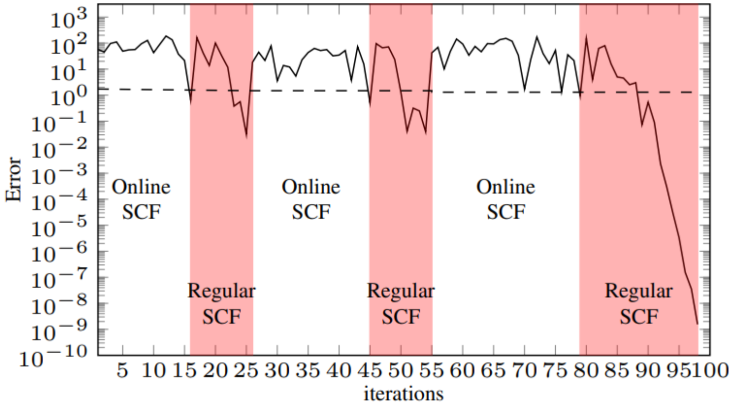



- Case study: solve $(Avv^{\top}A^{\top})v = \lambda v$
  - Vanilla fixed-point method: does not work at all (0% convergence ratio)
  - DIIS: ~40% convergence ratio
  - Online PCA: the top curves, half has 100% convergence ratio

# Experiment

| Methods | Hartree-Fock | | | DFT with B3LYP | | |
|---|---|---|---|---|---|---|
| | #(Nonconverged molecules) | | Average #(iterations) | #(Nonconverged molecules) | | Average #(iterations) |
| Regular SCF | 124 | (9.27%) | 25.49 | 407 | (30.42%) | 21.09 |
| Full Online SCF | 13 | (0.97%) | 584.68 | 217 | (16.22%) | 1835.24 |
| Adaptive Online SCF | 0 | (0%) | 42.97 | 0 | (0%) | 60.58 |




- For real-world SCF equations such as Hartree-Fock and DFT, our proposed method with adaptations (Online SCF) can also achieve high convergence ratio with a moderate increase of iterations.
- We also proposed an adaptive switching mechanism between online and regular mode, to balance efficiency and convergency.

# Thank you!

Xihan Li

Department of Computer Science, University College London

Email: xihan.li@cs.ucl.ac.uk

Webpage: https://snowkylin.github.io