ILLINOIS

An ensemble variational principle for excited state determination in VMC

Lucas K. Wagner

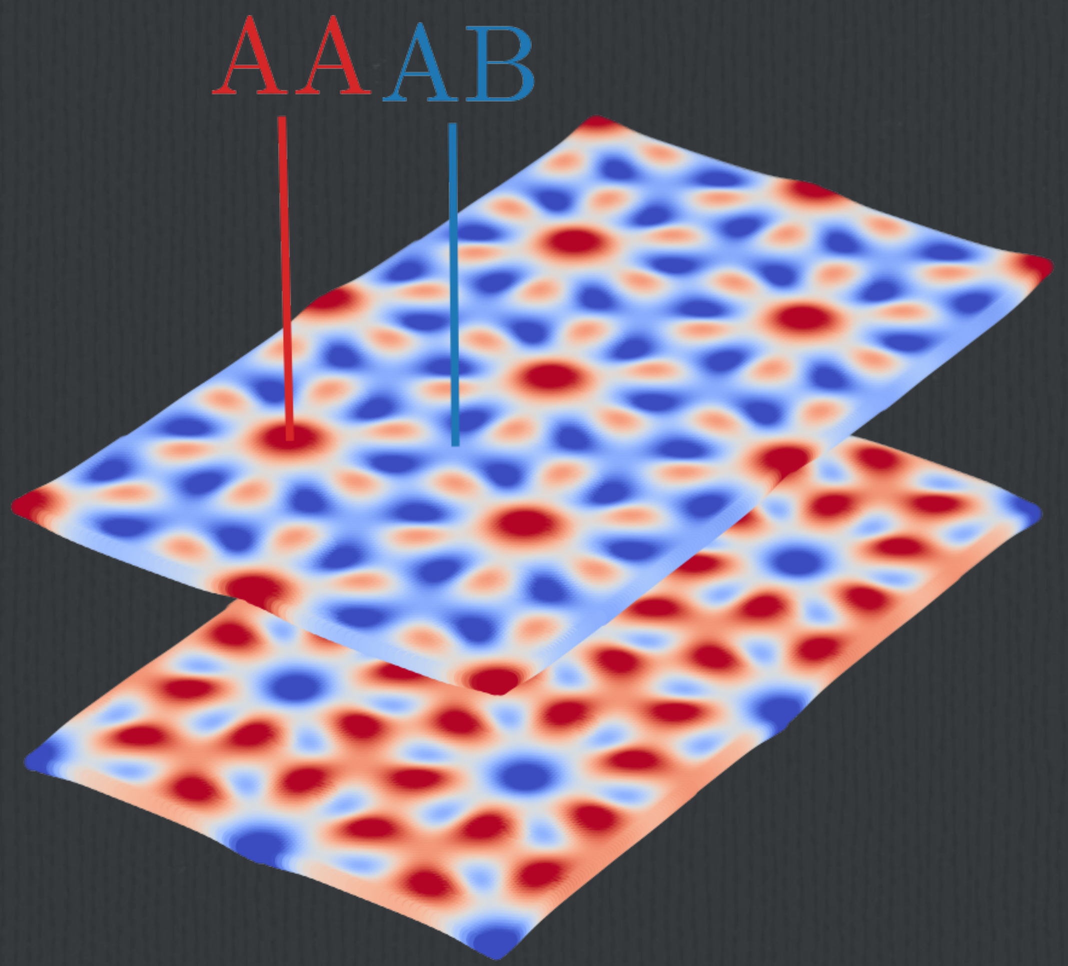
“How many spectra have you seen computed by QMC?”

U.S. DEPARTMENT OF ENERGY | Office of Science

NSF

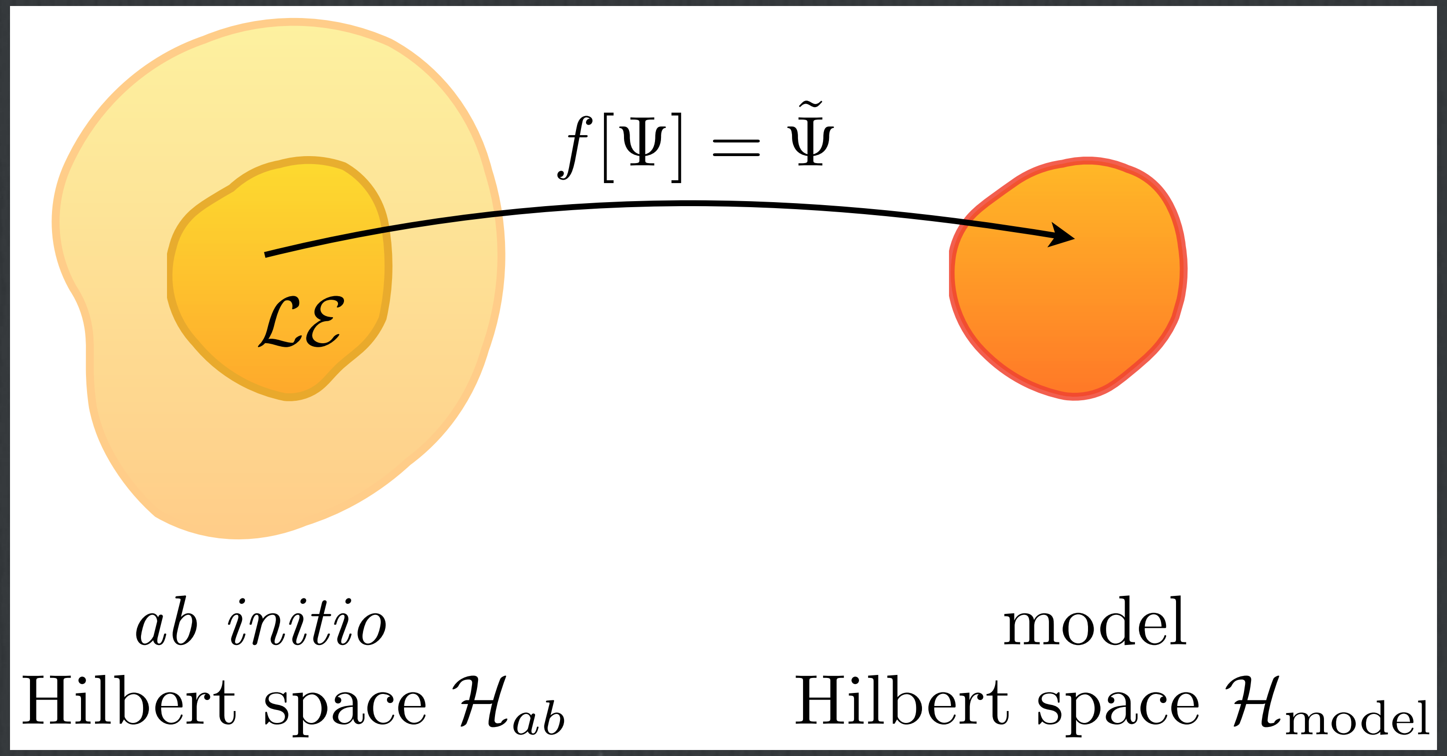
DOE BES DE-SC0020177.
U.S. NSF Award No. 1931258

Some of my other interests



Coarse-grained models for bilayer graphene

Krongchon et al.
PRB 108 235403


$f[\Psi] = \tilde{\Psi}$
$\mathcal{LE}$
ab initio Hilbert space $\mathcal{H}_{ab}$
model Hilbert space $\mathcal{H}_{\text{model}}$

Non-perturbative renormalization from many-body solutions

Chang, Joshi, Wagner
arXiv:2302.02899
Chang et al.
arXiv:2311.05987

```
❯ pip install pyqmc
```

Making QMC development easier

Wheeler et al.
J. Chem. Phys.
158 114801

Thanks

Work

Will Wheeler

Kevin Kleiner

Conversation

Eric Neuscamman

Claudia Filippi

Kieron Burke

Summary

$$O[\{\Psi_i\}] = \sum_i w_i E[\Psi_i] + \lambda \sum_{i<j} |S_{ij}|^2$$

$$w_i > w_j, \ \forall \ i < j$$

$$\lambda > \max_{i<j} \left[(E_j - E_i) \frac{w_i w_j}{w_i - w_j} \right]$$

$\mathcal{O}$ is minimized when the wave functions are the lowest N eigenstates.

Variational principle for ensembles of states.

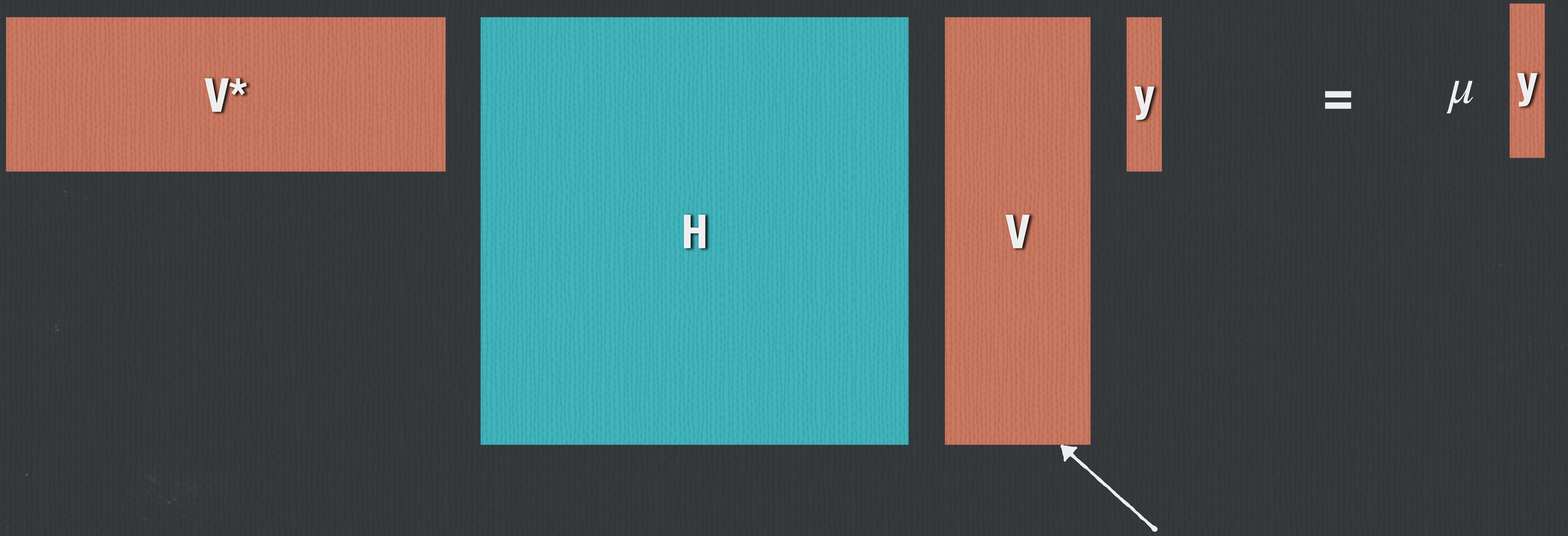
Leads to several optimization strategies.

arXiv:2312.00693

Context on excited states in quantum Monte Carlo

Rayleigh-Ritz methods

Rayleigh-Ritz



V can be optimized.

A Rayleigh-Ritz Example

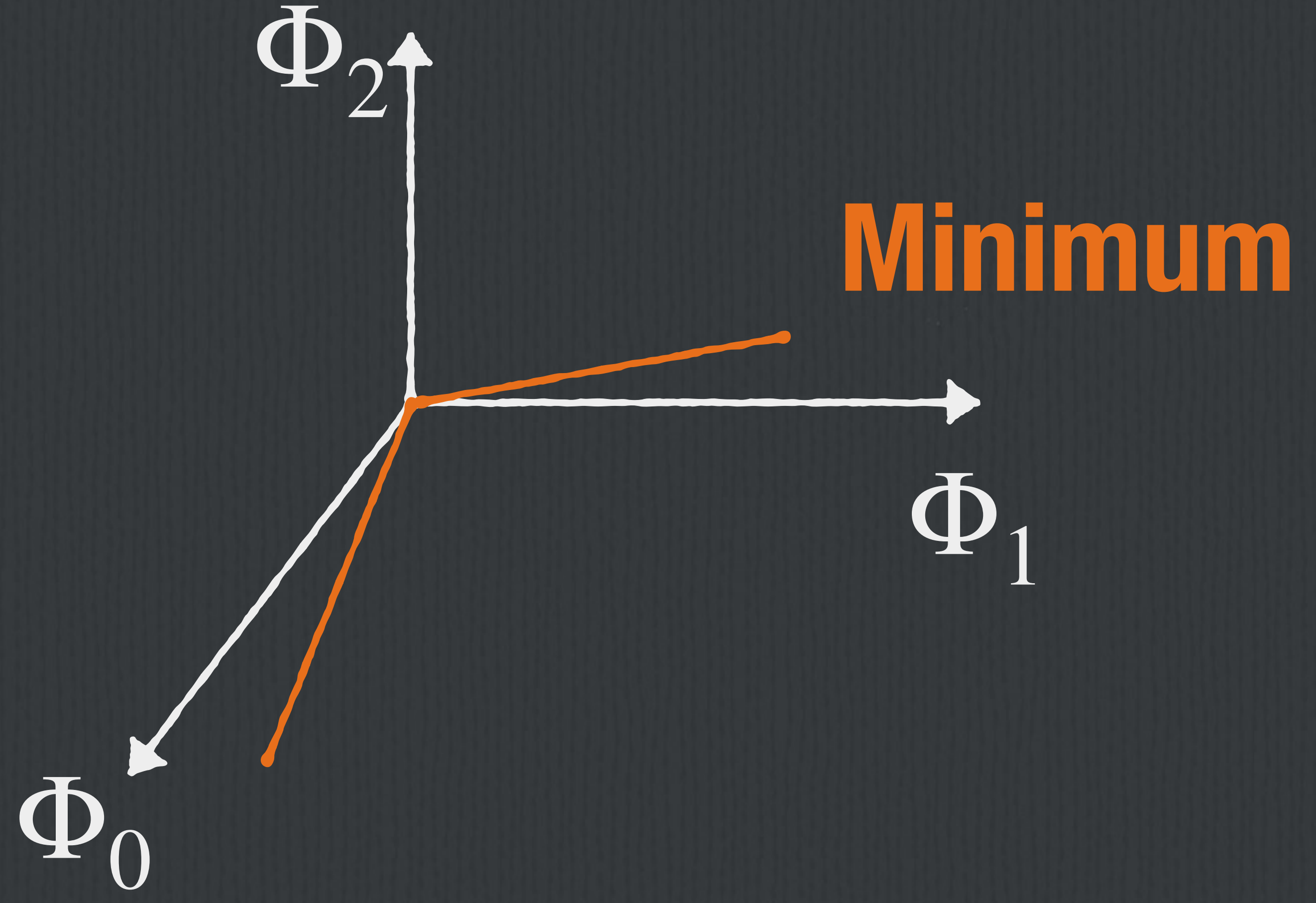
Find wave functions $\{\Psi_i\}$ to minimize

$$\mathcal{O}[\{\Psi_i\}] = \sum_i E[\Psi_i] + \lambda \sum_{i<j} \langle\, | \langle S_i | S_j \rangle |^2$$

This functional is invariant to any unitary transformation of Ψ_i.

Minimum is on the plane of the lowest N eigenstates.

The minimum is not the set of eigenstates Φ_i, except as $\lambda \to \infty$.



Rayleigh-Ritz:

$$\begin{array}{cccc} H & \Psi & & S & \Psi \\ \begin{bmatrix} \langle\Psi_0|H|\Psi_0\rangle & \langle\Psi_0|H|\Psi_1\rangle \\ \langle\Psi_1|H|\Psi_0\rangle & \langle\Psi_1|H|\Psi_1\rangle \end{bmatrix} & \begin{bmatrix} |\Psi_0\rangle \\ |\Psi_1\rangle \end{bmatrix} & = E_i & \begin{bmatrix} \langle\Psi_0|\Psi_0\rangle & \langle\Psi_0|\Psi_1\rangle \\ \langle\Psi_1|\Psi_0\rangle & \langle\Psi_1|\Psi_1\rangle \end{bmatrix} & \begin{bmatrix} |\Psi_0\rangle \\ |\Psi_1\rangle \end{bmatrix} \end{array}$$

Examples of Rayleigh-Ritz methods

Ceperley, Bernu: V is a set of projected wave functions

State-averaged method (Filippi): V is a set of multi-slater Jastrow

Pfau et al: V is a set of optimized NN states.

<u>Ceperley, Bernu</u> J. Chem. Phys 89 6316 (1988)
Filippi, Zaccheddu, Buda JCTC 5 2074 (2009)
Pfau et al. arxiv:2308.16848v1

Why I prefer to avoid Rayleigh-Ritz

1. $\langle \Psi_i | H | \Psi_j \rangle$ often has high variance

2. Most good parameters are nonlinear -> hard to superimpose states

3. Stochastically evaluated matrices result in biased diagonalization.

Note: all these can be overcome with some effort

Why do we want to avoid $\langle \Psi_i | H | \Psi_j \rangle$?

$$\int \frac{\Psi_i^* H \Psi_j}{\rho} \rho = \int \frac{\Psi_i^* \Psi_j}{\rho} \frac{H \Psi_j}{\Psi_j} \rho = \left\langle \underset{\text{Goes +/-}}{\frac{\Psi_i^* \Psi_j}{\rho}} \underset{\text{Is huge}}{\frac{H \Psi_j}{\Psi_j}} \right\rangle$$

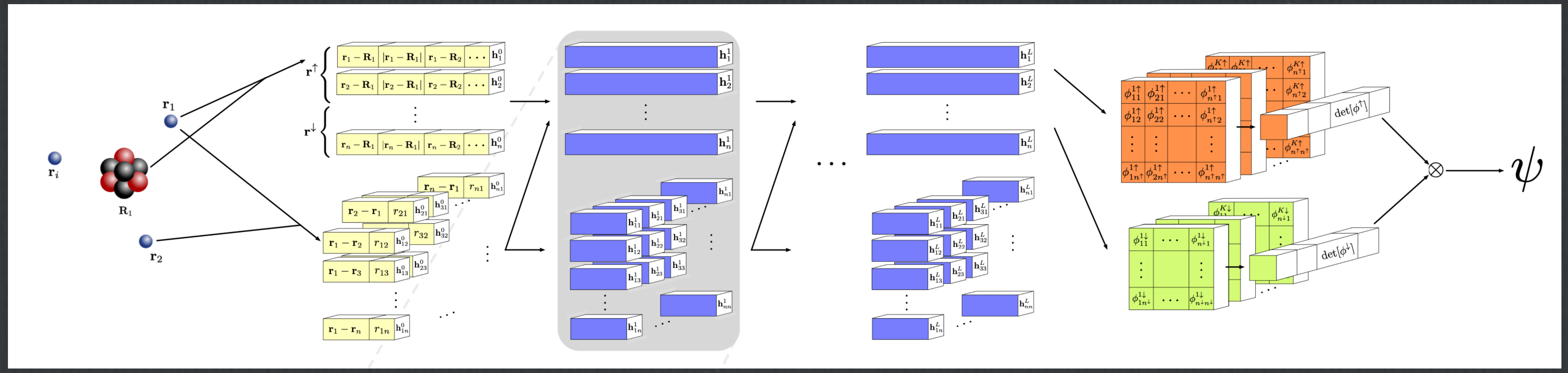
Variance can be 10-100 times larger than diagonal elements!

Goes +/-

Is huge

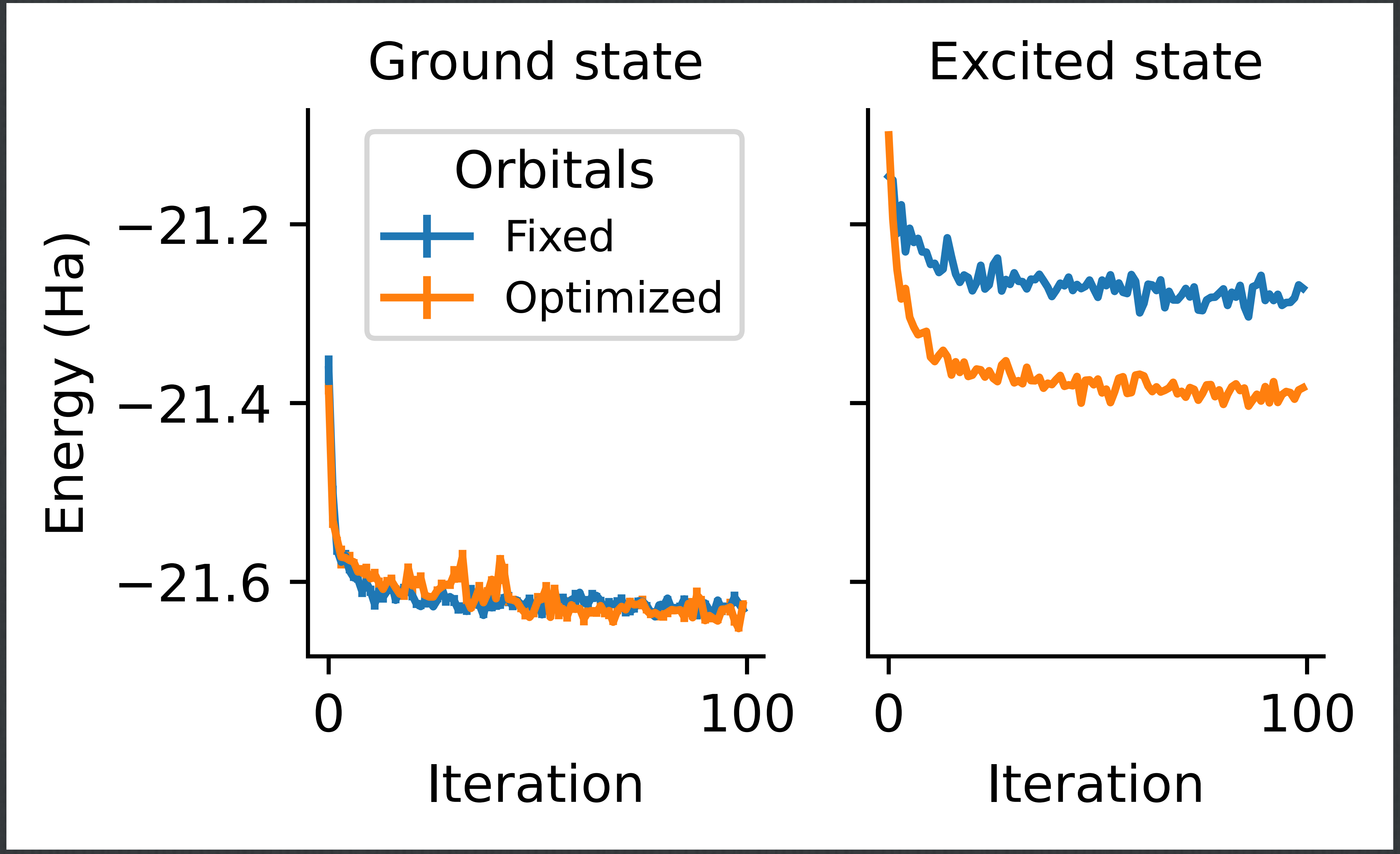
***note: It's possible we could do something similar to the linear method (recast into covariances), but no one has tried this as far as I know.**

Nonlinear parameters



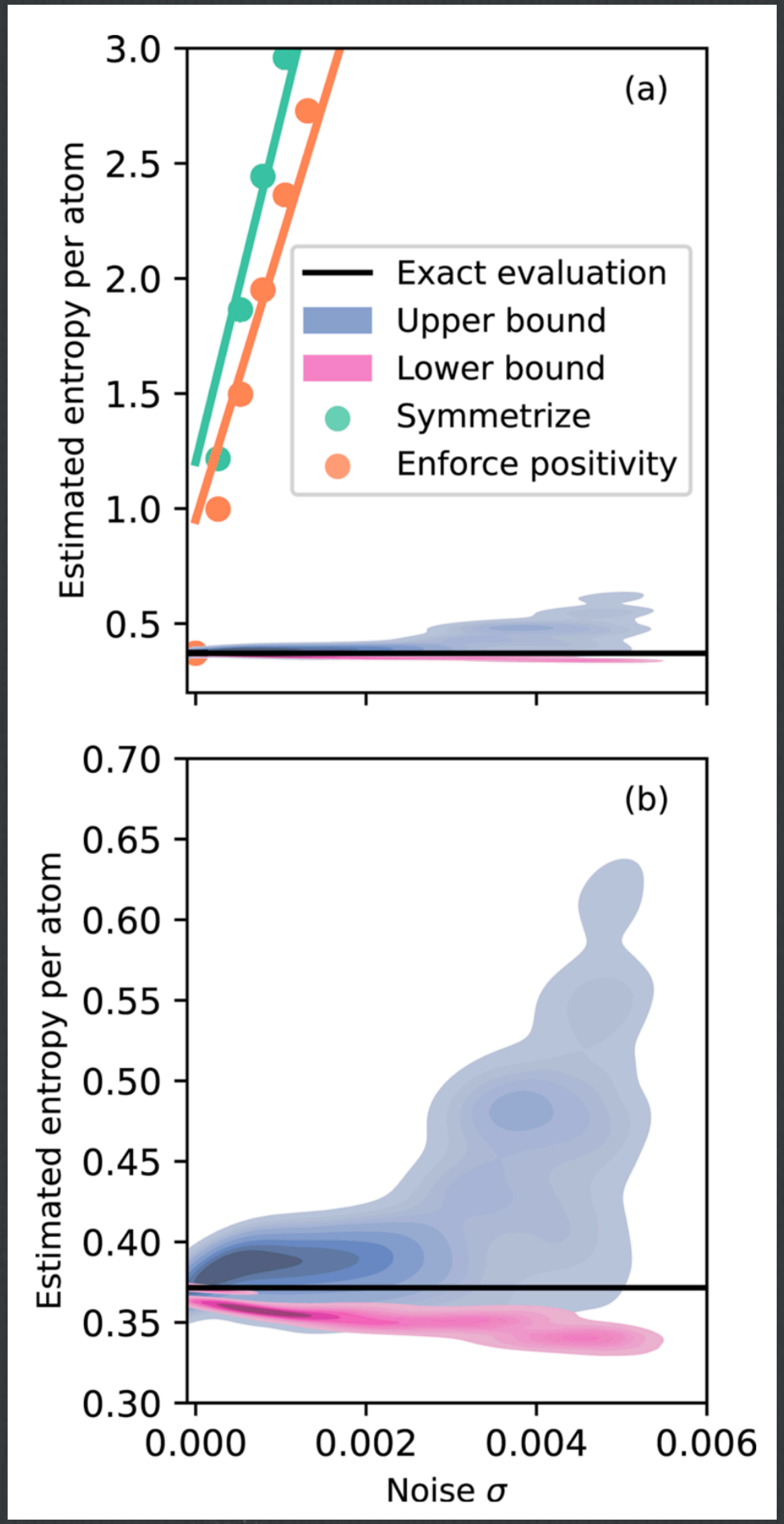
NN wave functions seem pretty good

Also exponents (Umrigar), pairing, Jastrow



Minimal wave functions can be much more accurate if the orbitals are optimized (case of CO molecule)

Diagonalization of random matrices



Bias due to nonlinear function.

Even for “small” noise, bias in the von Neumann entropy can be a factor of 5-6.

Can be regularized away sometimes but needs to be done on a case-by-case basis.

Yuan, Chang, Wagner. J. Chem. Phys. 157, 194101 (2022)

Methods that avoid construction and diagonalization of the Hamiltonian

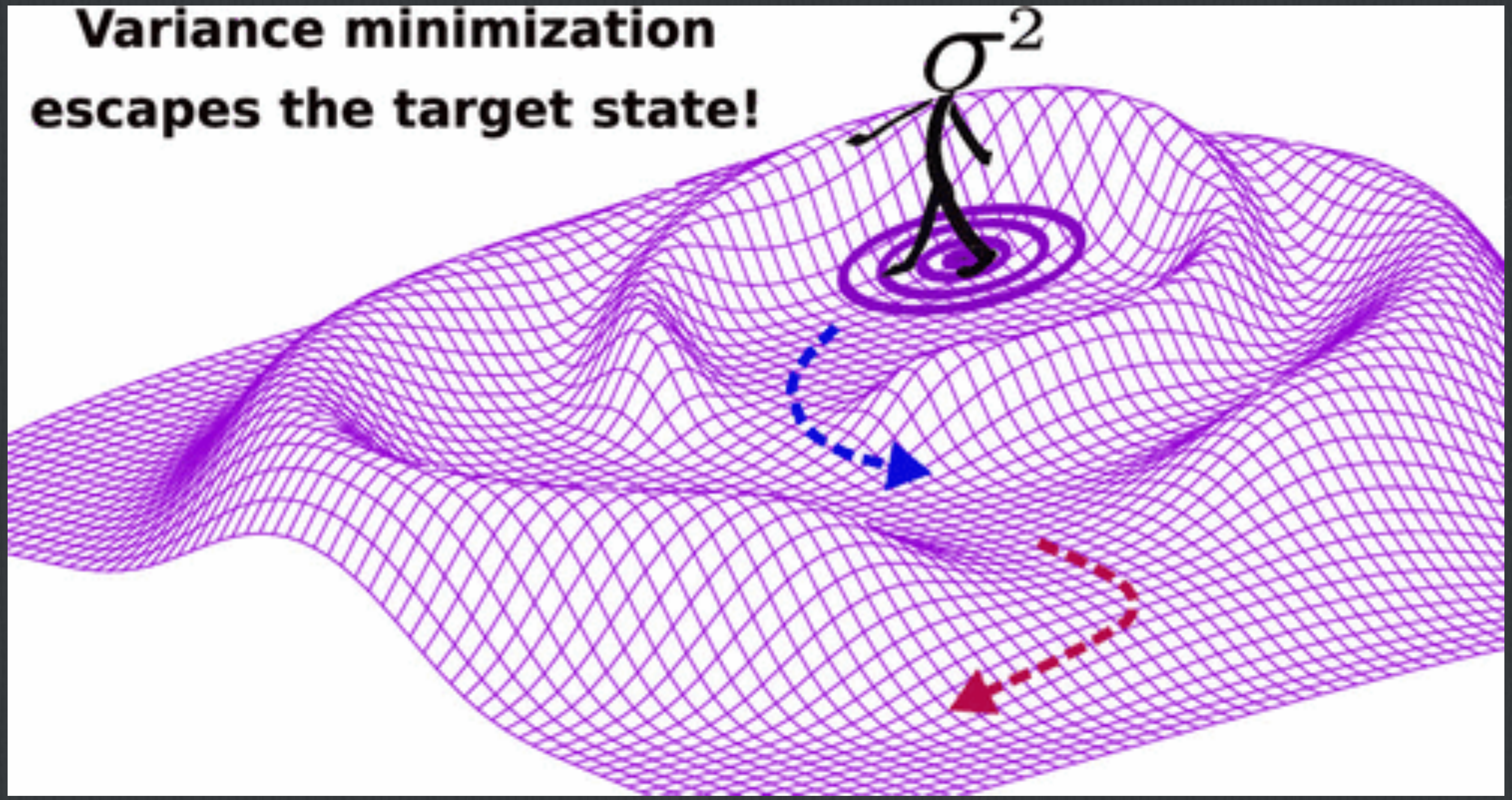
Targeted methods

$$\mathcal{O}[\Psi] = \frac{\omega - E}{(\omega - E)^2 + \sigma^2}$$

Choose ω carefully to have a minimum at an eigenstate.

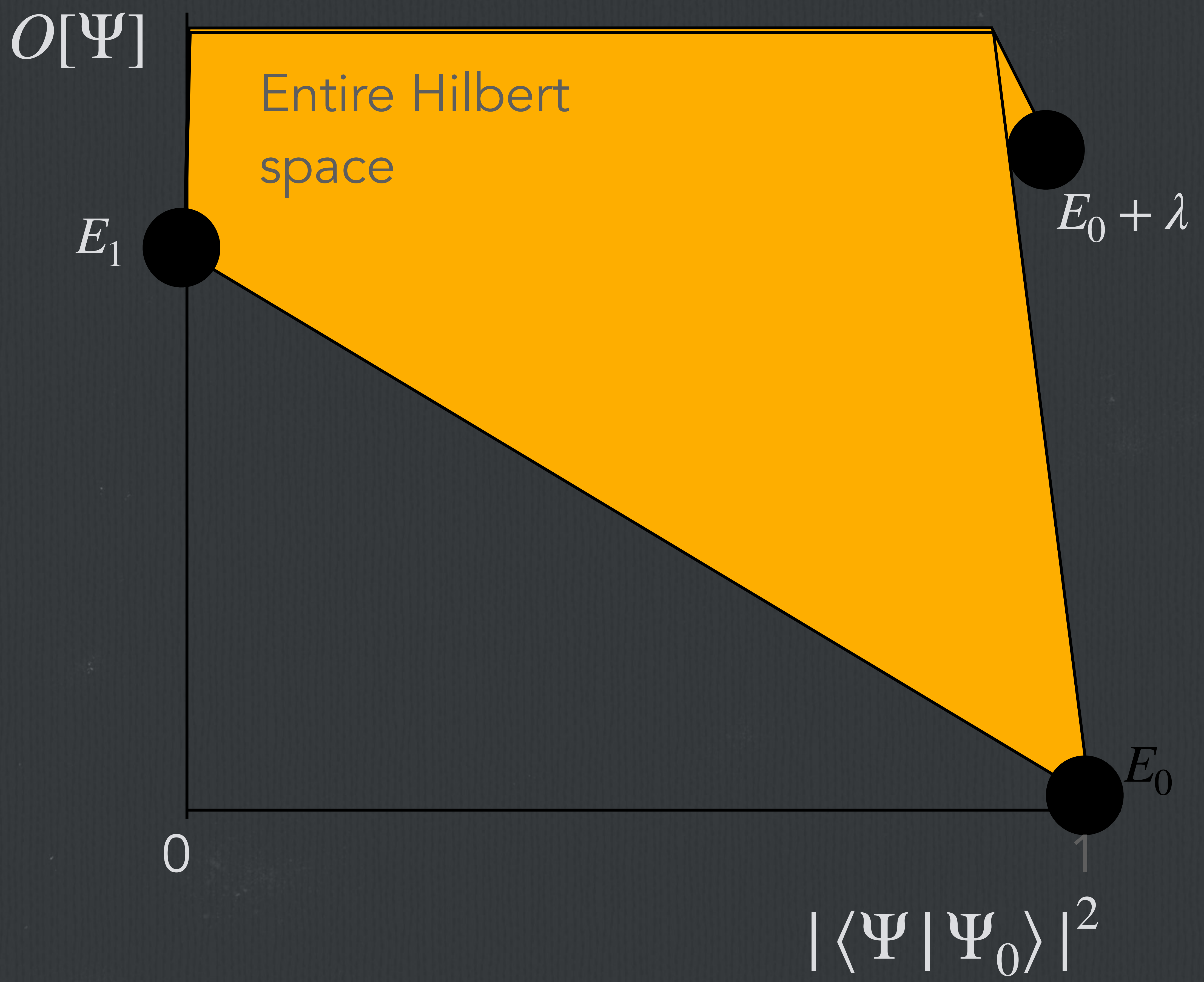
High variance

For approximate wave functions, can have multiple minima.



Flores, Neuscamman. J. Phys. Chem A 123 1487 (2019)
Cuzzocrea et al. JCTC 16 4203 (2020)

Ladder of states approach (Gram-Schmidt)



$$O[\Psi] = E[\Psi] + \lambda|\langle\Psi_0|\Psi\rangle|^2$$

$$\lambda > E_1 - E_0$$

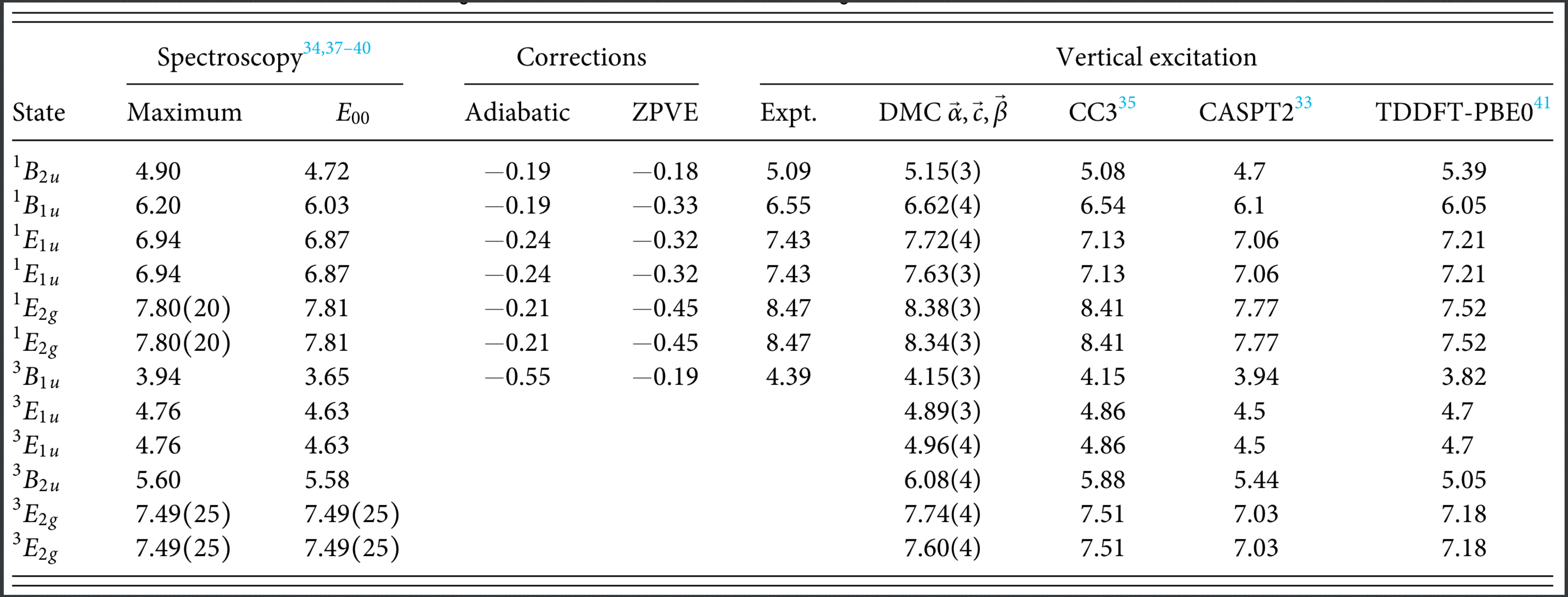
Does not run afoul of our requirements

No "empirical parameters".

However, propagation of error?

Higgot, Wang, Brierley Quantum 3, 156 (2019)
Pathak, Wagner. J. Chem. Phys. 154, 034101 (2021)

We are starting to get spectra (benzene)



	Spectroscopy[34,37–40]		Corrections		Vertical excitation				
State	Maximum	E_{00}	Adiabatic	ZPVE	Expt.	DMC $\vec{\alpha}, \vec{c}, \vec{\beta}$	CC3[35]	CASPT2[33]	TDDFT-PBE0[41]
$^1B_{2u}$	4.90	4.72	−0.19	−0.18	5.09	5.15(3)	5.08	4.7	5.39
$^1B_{1u}$	6.20	6.03	−0.19	−0.33	6.55	6.62(4)	6.54	6.1	6.05
$^1E_{1u}$	6.94	6.87	−0.24	−0.32	7.43	7.72(4)	7.13	7.06	7.21
$^1E_{1u}$	6.94	6.87	−0.24	−0.32	7.43	7.63(3)	7.13	7.06	7.21
$^1E_{2g}$	7.80(20)	7.81	−0.21	−0.45	8.47	8.38(3)	8.41	7.77	7.52
$^1E_{2g}$	7.80(20)	7.81	−0.21	−0.45	8.47	8.34(3)	8.41	7.77	7.52
$^3B_{1u}$	3.94	3.65	−0.55	−0.19	4.39	4.15(3)	4.15	3.94	3.82
$^3E_{1u}$	4.76	4.63				4.89(3)	4.86	4.5	4.7
$^3E_{1u}$	4.76	4.63				4.96(4)	4.86	4.5	4.7
$^3B_{2u}$	5.60	5.58				6.08(4)	5.88	5.44	5.05
$^3E_{2g}$	7.49(25)	7.49(25)				7.74(4)	7.51	7.03	7.18
$^3E_{2g}$	7.49(25)	7.49(25)				7.60(4)	7.51	7.03	7.18

All-at-once Gram-Schmidt

$$\Delta \vec{p}_i = \tau S^{-1} \left(\nabla E[\Psi_i] + \sum_{j<i} |\langle S_i | S_j \rangle|^2 \right)$$

Note that each state is only orthogonalized to lower ones.

Fixed point is the eigenstates.

BUT: No objective function.

Entwistle, Schätzle, Erdman, Hermann, Noé. Nat. Comm. 14 274 (2023)

The meat of the talk: An ensemble objective function

Why it matters to have a variational principle

Much of what we do is to compare ansätze.

For approximate wave functions, there is no variational upper bound on the excited state energies.*

A variational upper bound for an ensemble of states allows us to compare approximate wave functions without reference to experiment, in the same way as we do for ground states.

* Symmetry..

Summary

Define

$$O[\{\Psi_i\}] = \sum_i w_i E[\Psi_i] + \lambda \sum_{i<j} |S_{ij}|^2$$

$$S_{ij} = \langle \Psi_i | \Psi_j \rangle$$

If

$$w_i > w_j, \quad \forall \quad i < j$$

$$\lambda > \max_{i<j} \left[(E_j - E_i) \frac{w_i w_j}{w_i - w_j} \right]$$

Then

$$O[\{\Psi_i\}] \geq O[\{\Phi_i\}]$$

Wheeler, Kleiner, Wagner. arXiv:2312.00693

Theophilou: variational upper bound for energies

If $\langle \Psi_i | \Psi_j \rangle \propto \delta_{ij}$,

$$w_i > w_j, \quad \forall \quad i < j$$

$$\sum_i w_i E[\Psi_i] \geq \sum_i w_i E[\Phi_i] \, .$$

Weights gets rid of unitary transform problem.

Theophilou, Journal of Physics C: Solid State Physics 12, 5419 (1979)
Gross, Oliveira, Kohn. Phys. Rev. A 37 2809 (1988)

Bulding our algorithm

Theophilou: if $\langle\Psi_i|\Psi_j\rangle \propto \delta_{ij}$, $\sum_i w_i E[\Psi_i] \geq \sum_i w_i E[\Phi_i]$.

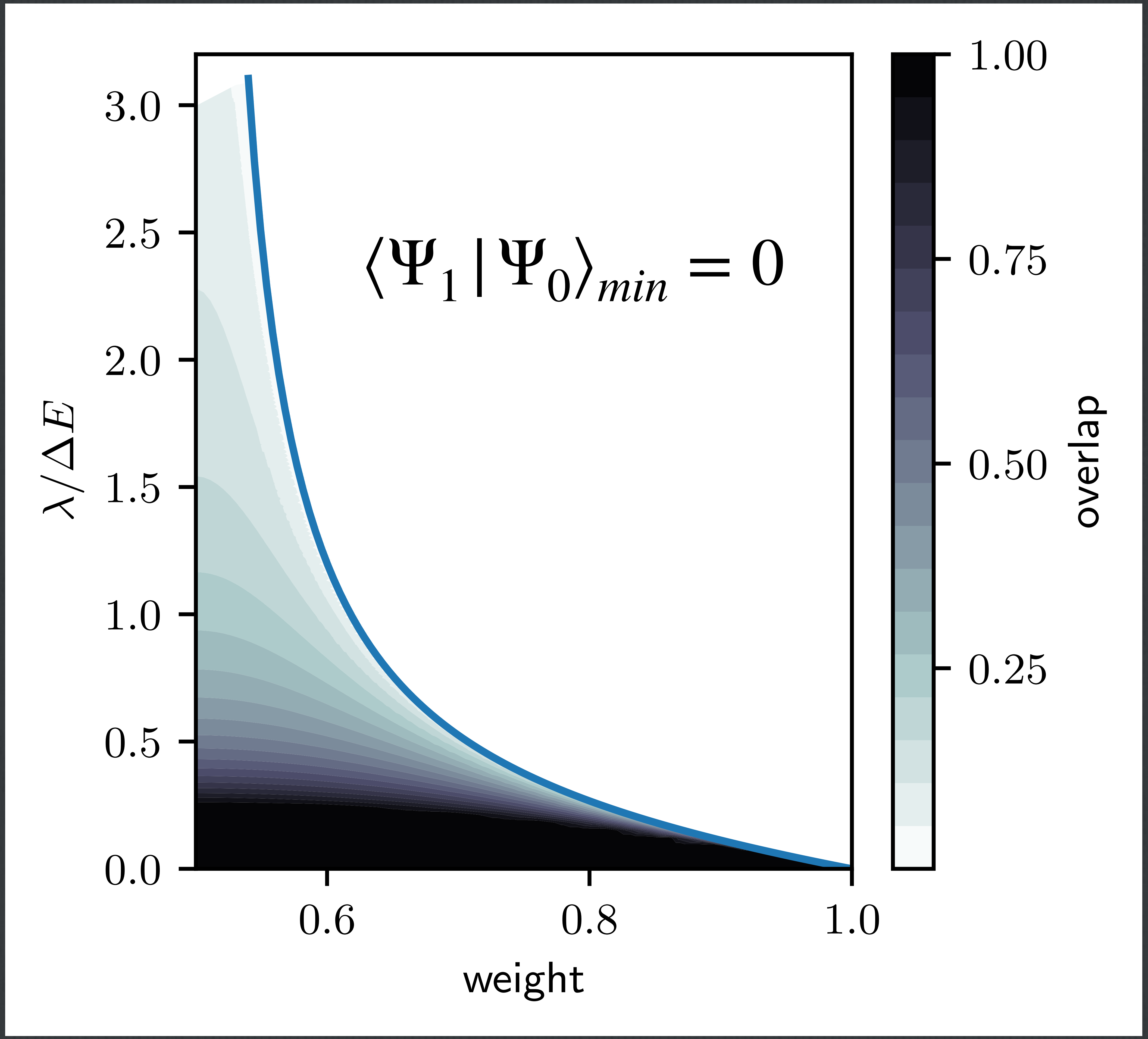
$$O[\{\Psi_i\}] = \sum_i w_i E[\Psi_i] + \lambda \sum_{i<j} |S_{ij}|^2$$

Immediately:

If $\lambda \to \infty$
then $\mathcal{O}[\{\Psi_i\}] \geq \mathcal{O}[\{\Phi_i\}]$

Can we do better? Yes!

Minimum for a two state system



If

$$\lambda > (E_1 - E_0)\frac{w_0 w_1}{w_1 - w_0}$$

then the minimum is at EXACTLY zero overlap and $\mathcal{O}[\{\Psi_i\}] \geq \mathcal{O}[\{\Phi_i\}]$

Generalized limit: outline

$$O[\{\Psi_i\}] = \sum_i w_i E[\Psi_i] + \lambda \sum_{i<j} |S_{ij}|^2$$

$$w_i > w_j, \quad \forall \quad i < j$$

$$\lambda > \max_{i<j} \left[(E_j - E_i) \frac{w_i w_j}{w_i - w_j} \right]$$

The Hessian is positive definite under these conditions when $\{\Psi_i\} = \{\Phi_i\}$

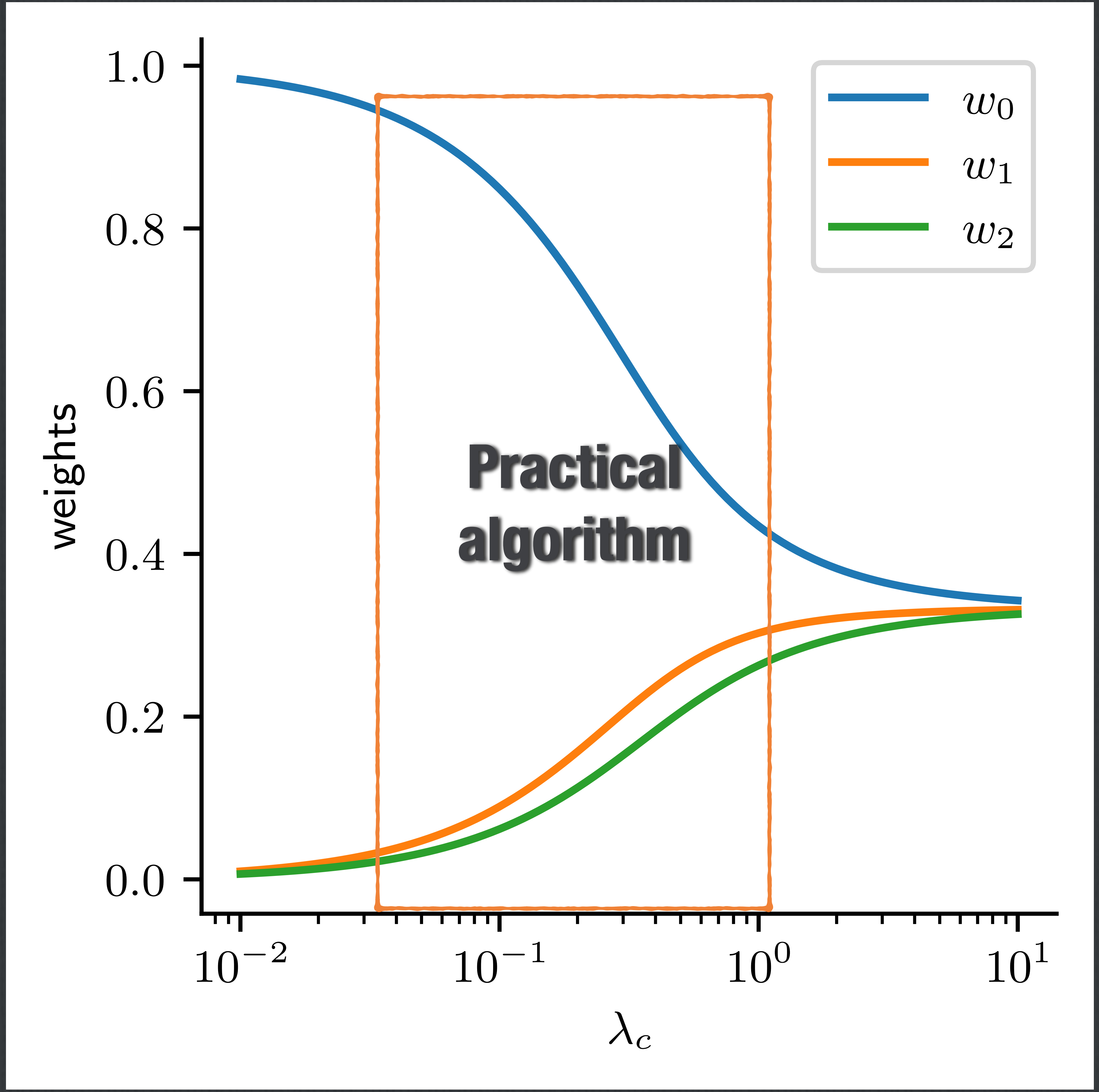
This plus Theophilou → minimum of $\mathcal{O}$ **is located at the lowest N eigenstates of the Hamiltonian.**

Practical algorithm: choosing weights based on λ

$$w_i \propto \frac{1}{1 + E_i w_0/\lambda_c}\,.$$

Only need a rough estimate of E_i.

Set λ 2-3 times larger than λ_c



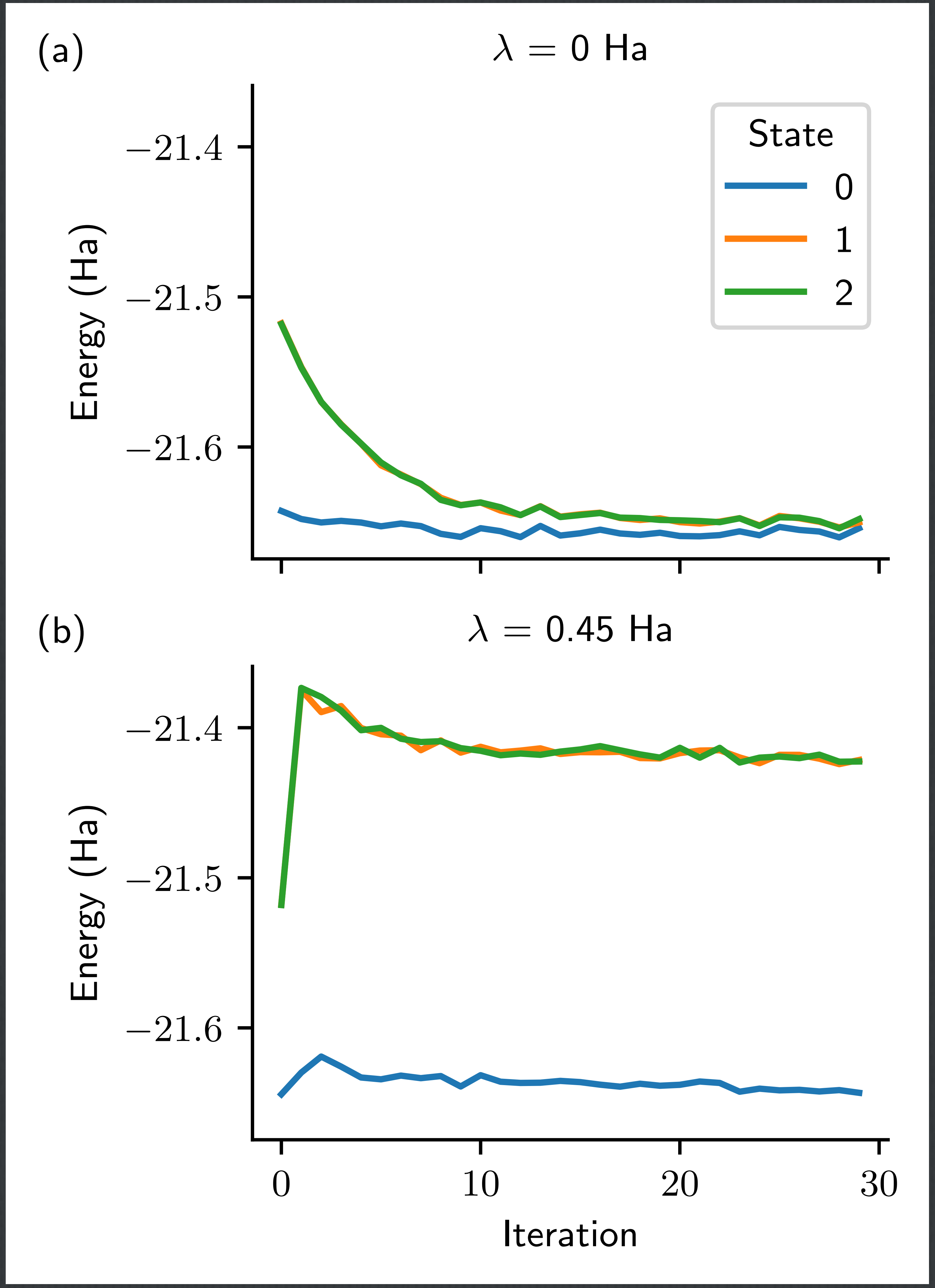
Application to CO

CO molecule

Optimizing determinants, orbitals, and Jastrow parameters individually for each state.

Starting from a linear superposition, if $\lambda > \lambda_c$, converges to the excited states and ground state.

Cost is $\sim N_{states}$ times the one state cost.



More practical algorithm

$$O[\{\Psi_i\}] = \sum_i w_i E[\Psi_i] + \sum_{i<j} \lambda_{ij} |S_{ij}|^2$$

Choose:

$$\lambda_{ij} = \alpha_{ij} w_j,$$

$$\nabla_j O = w_j \left(\nabla_j E[\Psi_j] + \sum_{i<j} 2\alpha_{ij} S_{ij} \nabla_j S_{ij} \right) + \sum_{k>j} 2 w_k \alpha_{jk} S_{jk} \nabla_j S_{jk} \,.$$

where

$$\alpha_{ij} > \frac{E_j - E_i}{1 \cancel{- \frac{w_j}{w_i}}}$$

Take the limit $w_j/w_i \to 0$**, then**

$$\nabla_j O = w_j \left(\nabla_j E[\Psi_j] + \sum_{i<j} 2\alpha_{ij} S_{ij} \nabla_j S_{ij} \right)$$

Same updates as all-at-once Gram Schmidt, with a rescaling!

What we gained

A variational principle for ensembles of states.

A class of methods for optimization (some known methods are included)

Good for variational Monte Carlo:

- Avoids $\langle \Psi_i | H | \Psi_j \rangle$
- Nonlinear parameters are no problem
- No need to superimpose states
- No need to diagonalize matrices.

A bit about pyqmc

Incomplete feature list

Integration with pyscf
All-python (numpy, cupy, future: jax, numba)
About 5000 lines of code

Molecules and solids (3D and 2D Ewald)
Multiple determinants
Several Jastrow factors
Easy to add wave functions

Energy optimization
Excited state optimization
Diffusion Monte Carlo with (mostly) Cyrus's timestepper
Easy to write other algorithms

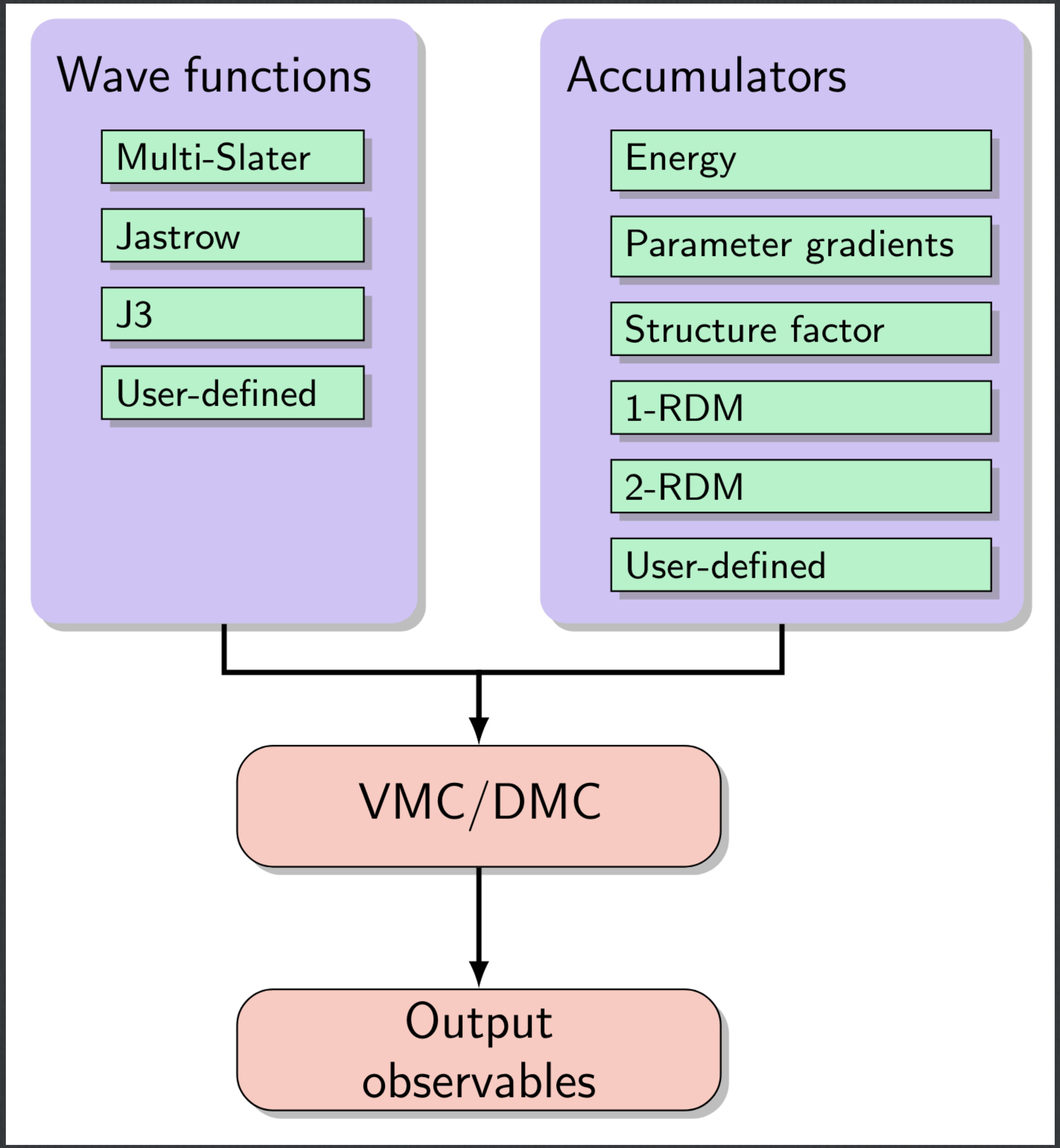
1-RDM
2-RDM
S(k)
Easy to write other observables

Modularity

User-defined objects are first class.

Many built-in (1-RDM, 2-RDM, S(k)…)

```python
import numpy as np
class DipoleAccumulator:
    def __init__(self):
        pass

    def __call__(self, configs, wf):
        return {'electric_dipole':np.sum(configs.configs,axis=1) }

    def shapes(self):
        return {"electric_dipole": (3,)}

    def avg(self, configs, wf):
        d = {}
        for k, it in self(configs, wf).items():
            d[k] = np.mean(it, axis=0)
        return d

    def keys(self):
        return self.shapes().keys()

import pyqmc.recipes
pyqmc.recipes.VMC("h2o.hdf5", "dipole.hdf5",
                  load_parameters="h2o_sj_800.hdf5",
                  accumulators={'extra_accumulators':{'dipole':DipoleAccumulator()}})
```



Integration with pyscf/easy startup

```python
def run_si_scf(chkfile="si_scf.chk", a=5.43):
    cell = gto.Cell(
        atom="Si 0. 0. 0.; Si {0} {0} {0}".format(a / 4),
        unit="angstrom",
        basis="ccecp-ccpvtz",
        ecp="ccecp",
        a=(np.ones((3, 3)) - np.eye(3)) * a / 2,
    )
    cell.exp_to_discard = 0.1
    cell.build()

    kpts = cell.make_kpts([8, 8, 8])
    mf = scf.KRKS(cell, kpts=kpts)
    mf.chkfile = chkfile
    mf.run()
```

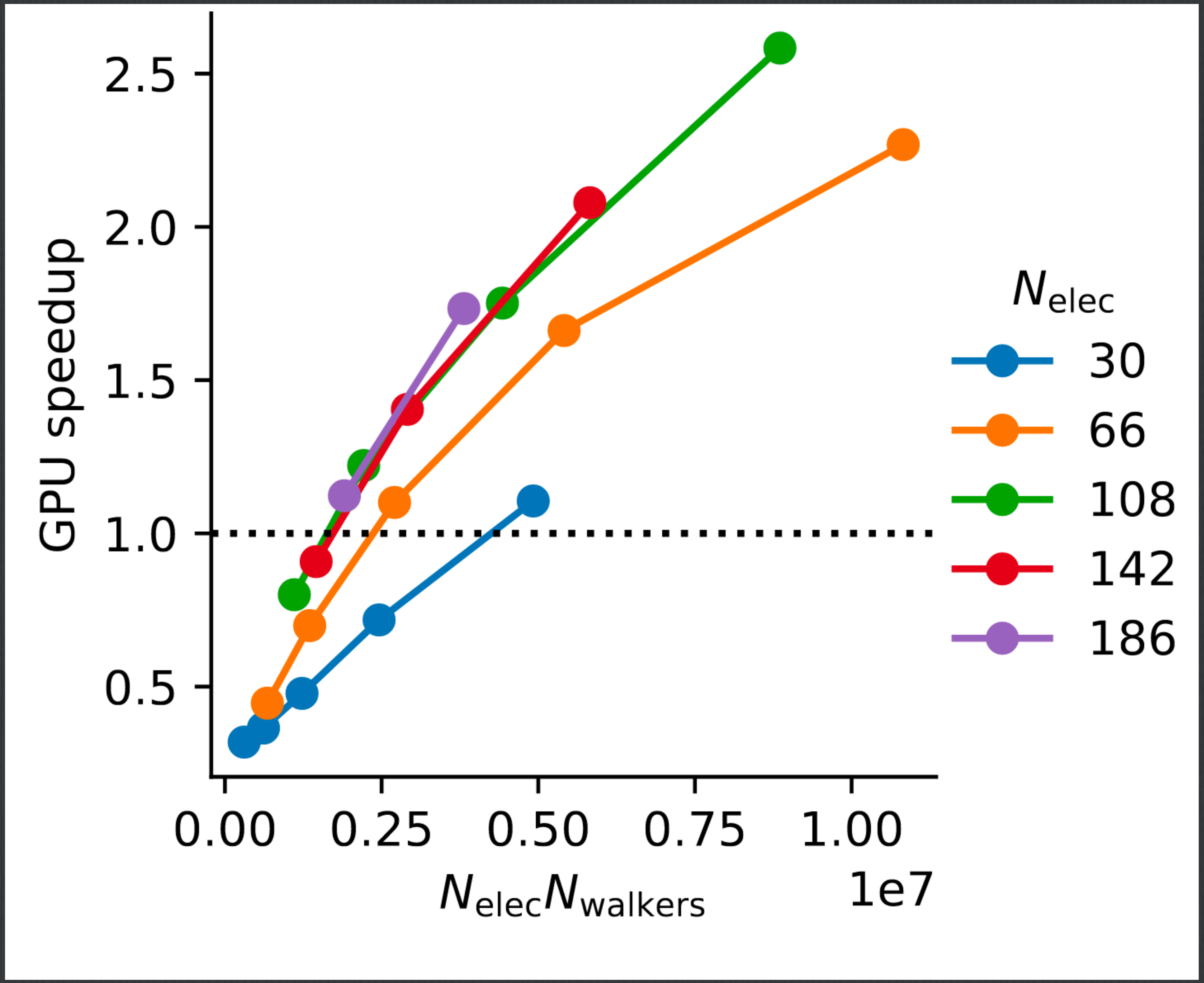
```
❯ pip install pyqmc
```

Solid silicon on the primitive cell

```python
import pyqmc.api as pyq
import numpy as np

def run_si_qmc(chkfile="si_scf.chk", n_conventional=2):
    # Define periodic supercell in PyQMC
    conventional_S = np.ones((3, 3)) - 2 * np.eye(3)
    S = n_conventional * conventional_S

    pyq.OPTIMIZE(chkfile, "si_opt.chk", S=S)
    pyq.DMC(chkfile, "si_dmc.chk", load_parameters="si_opt.chk", S=S, slater_kws={'twist':0} )
```

Any supercell from that starting point (supports orbital optimization)

Reasonable speedup on GPU

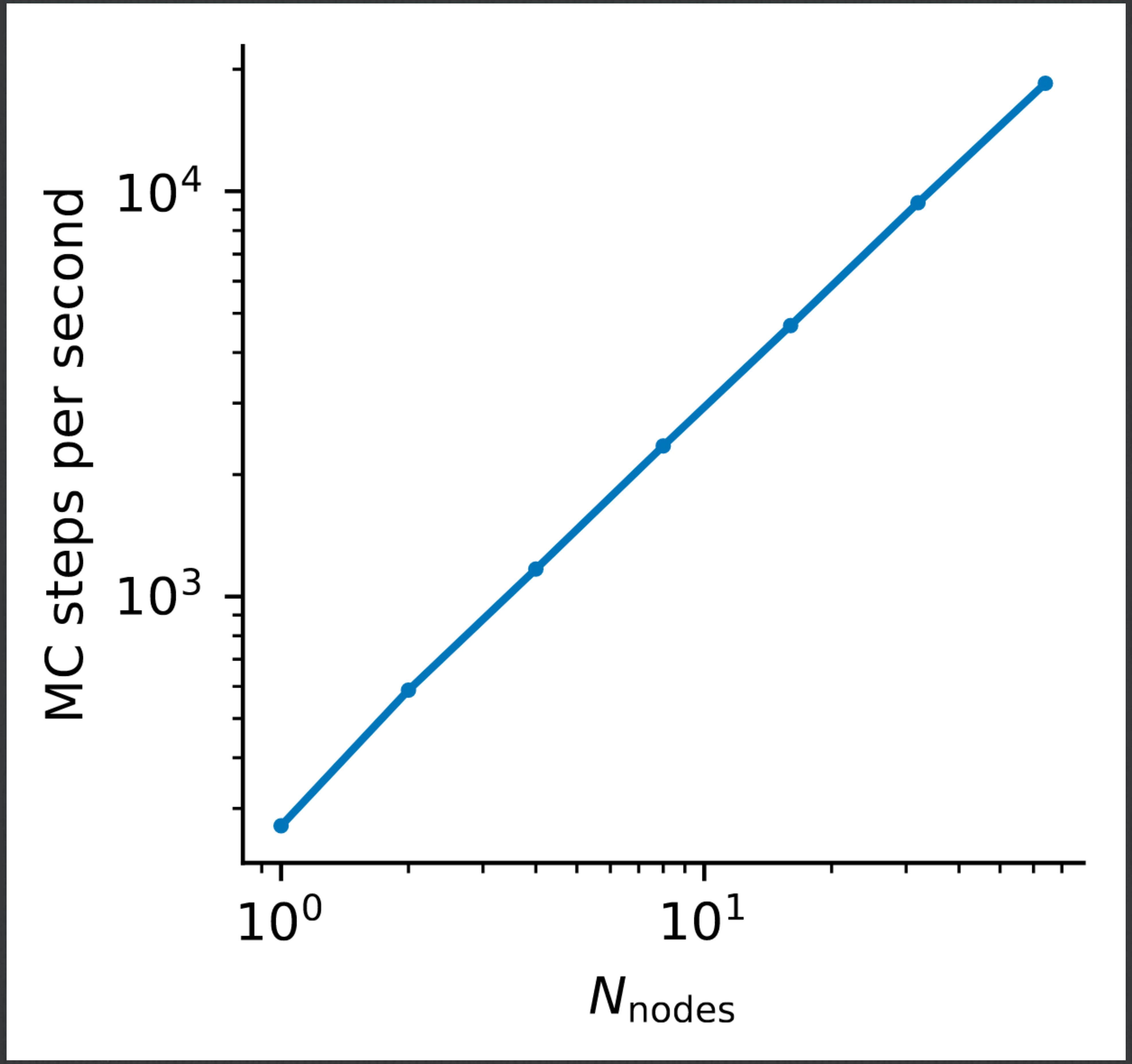


Summit whole-node speedup.

Code is exactly the same (CuPy or JAX for GPU)

Known bottleneck (atomic orbitals) that maybe QMCkl will help with!

Parallelism



futures model.

By using different executors, can use
MPI
TCP/IP
Dask
Kubernetes
Multiple supercomputers simulatenously (parsl)

Summary

$$O[\{\Psi_i\}] = \sum_i w_i E[\Psi_i] + \lambda \sum_{i<j} |S_{ij}|^2$$

$$w_i > w_j, \;\; \forall \;\; i < j$$

$$\lambda > \max_{i<j} \left[(E_j - E_i) \frac{w_i w_j}{w_i - w_j} \right]$$

```
❯ pip install pyqmc
```

Cost and difficulty is about the same as N ground state calculations. Does not need to have separate symmetry.

Code meant for method devlopment.
“Good enough” performance (getting better)
Lots of stuff with little code.