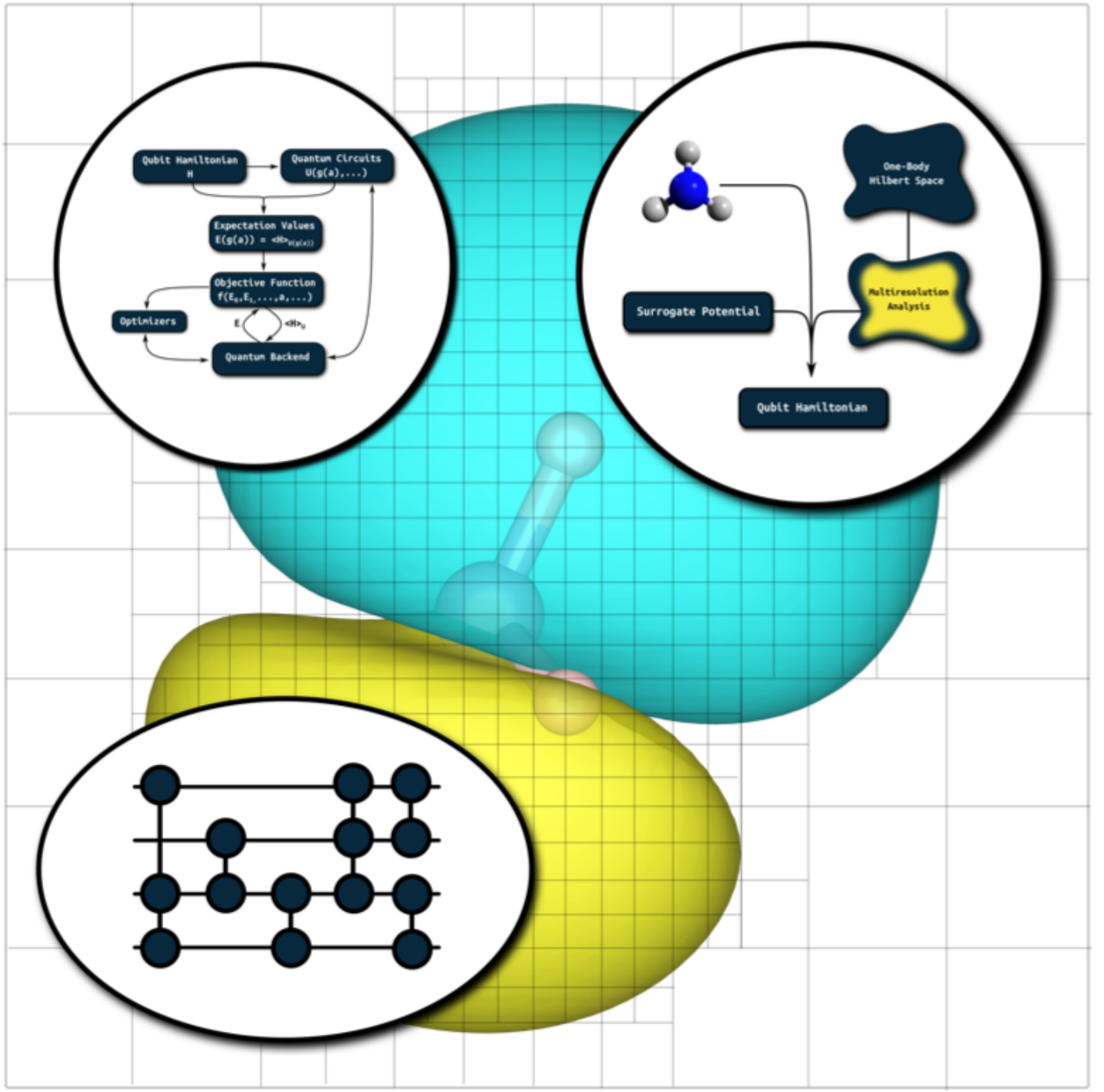
Quantum Algorithms
for Chemistry and Beyond

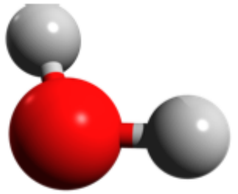
Jakob S. Kottmann

University of Toronto

@JakobKottmann

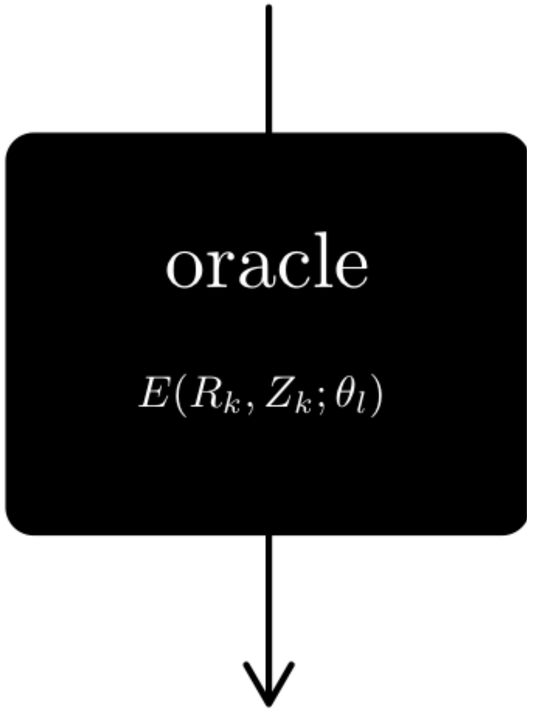
github/tequilahub



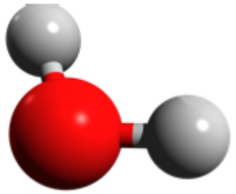


coordinates and charges

$\equiv \{R_k, Z_k\}$



energy



coordinates and charges

$\equiv \{R_k, Z_k\}$

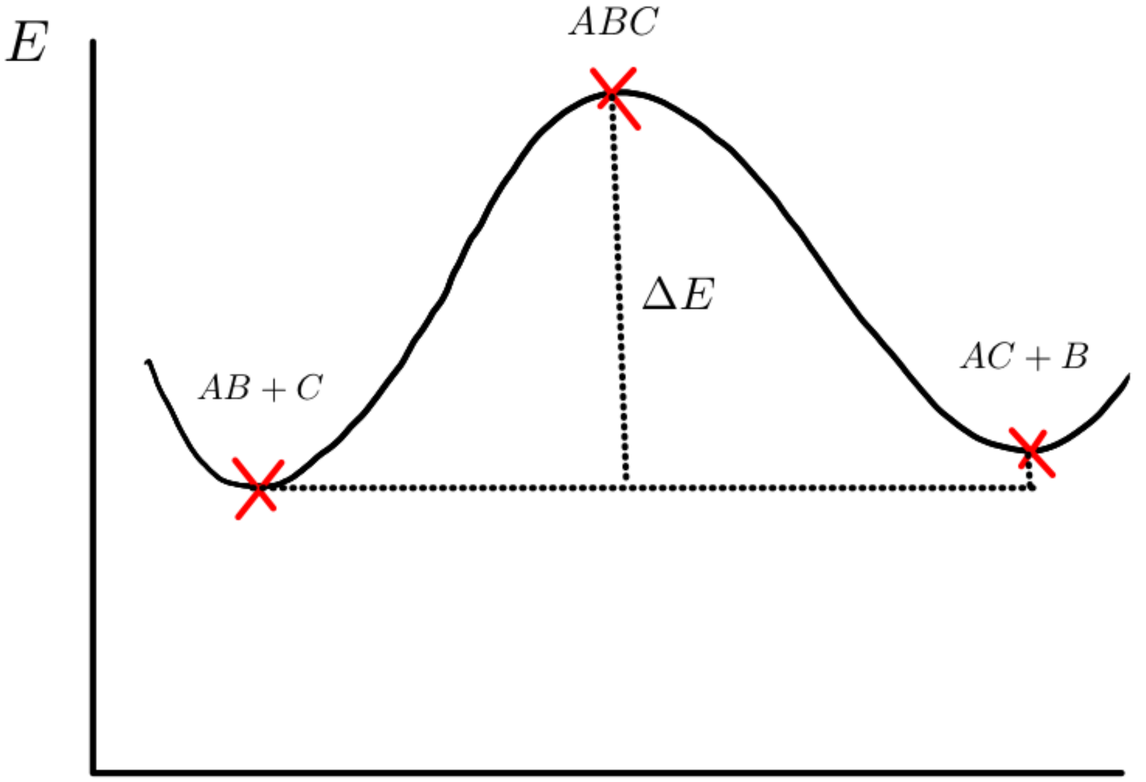
oracle

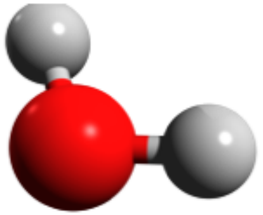
$E(R_k, Z_k; \theta_l)$

$\frac{\partial E}{\partial R_k}$

$\frac{\partial E}{\partial Z_k}$

energy





coordinates and charges

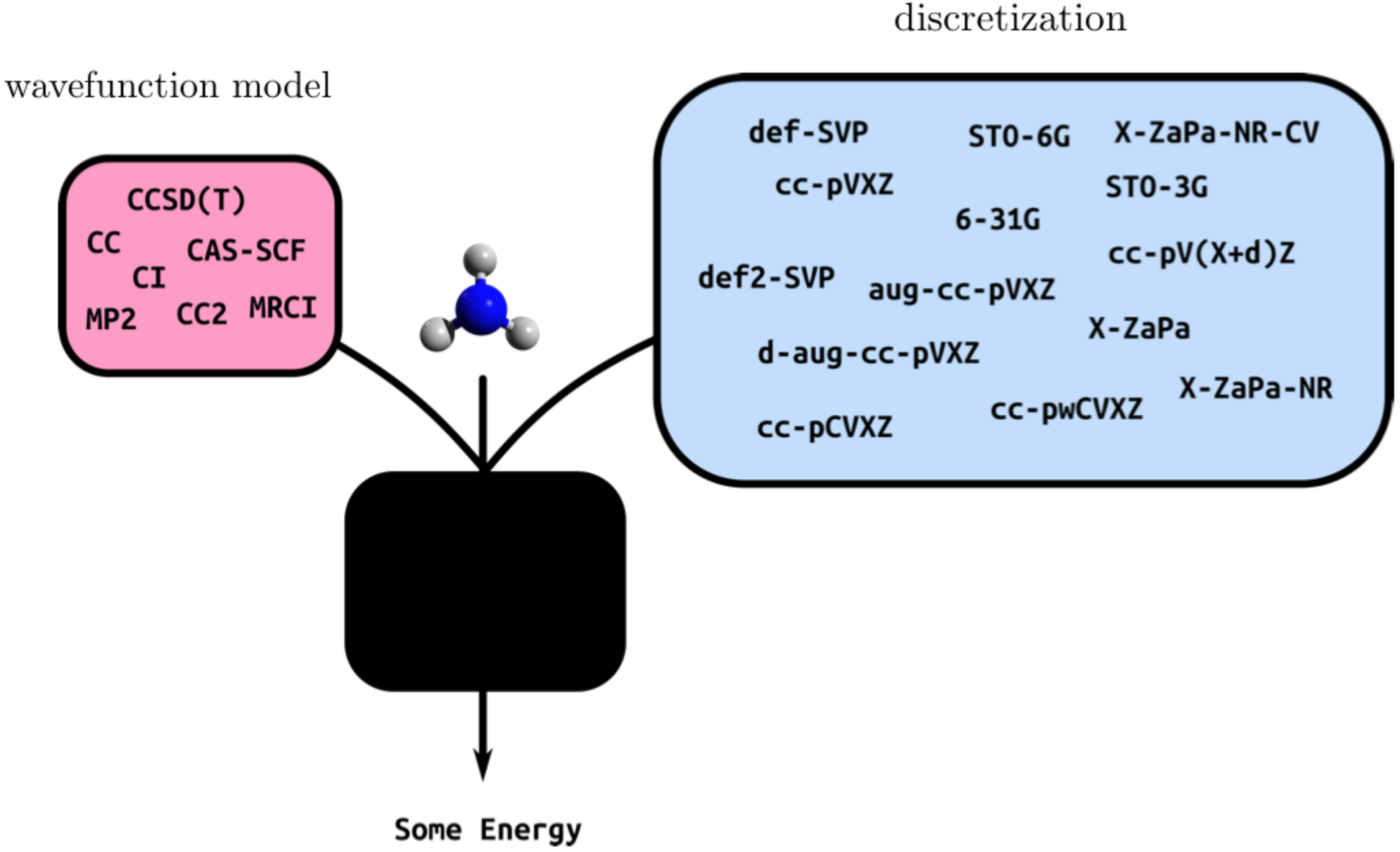
$$\equiv \{R_k, Z_k\} \to V(r) = -\sum_k \frac{Z_k}{\|r - R_k\|}$$

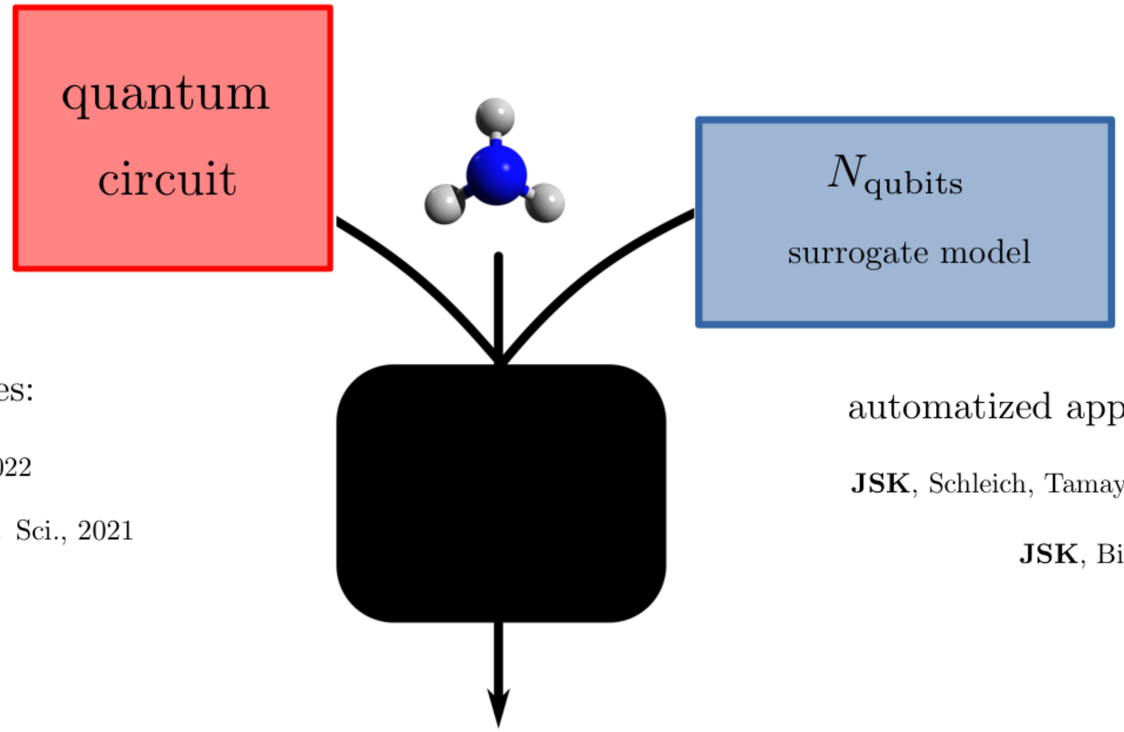
Hamiltonian

$$H(r_0, \ldots, r_N) = -\sum_i \left(\frac{\nabla_{r_i}^2}{2} + V(r_i) \right) + \sum_{i<j} \frac{1}{\|r_i - r_j\|}$$

determine lowest eigenvalue

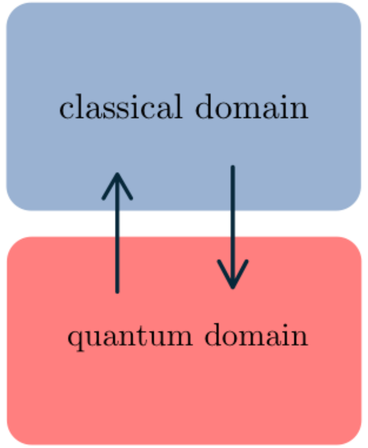
energy


discretization
wavefunction model
CCSD(T)
CC
CAS-SCF
CI
MP2
CC2
MRCI
def-SVP
STO-6G
X-ZaPa-NR-CV
cc-pVXZ
STO-3G
6-31G
cc-pV(X+d)Z
def2-SVP
aug-cc-pVXZ
X-ZaPa
d-aug-cc-pVXZ
X-ZaPa-NR
cc-pwCVXZ
cc-pCVXZ
Some Energy



Energy

+ interpretable quantifiers



automatized approaches:

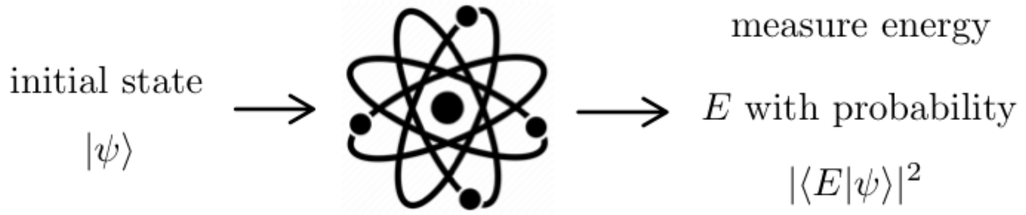
JSK, Aspuru-Guzik, PRA, 2022

JSK, Anand, Aspuru-Guzik, Chem. Sci., 2021

automatized approaches:

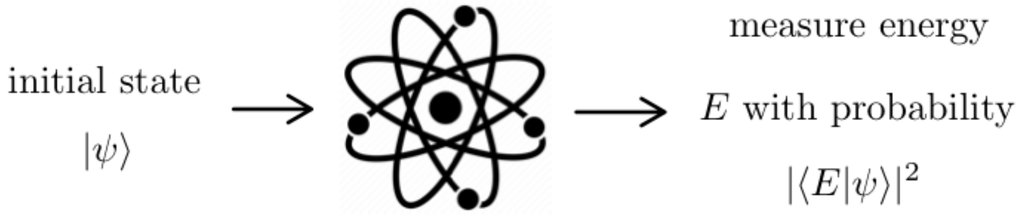
JSK, Schleich, Tamayo-Mendoza, Aspuru-Guzik, JPCL, 2021

JSK, Bischoff, Valeev, JCP, 2020


initial state
$|\psi\rangle$
measure energy
E with probability
$|\langle E|\psi\rangle|^2$

Classical: decent wavefunction $\rightarrow$ decent energy

Quantum: decent wavefunction $\rightarrow$ exact energy



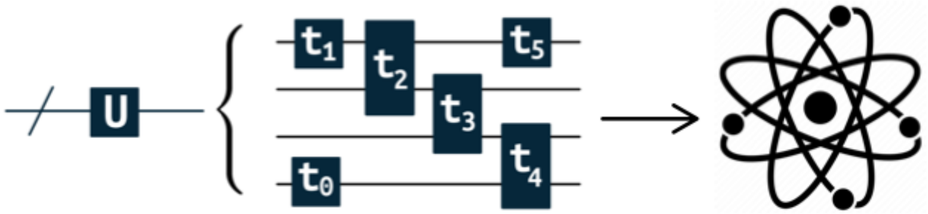
Classical: decent wavefunction $\rightarrow$ decent energy

Quantum: decent wavefunction $\rightarrow$ exact energy

initial state $|\psi\rangle$

measure energy E with probability $|\langle E|\psi\rangle|^2$

optimized initial state $|\psi(\mathbf{t})\rangle$



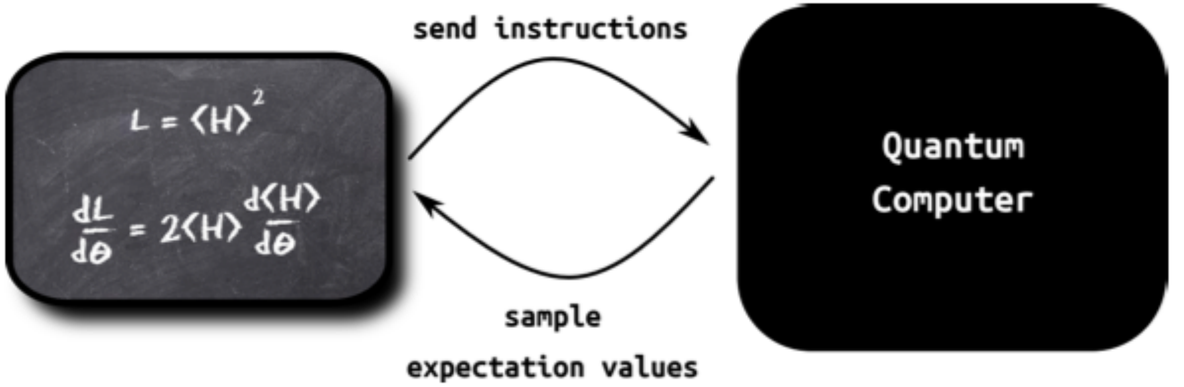
measure expectation value

optimize parameters $\mathbf{t}$

Sumner
Alperin-Lea
UofT/Chem

Alba
Cervera-Lierta
Barcelona Supercomputing Center

Teresa
Tamayo-Mendoza
Harvard

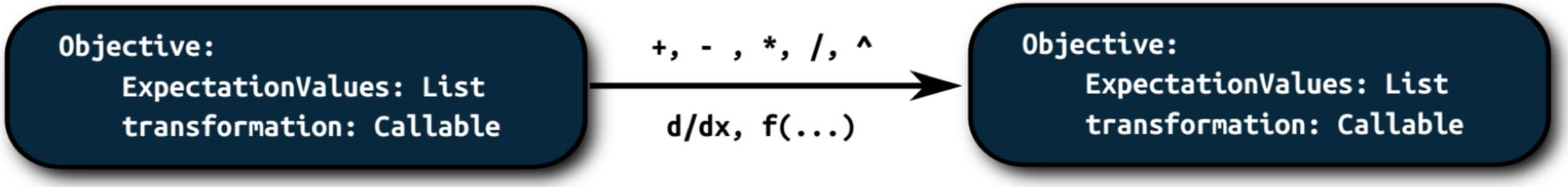


github.com/tequilahub

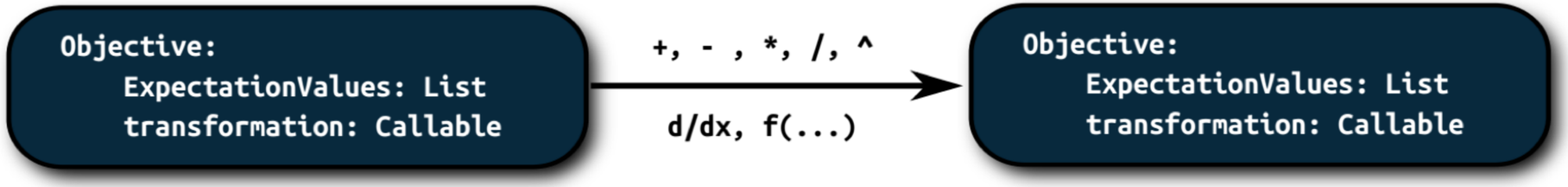
API inspired by madness library

JSK, Alperin-Lea, Cervera-Lierta, Tamayo-Mendoza, …, Quantum Science & Technology, 2021

concept


Objective:
ExpectationValues: List
transformation: Callable
+, - , *, /, ^
d/dx, f(...)
Objective:
ExpectationValues: List
transformation: Callable

concept



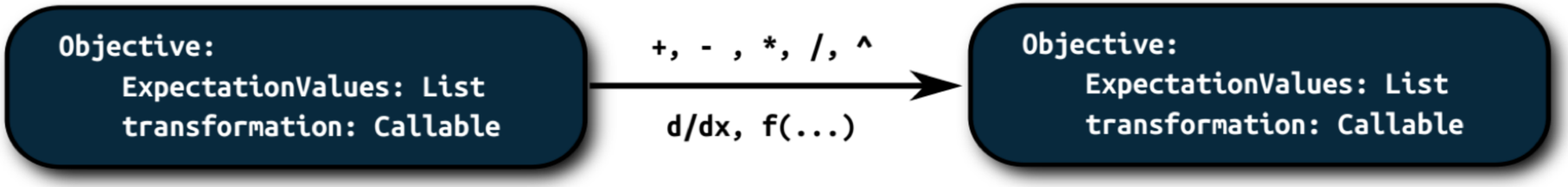
Example:

high level code

```
O1 = E0 + E1
O2 = 0.5*E0**2
O3 = O1**O2
```

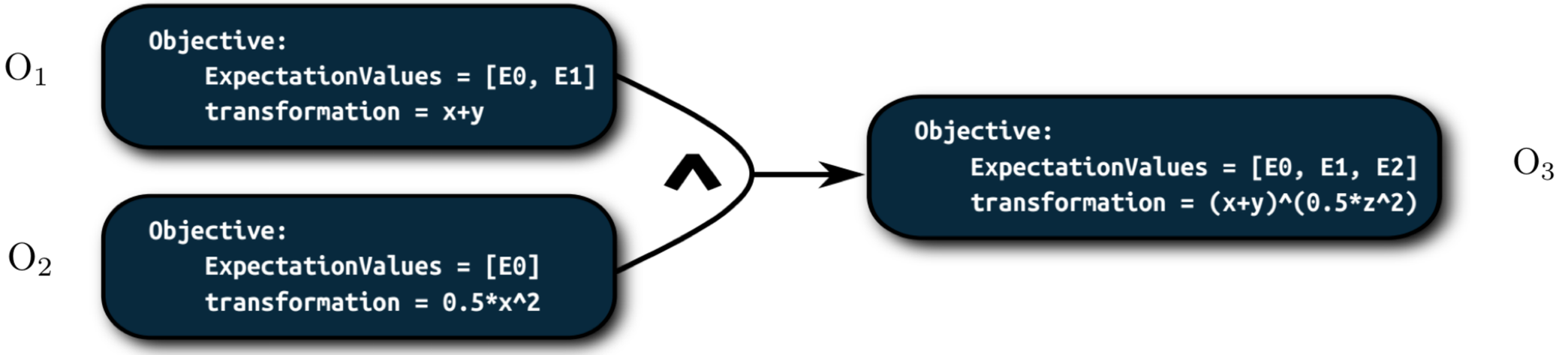
---

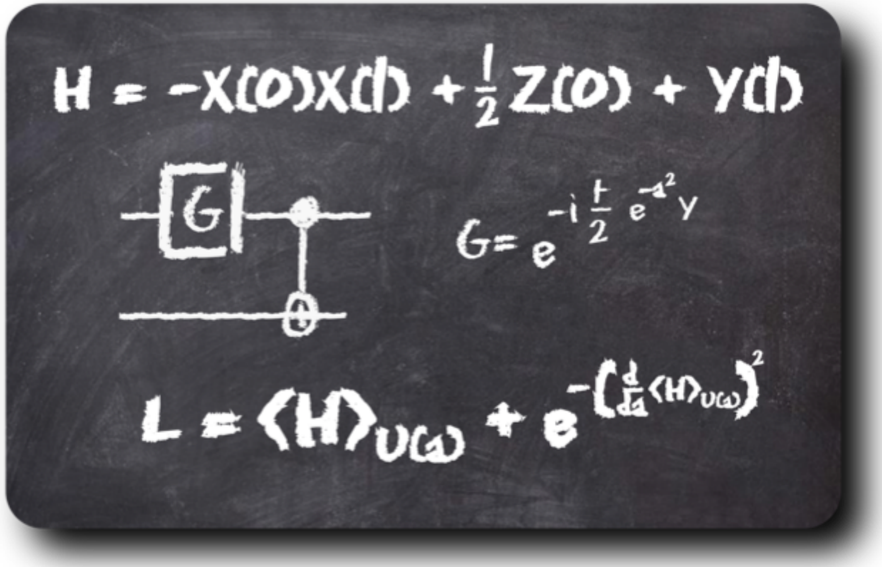
concept

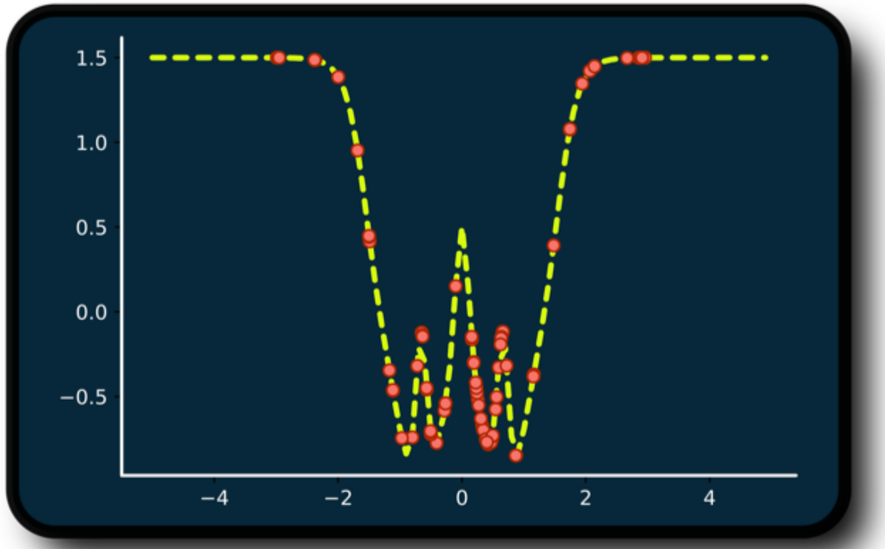


Example:
high level code

```
O1 = E0 + E1
O2 = 0.5*E0**2
O3 = O1**O2
```





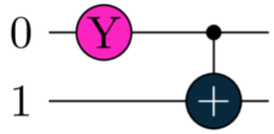



```
a = tq.Variable("a")
U = tq.gates.Ry(angle=(-a**2).apply(tq.numpy.exp)*pi, target=0)
U += tq.gates.X(target=1, control=0)
H = tq.QubitHamiltonian.from_string("-1.0*X(0)X(1)+0.5Z(0)+Y(1)")
E = tq.ExpectationValue(H=H, U=U)
dE = tq.grad(E, "a")
objective = E + (-dE**2).apply(tq.numpy.exp)
result = tq.minimize(method="phoenics", objective=objective)
```

$$L(a) = \langle H \rangle_{U(a)} + e^{-\left(\frac{\partial}{\partial a}\langle H \rangle_{U(a)}\right)^2}$$

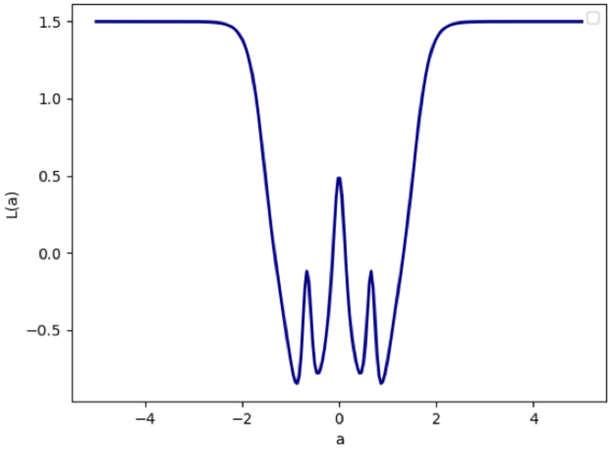
$$H = X(0)X(1) + \frac{1}{2}Z(0) + Y(1)$$

$$U = $$

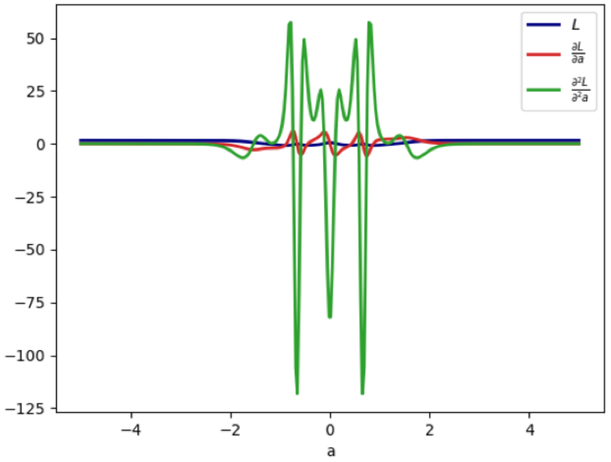



```
a = tq.Variable("a")
f = (-a**2).apply(tq.numpy.exp)

U = tq.gates.Ry(angle=f*numpy.pi, target=0)
U += tq.gates.CNOT(0,1)

H = tq.paulis.from_string("-1.0*X(0)X(1)+0.5*Z(0)+Y(1)")

E = tq.ExpectationValue(H=H, U=U)
dE = tq.grad(E, "a")

L = E + (-dE**2).apply(tq.numpy.exp)
```

```
tq.compile(L)
```




```
dL = tq.grad(L, "a")
dL2 = tq.grad(dL, "a")
```



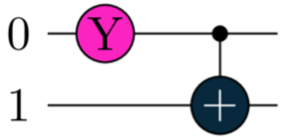

```
print(L)
```

```
>>> Objective with 3 unique expectation values
    total measurements = 9
    variables          = [a]
    types              = not compiled
```

example_objective.py

$$L(a) = \langle H \rangle_{U(a)} + e^{-\left(\frac{\partial}{\partial a}\langle H \rangle_{U(a)}\right)^2}$$

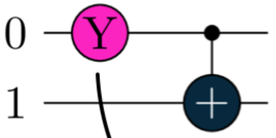
$$H = X(0)X(1) + \frac{1}{2}Z(0) + Y(1)$$

$U =$ 

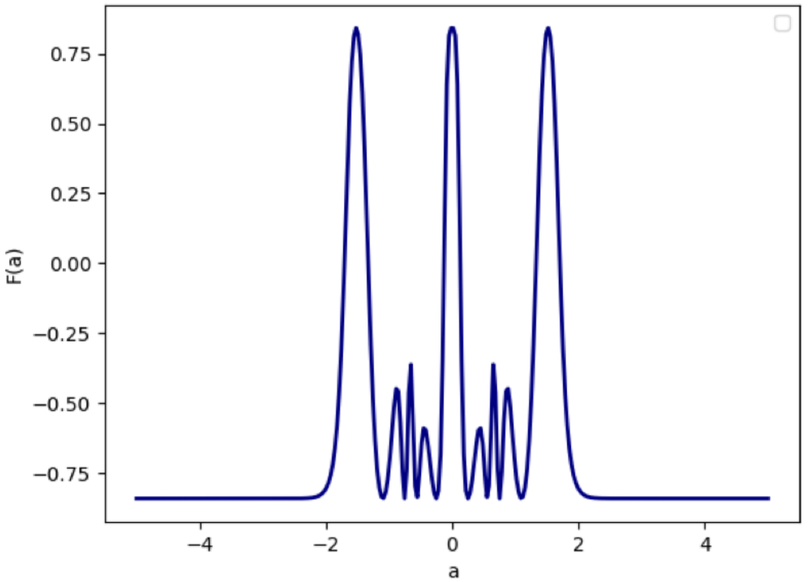

```
L = tq.compile(L)
U2 = tq.gates.Ry(angle=L, target=0)
U2+= tq.gates.CNOT(0,1)
```

$$F(a) = \sin\left(\langle H_2 \rangle_{U_2}\right)$$

$$H_2 = X(0) + X(1) + X(0)X(1)$$

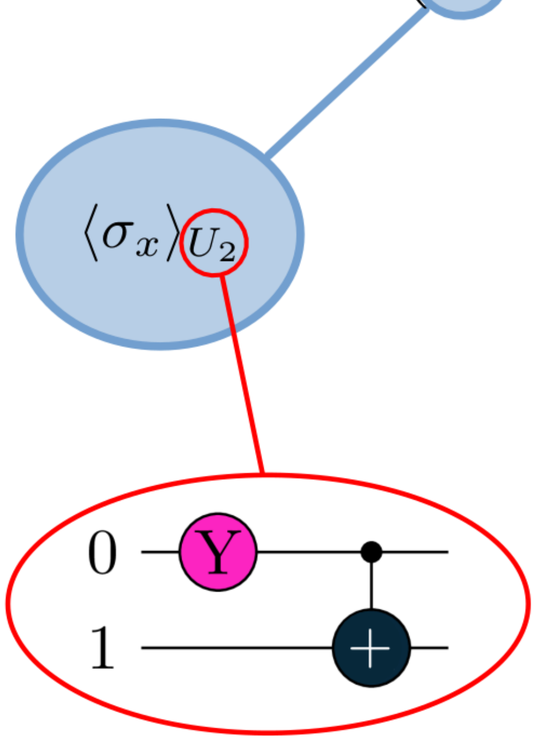
$U_2 =$ 

$$e^{-i\frac{L(a)}{2}Y(0)}$$



example_nested_objective.py

$$\begin{pmatrix} 0 & E_1 \\ E_2 & 0 \end{pmatrix} \begin{pmatrix} E_1 \\ 1 \end{pmatrix} \xrightarrow[\text{tq.simulate}]{a=1.0, b=0.5} -\begin{pmatrix} 0.841 \\ 0.738 \end{pmatrix}$$



Gaurav Saxena, U. Calgary

```
import tequila as tq

U1 = tq.gates.Rx(angle="a", target=0)
U2 = tq.gates.Ry(angle="b", target=0)
U2 = tq.gates.CNOT(0,1)

H1 = tq.paulis.Y(0)
H2 = tq.paulis.X(0)+tq.paulis.Z(1)

E1 = tq.ExpectationValue(U1,H1)
E2 = tq.ExpectationValue(U2,H2)

M = tq.QTensor([0.0,E1,E2,0.0], shape=(2,2))
c = tq.QTensor([E1,1.0], shape=(2,))
b = M.dot(c)

variables={"a":1.0, "b":0.5}
result = tq.simulate(b, variables)
```

```
Im, Re = tq.braket(bra=Ui, ket=Uj, H=H)
```

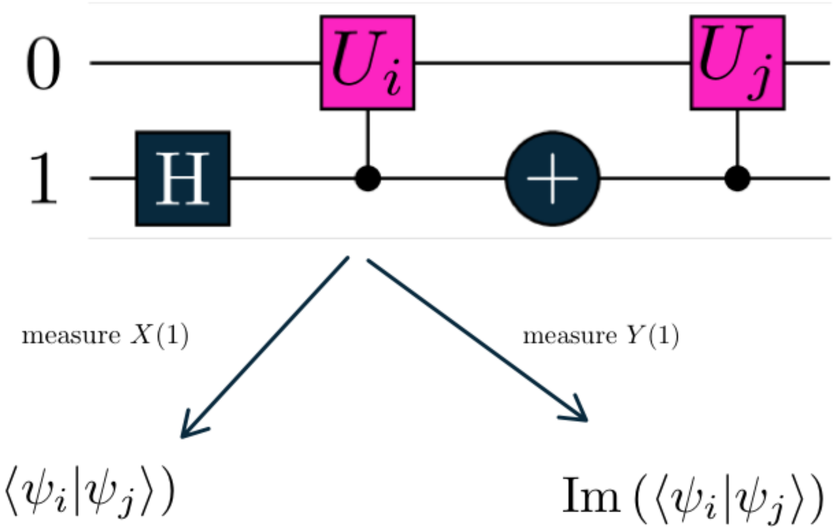
Francesco Scala, U. Pavia

$$\mathrm{Re}\left(H_{ij}\right), \mathrm{Im}\left(H_{ij}\right) = \langle\psi_i|H|\psi_j\rangle$$

$$\downarrow$$

$$\sum_k c_k \langle U_i|P_k U_j\rangle \longrightarrow \langle\psi_i|\psi_j\rangle$$

$$\downarrow$$



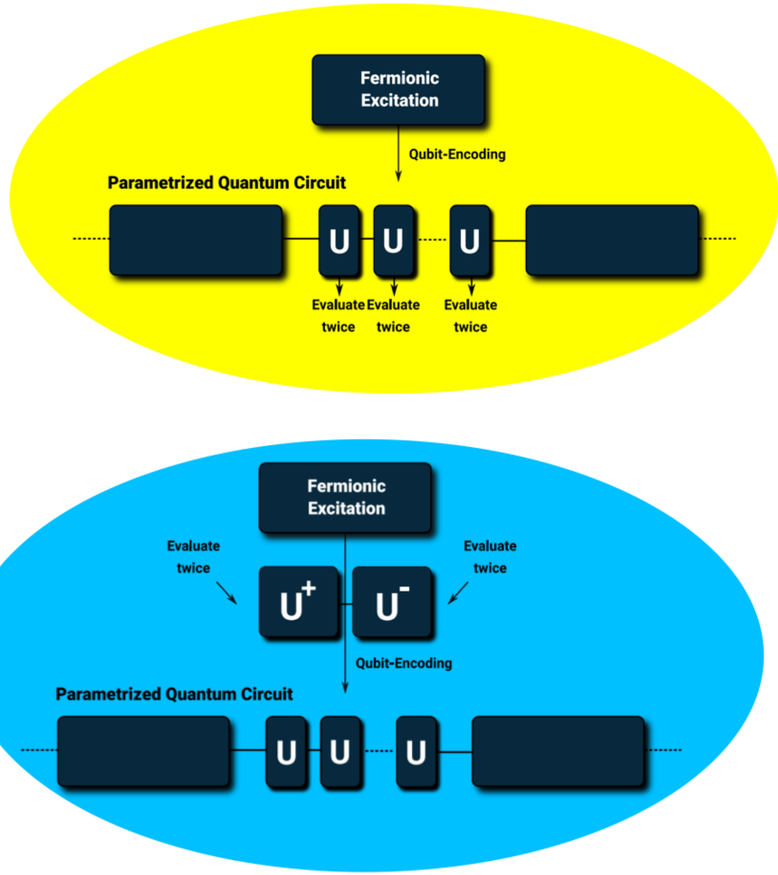
qosf

Examples in Quantum Chemistry

Abhinav
Anand
UofT/Chem

gradient cost for n electron excitation

Generator Form	Gradient Cost	Strategy
$G_{\mathbf{pq}} = \sum_{\mathbf{i}} c_{\mathbf{i}} \sigma_{\mathbf{i}}$	$\mathcal{O}\left(2^{2n}\right)$	shift-rule Eq. (6)
$G_{\mathbf{pq}} = \frac{1}{2}\left(G_+ + G_-\right)$	4	fermionic-shift Eq. (16)
Real Wavefunctions		
$G_{\mathbf{pq}} = \frac{1}{2}\left(G_+ + G_-\right)$	2	fermionic-shift Eq. (19)
Generator Approximation		
$G_{\mathbf{pq}} \approx G_{\pm}$	2	shift-rule Eq. (6)



→ generalizable

follow-ups:

Izmaylov *et.al.* 2021

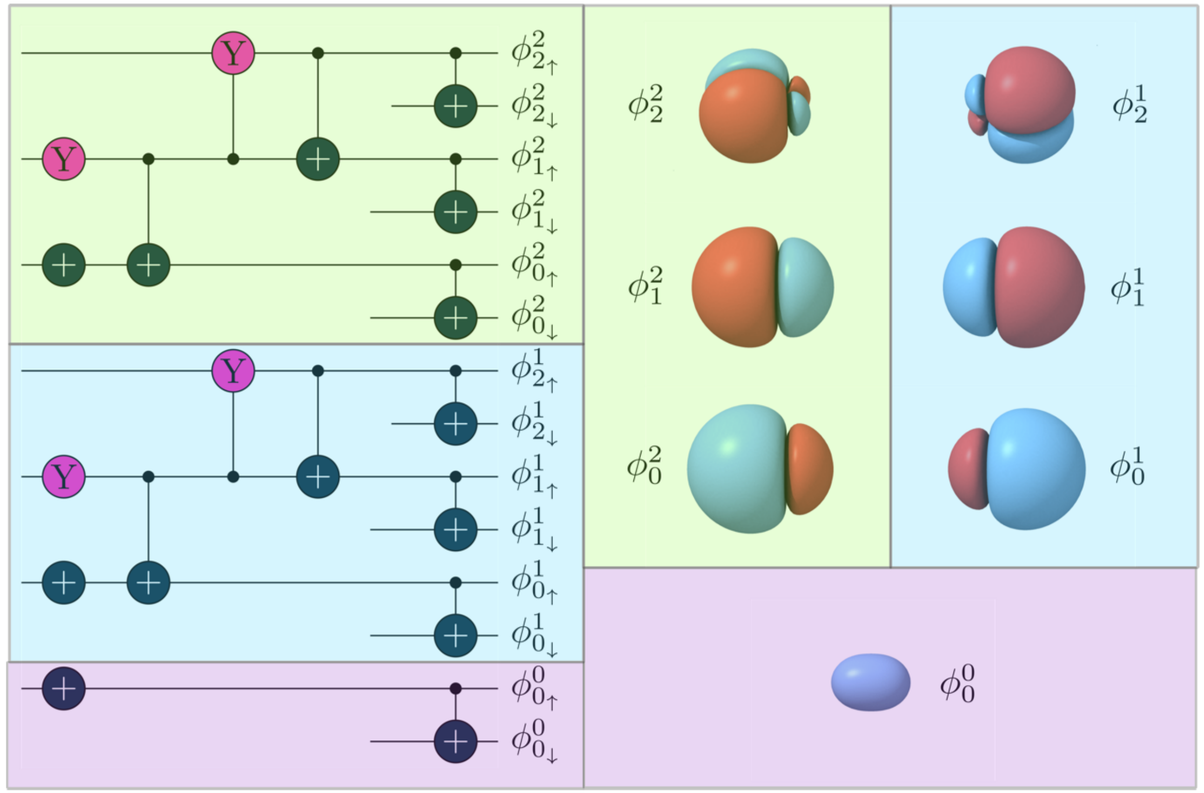
Anselmetti *et.al.* 2021

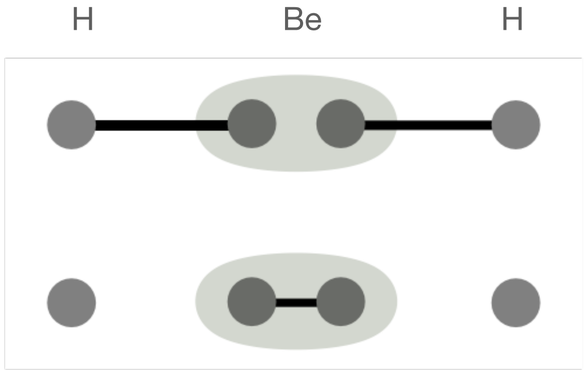
Wierichs *et.al.* 2021

Basic building blocks for Unitary Coupled-Cluster

recent review: A. Anand *et.al.* 2021

JSK, A. Anand, A. Aspuru-Guzik, Chemical Science 2021



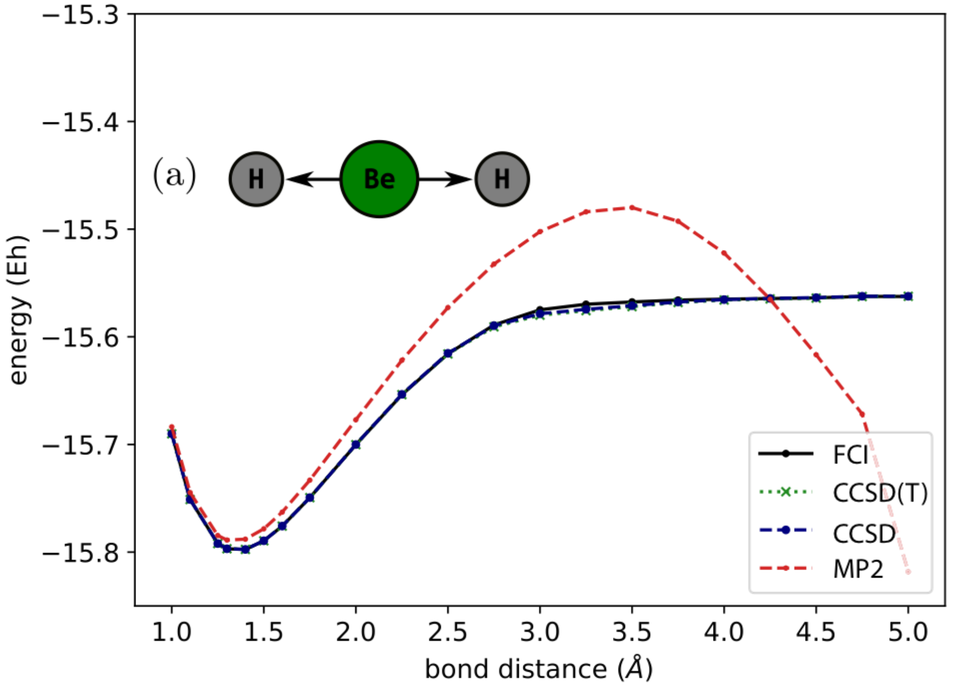


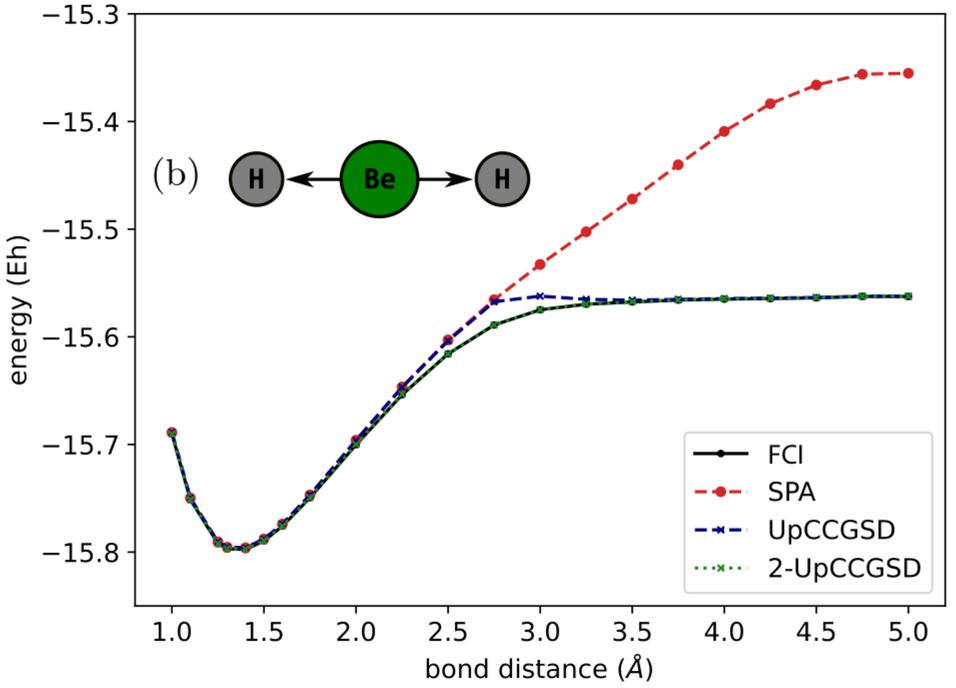
Classically Simulable: Becomes a classical pre-compilation step

Gives good initial states

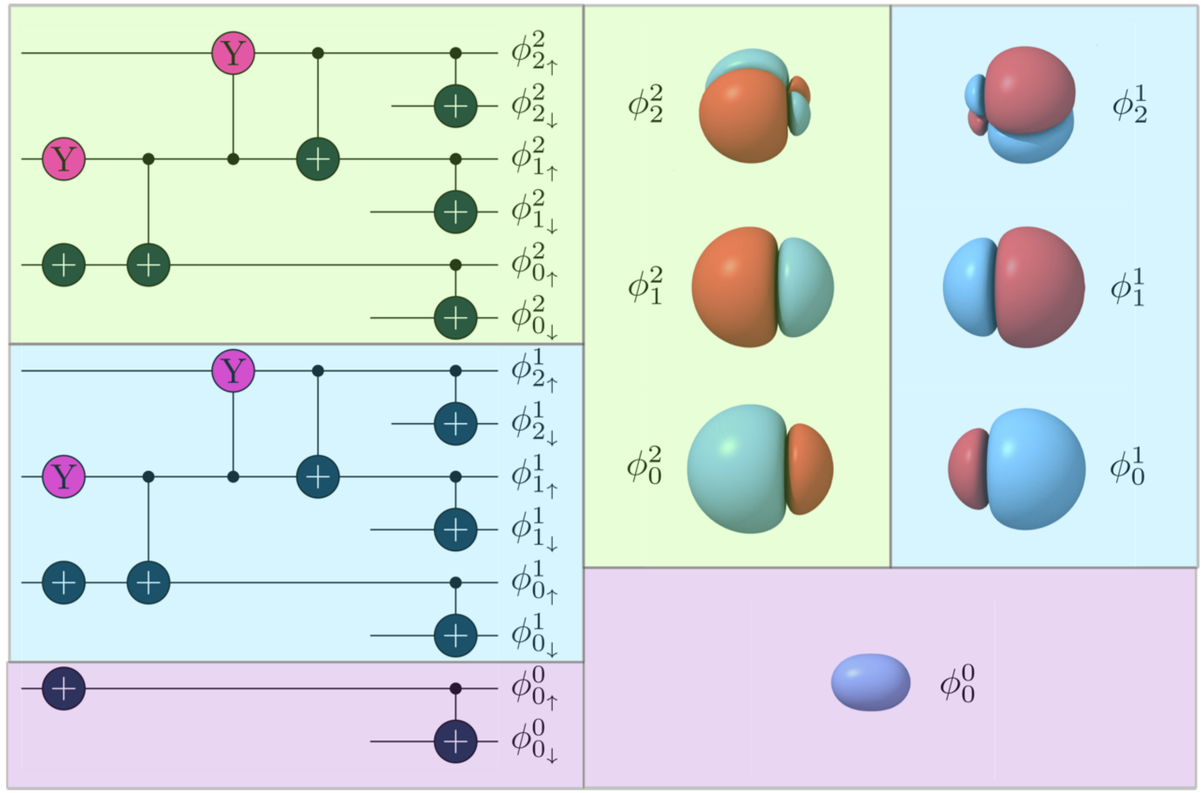
Solves prominent “benchmark” systems (H2 and LiH) exactly

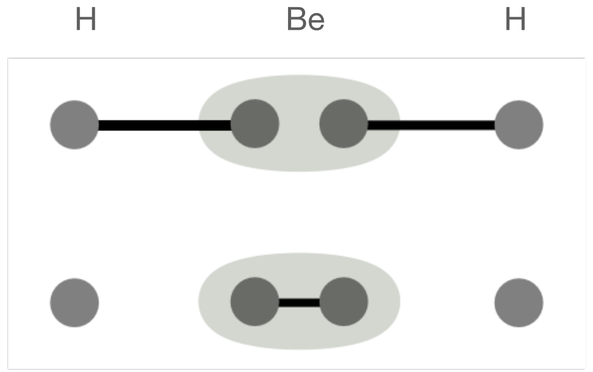
Separable Pair Approximations: JSK, Aspuru-Guzik, PRA, 2022

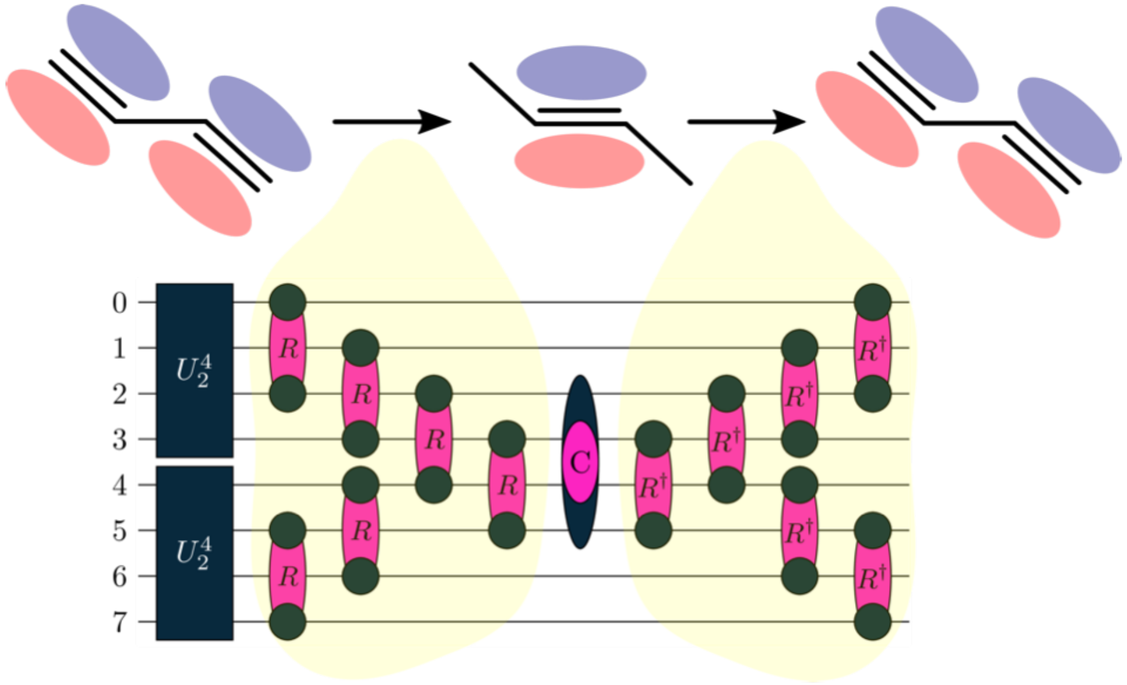




Orbital-Optimized-SPA equal results as “UpCCGSD”







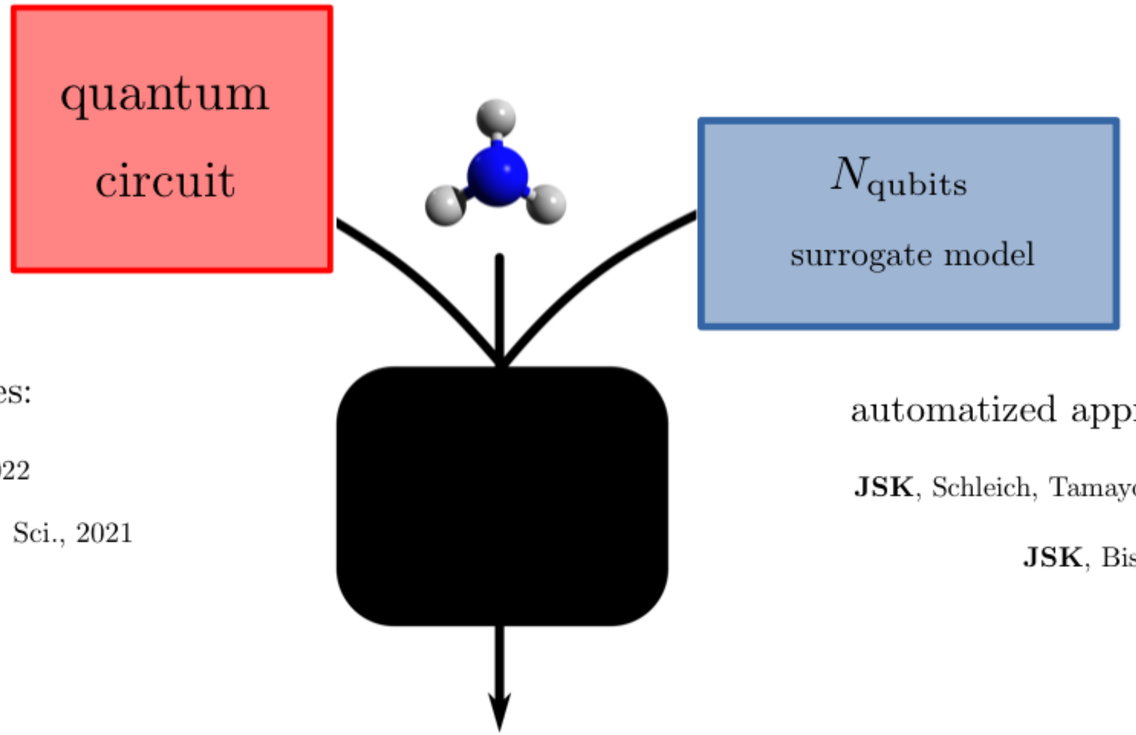
Classically Simulable: Becomes a classical pre-compilation step

Gives good initial states

Solves prominent “benchmark” systems (H2 and LiH) exactly

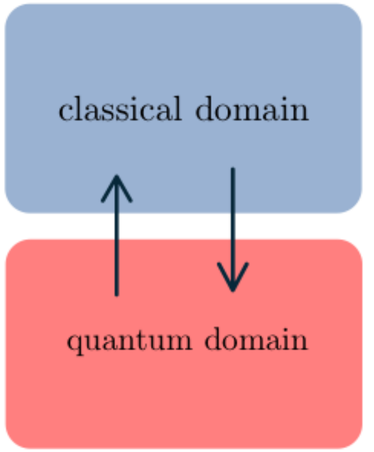
Separable Pair Approximations: JSK, Aspuru-Guzik, PRA, 2022

Extended Framework: JSK 2022 (tbp)



Energy

+ interpretable quantifiers



automatized approaches:

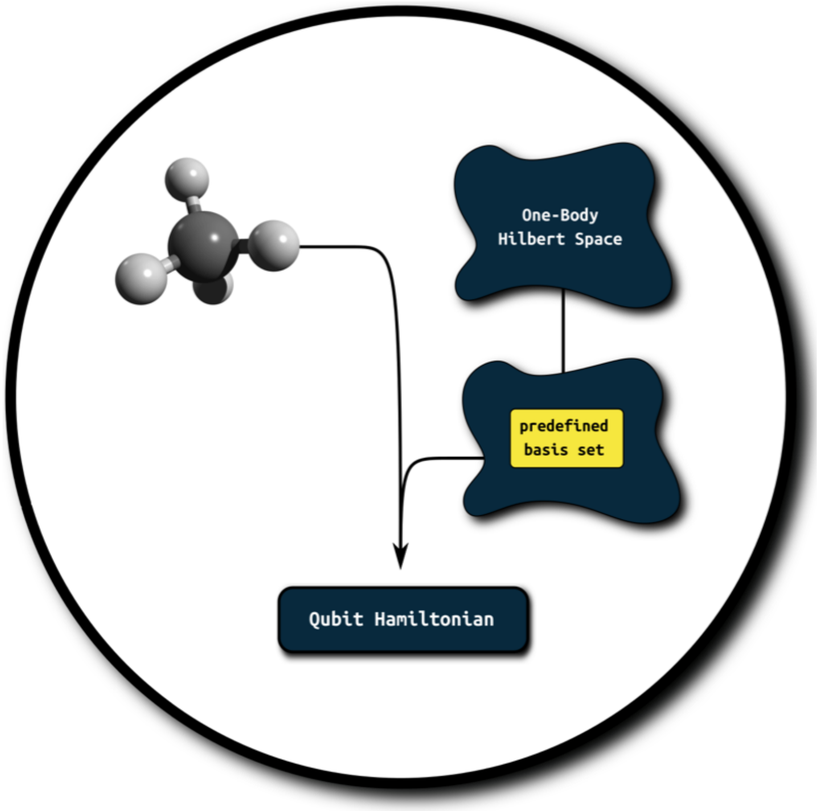
JSK, Aspuru-Guzik, PRA, 2022

JSK, Anand, Aspuru-Guzik, Chem. Sci., 2021

automatized approaches:

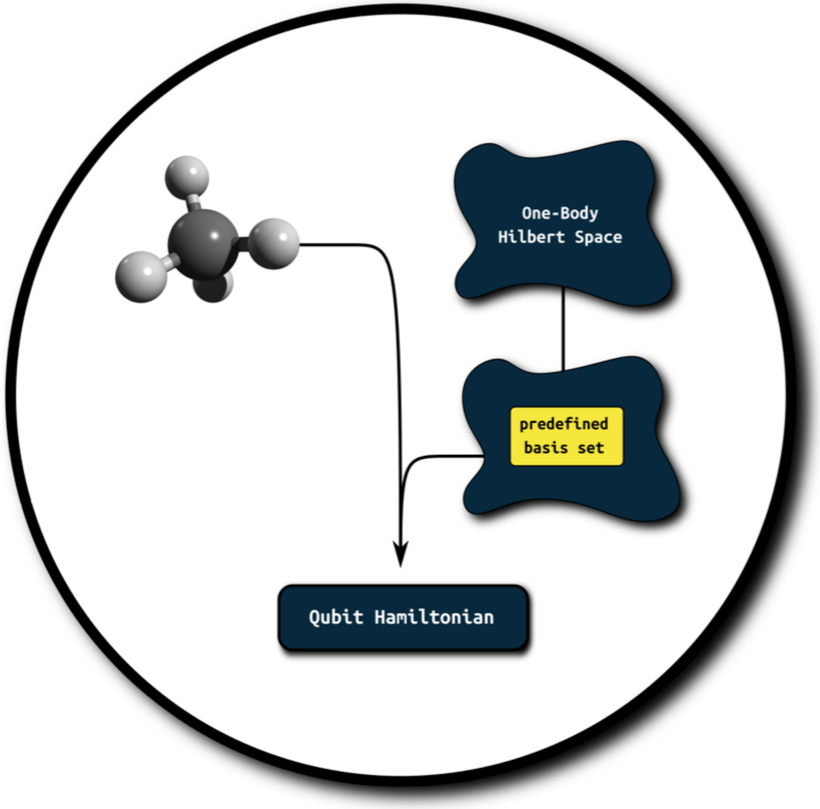
JSK, Schleich, Tamayo-Mendoza, Aspuru-Guzik, JPCL, 2021

JSK, Bischoff, Valeev, JCP, 2020




```
mol = tq.Molecule(geometry="ch4.xyz", basis_set="sto-3g")

H = mol.make_hamiltonian()
U = mol.make_ansatz(name="UpCCD")
E = tq.ExpectationValue(H=H, U=U)

result = tq.minimize(E)
exact =  tq.compute_energy("fci")
```

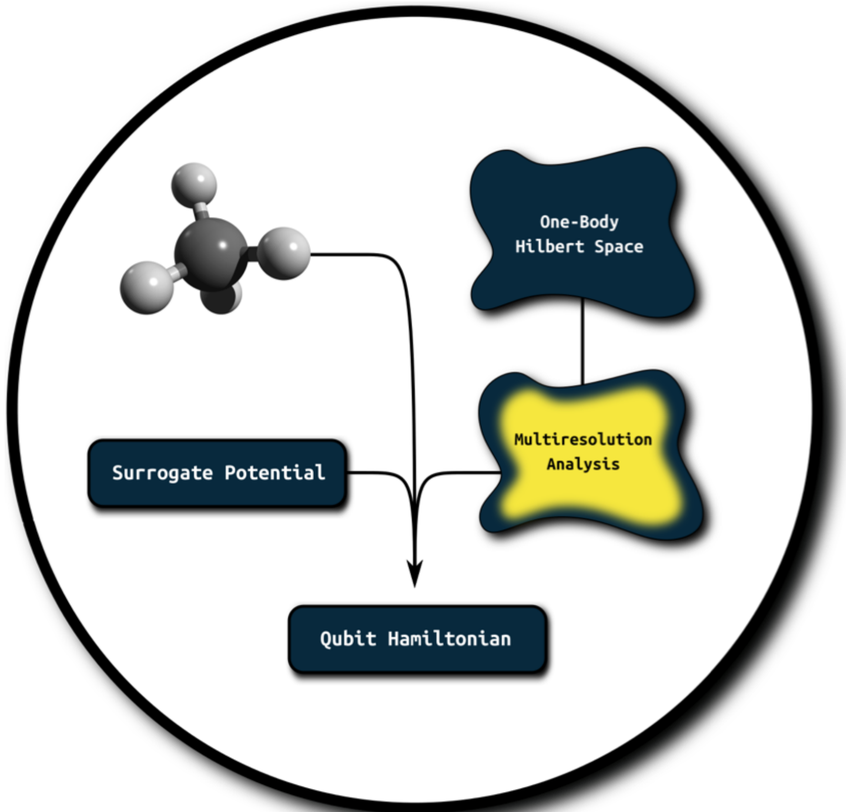
backends: PySCF or Psi4




```
mol = tq.Molecule(geometry="ch4.xyz", basis_set="sto-3g")

H = mol.make_hamiltonian()
U = mol.make_ansatz(name="UpCCD")
E = tq.ExpectationValue(H=H, U=U)

result = tq.minimize(E)
exact =  tq.compute_energy("fci")
```

backends: PySCF or Psi4

```
System Adapted:

8 orbitals
VQE/MRA-PNO : -40.2761
FCI/MRA-PNO : -40.2926

Basis Sets:

9 orbitals
VQE/STO-3G  : -39.7580
FCI/STO-3G  : -39.8060

17 orbitals
FCI/6-31G   : -40.3013
```




```
mol = tq.Molecule(geometry="ch4.xyz")

H = mol.make_hamiltonian()
U = mol.make_ansatz(name="UpCCD")
E = tq.ExpectationValue(H=H, U=U)

result = tq.minimize(E)
exact =  tq.compute_energy("fci")
```

backend: MADNESS

ch4.py

Philipp
Schleich
UofT/CS

Teresa
Tamayo-Mendoza
Harvard

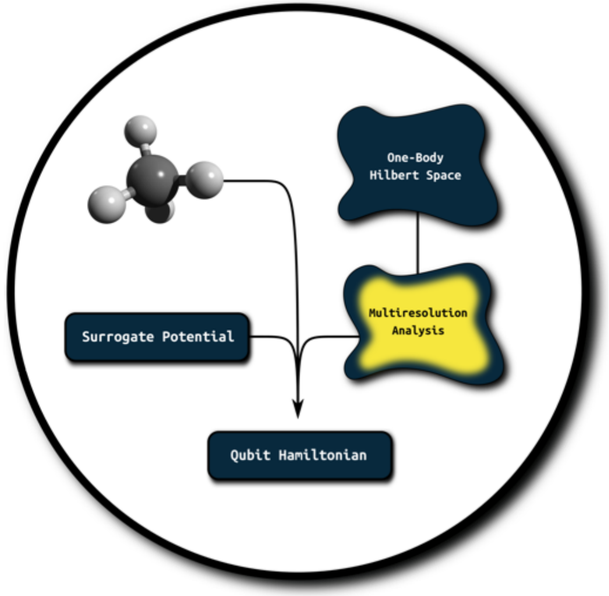
JSK*, Schleich, Tamayo-Mendoza, Aspuru-Guzik*, JPCL, 2021

tequila

qubit savings

e.g. from 50-100 to 10-20

black box method



github/madness
Harrison et. al.

effective 2-electron methods
ground and excited state
JSK*& Bischoff*JCTC 2018
JSK*& Bischoff*JCTC 2018

effective 1-electron methods
excited state
JSK, Höfener, Bischoff, PCCP 2015

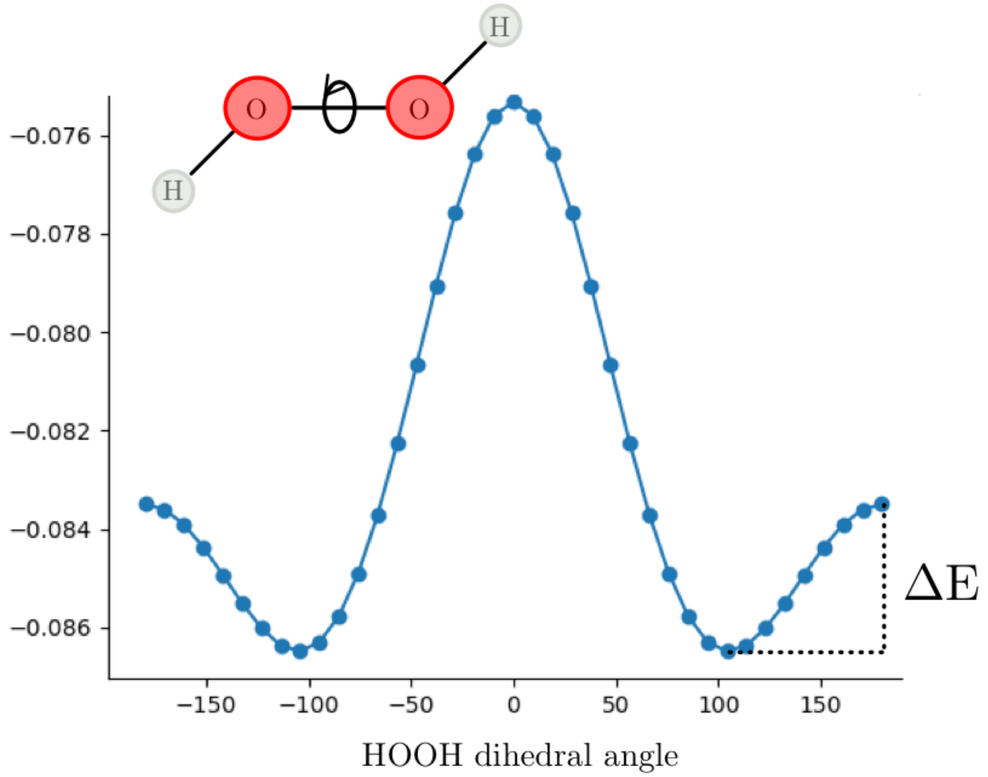
separated 2-electron methods
JSK, Bischoff, Valeev 2020

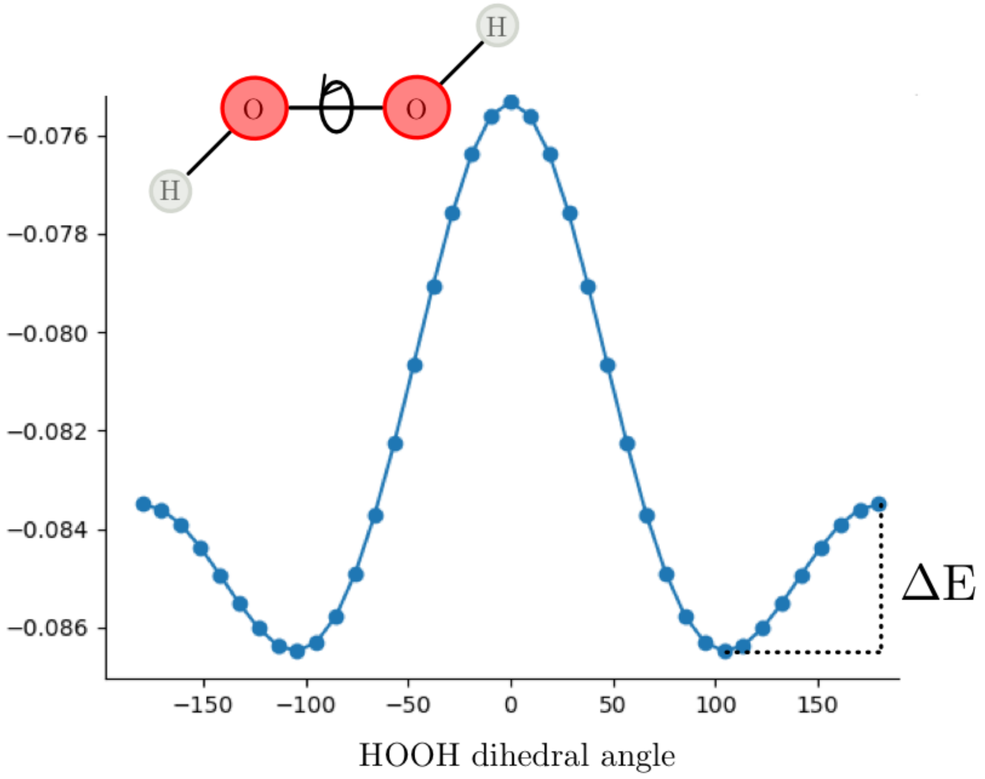
Surrogate model:
currenty around 100 electrons possible

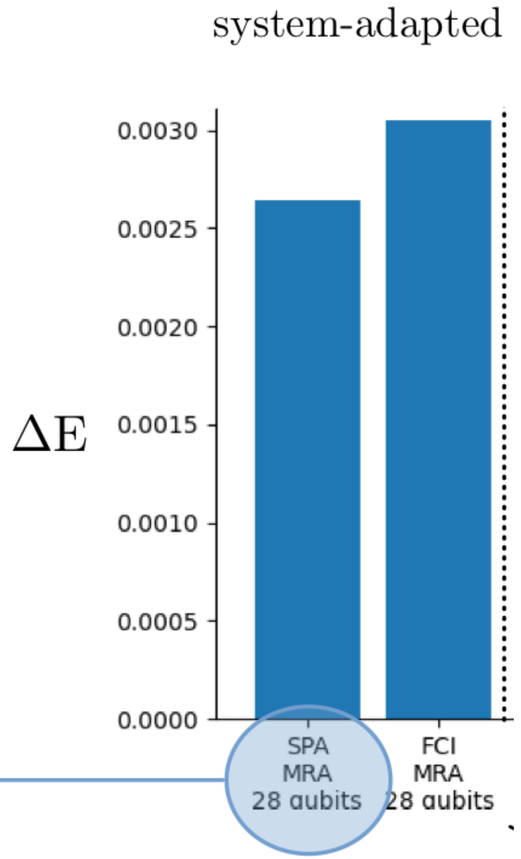
dependencies

high-level blogpost:
aspuru.substack.com/p/bits-are-cheap-and-qubits-expensive

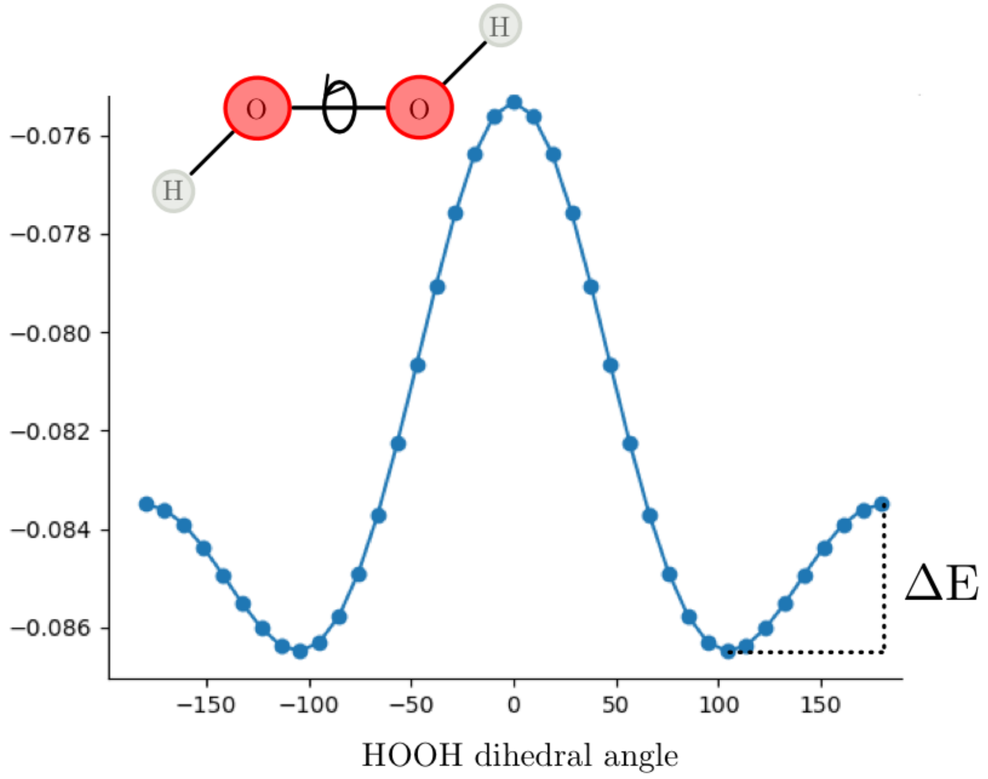
Example using everything

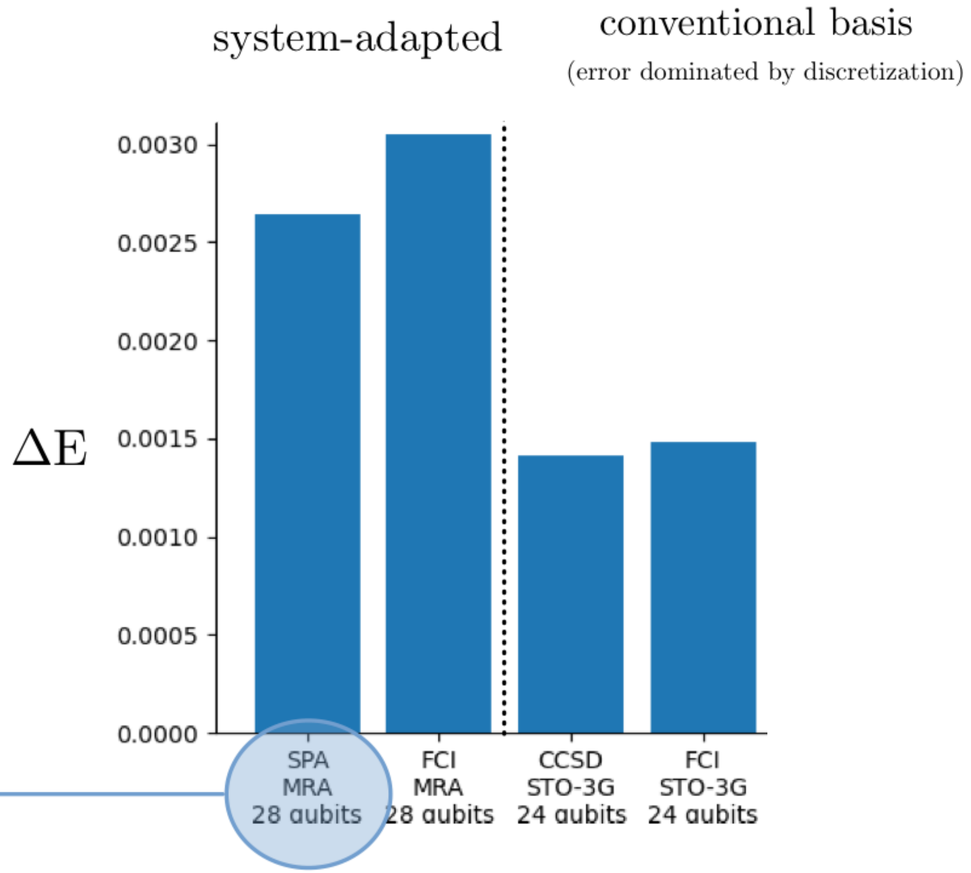

H
O
O
H
−0.076
−0.078
−0.080
−0.082
−0.084
−0.086
−150
−100
−50
0
50
100
150
HOOH dihedral angle
ΔE



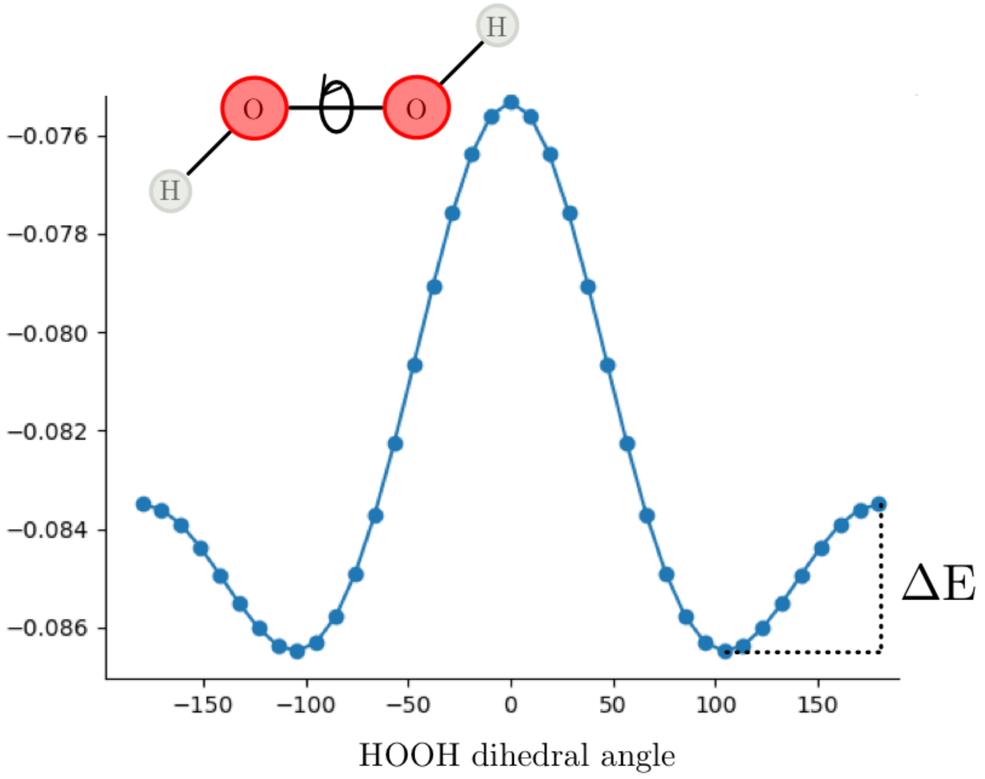


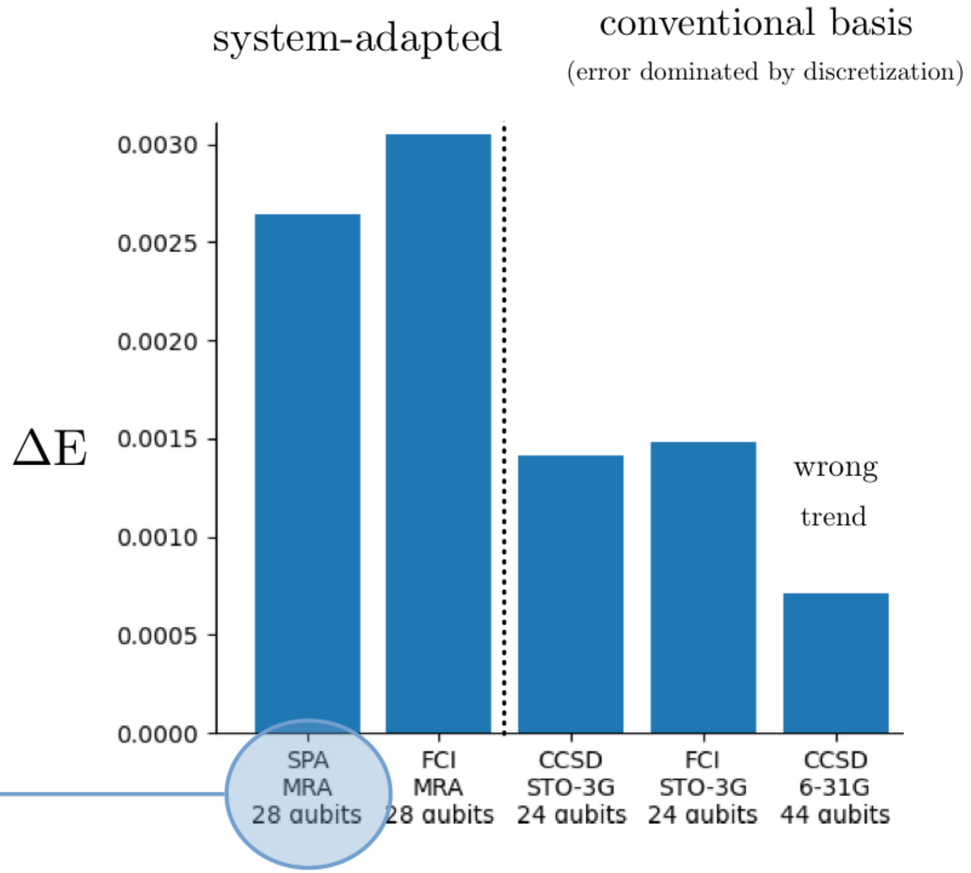

```
import tequila as tq

mol = tq.Molecule(geometry="h2o2.xyz")
H = mol.make_hamiltonian()
U = mol.make_ansatz(name="SPA")
E = tq.ExpectationValue(H=H, U=U)

result = tq.minimize(E)
```



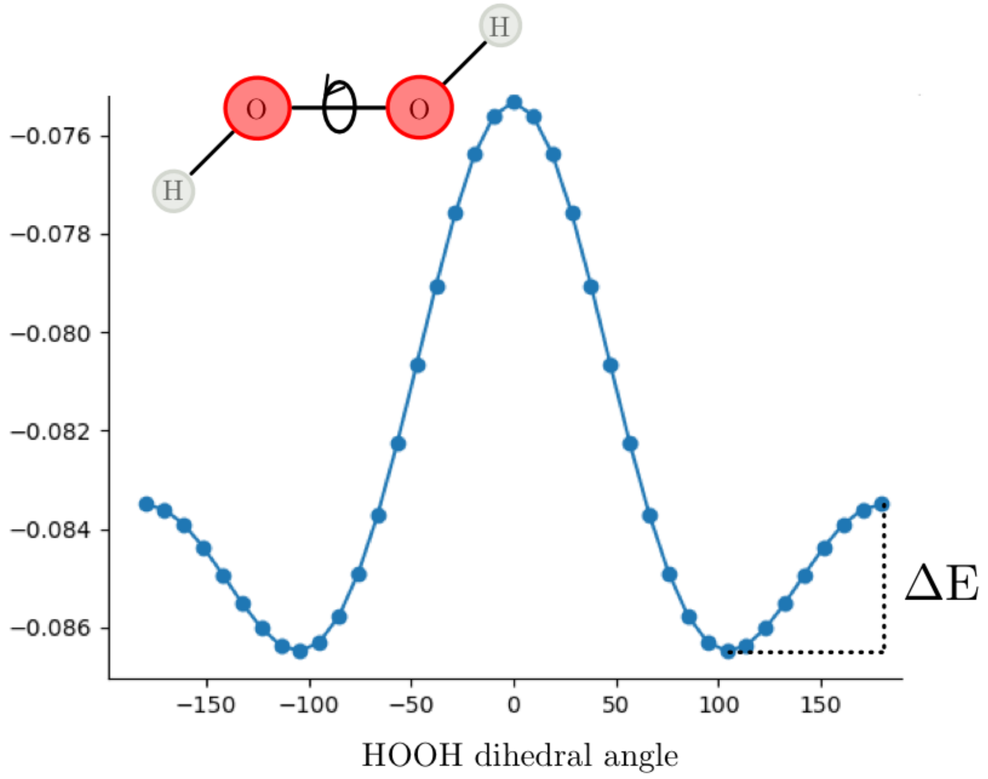


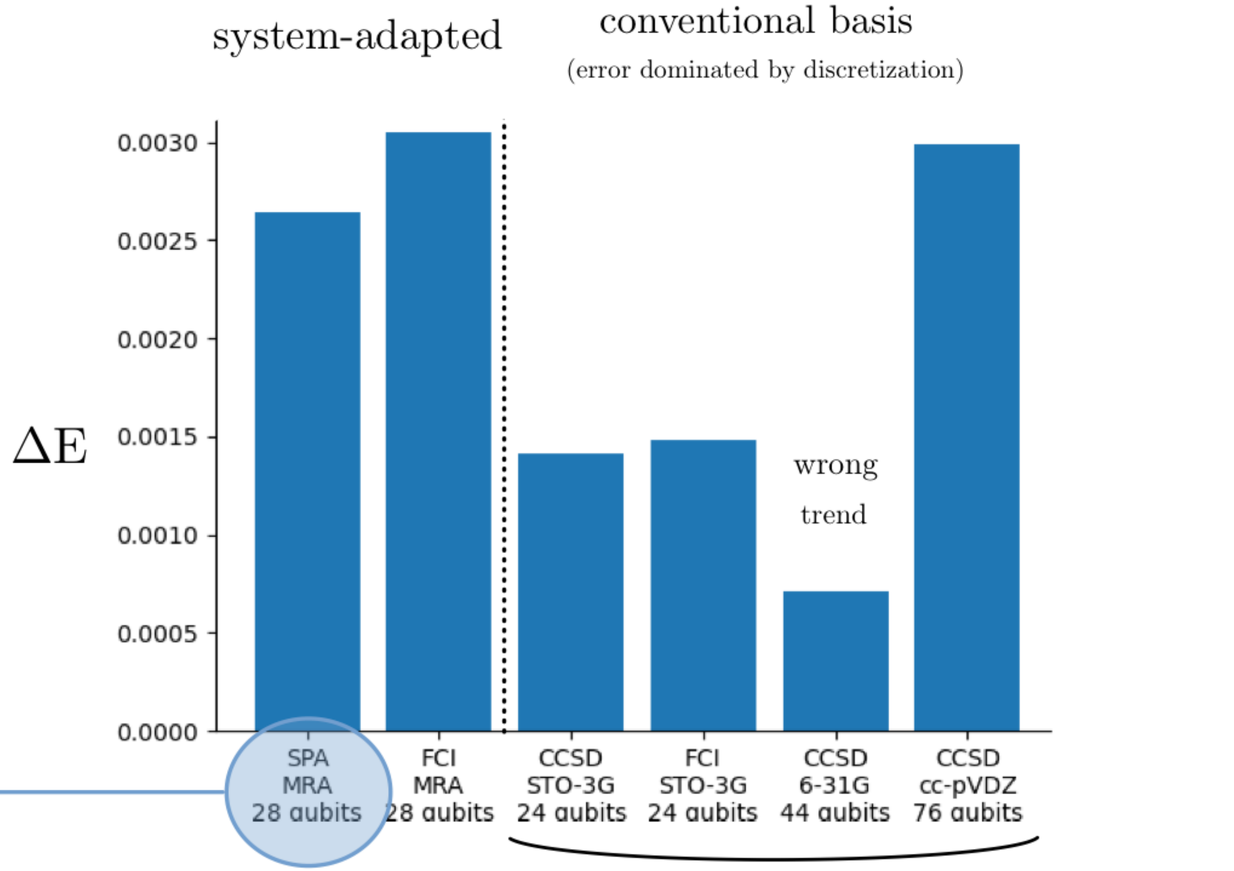

```
import tequila as tq

mol = tq.Molecule(geometry="h2o2.xyz")
H = mol.make_hamiltonian()
U = mol.make_ansatz(name="SPA")
E = tq.ExpectationValue(H=H, U=U)

result = tq.minimize(E)
```






```
import tequila as tq

mol = tq.Molecule(geometry="h2o2.xyz")
H = mol.make_hamiltonian()
U = mol.make_ansatz(name="SPA")
E = tq.ExpectationValue(H=H, U=U)

result = tq.minimize(E)
```






```
import tequila as tq

mol = tq.Molecule(geometry="h2o2.xyz")
H = mol.make_hamiltonian()
U = mol.make_ansatz(name="SPA")
E = tq.ExpectationValue(H=H, U=U)

result = tq.minimize(E)
```

Alán
Aspuru-Guzik
UofT

Philipp
Schleich
UofT/CS

Abhinav
Anand
UofT/Chem

Sumner
Alperin-Lea
UofT/Chem

Alba
Cervera-Lierta
Barcelona Supercomputing Center

Phillip
Jensen
UofT/Chem

Thi Ha
Kyaw
UofT/CS

Teresa
Tamayo-Mendoza
Harvard

Mario
Krenn
MPI Erlangen

Zi-Jian
Zhang
UofT/CS

UNIVERSITY OF TORONTO

the matter lab

qosf

Initial Team: Sumner Alperin-Lea, Alba Cervera-Lierta, Teresa Tamayo-Medoza, Cyrille Lavigne

AAG-Group (UofT): Philipp Schleich, Abhinav Anand, Matthias Degroote, Skylar Chaney, Maha Kesibi, Naomi Grace Curnow, Arkaprava Choudhurry

Izmaylov-Group (UofT): Tzu-Ching "Thomson" Yen, Vladyslav Verteletskyi, Zachary Bansingh

QOSF: Brandon Solo, Giorgios Tsilimigkounaikis,Claudia Zendeja-Morales, Tanya Garg

Github: Alejandro de la Serna, Leo Becker, David Wierichs, Arianne van den Griend ..... You?

DS3Lab (ETH): Maurice Weber

tequila

github/tequilahub