Stochastic Gradient Descent for Hybrid Quantum-Classical Optimization

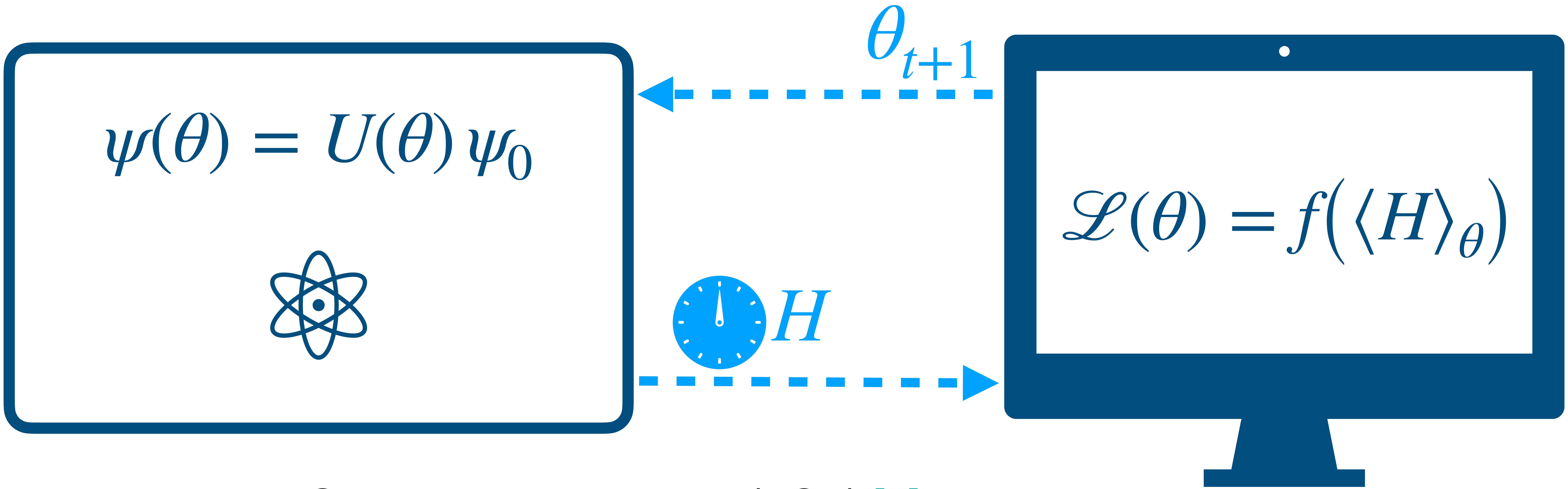
1910.01155

Center for Theoretical Physics of the Polish Academy of Science, July 2020

Frederik Wilde
frederikwil.de/slides/pas2020

Freie Universität Berlin

Hybrid quantum-classical algorithms



- Variational Quantum Eigensolver (VQE) [1]
  $\mathscr{L}(\theta) = \langle \psi(\theta) \,|\, H \,|\, \psi(\theta) \rangle$
- Quantum Approximate Optimization Algorithm (QAOA) [2]
  $|\,\psi(\boldsymbol{\beta},\boldsymbol{\gamma})\rangle = \mathrm{e}^{-\mathrm{i}\beta_p X}\,\mathrm{e}^{-\mathrm{i}\gamma_p H}\,\ldots\,\mathrm{e}^{-\mathrm{i}\beta_1 X}\,\mathrm{e}^{-\mathrm{i}\gamma_1 H}\;|+\rangle \qquad X = -\sum_i \sigma_i^x$

[1] A. Peruzzo et al., Nature Com. 2014 [2] Farhi et al., 1411.4028

Hybrid methods are promising for NISQ devices

- Ideally: Only do the quantum-easy-classically-hard part on the quantum device
- Variational principle has a long history
- Short coherence times are OK due to iterative process
- (Implicit) error mitigation by the classical optimizer [1]

[1] L. Cincio et al., 2007.01210

Optimization

0-th order:

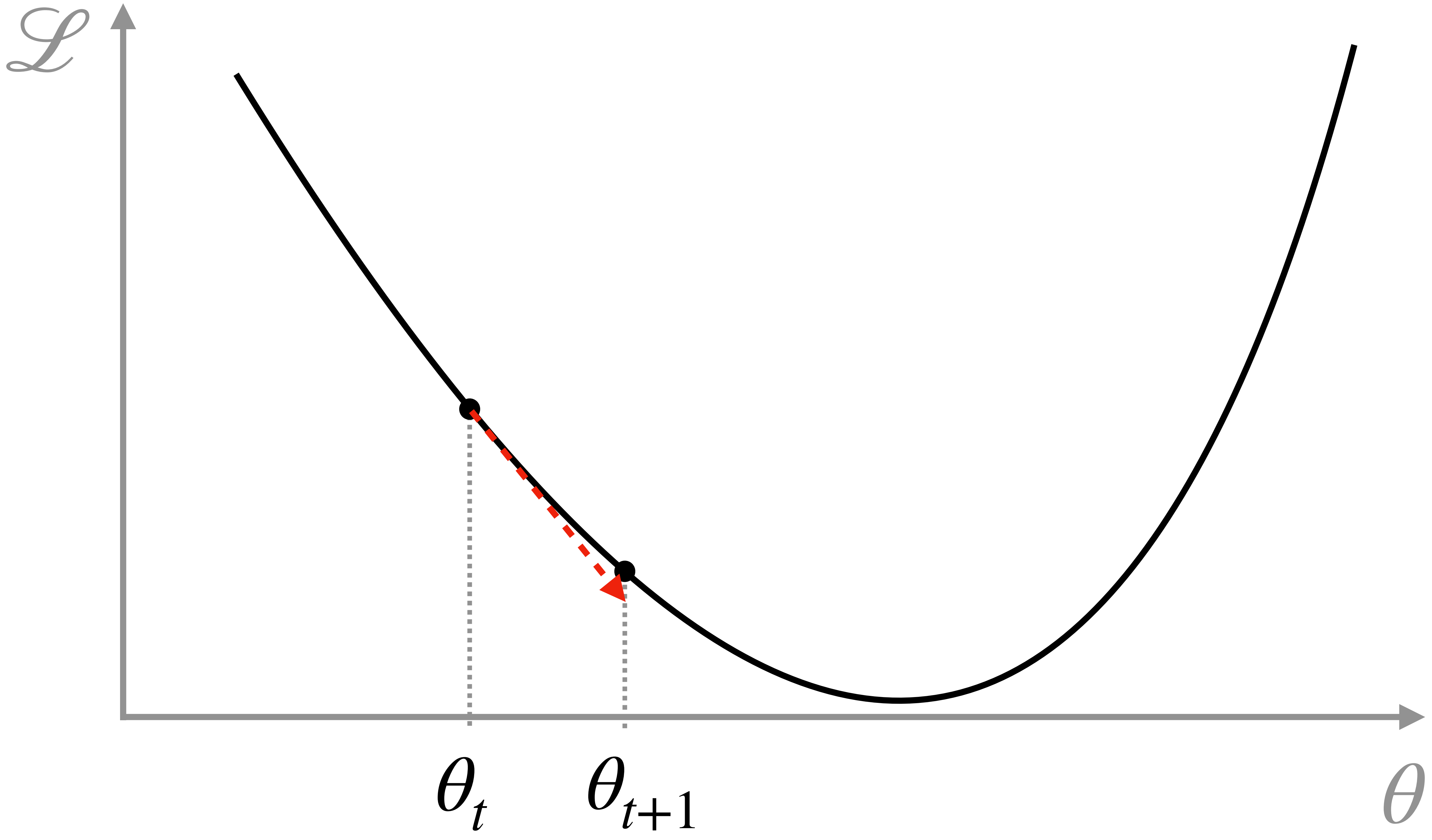
- SPSA (simultaneous pertubation stochastic approximation) and RSGF combined with ADAM
- swarm optimization
- genetic algorithm
- scikit-quant.org [2]

[good review] M. Benedetti et al., Quantum Sci. Technol. 2019
[2] W. Lavrijsen et al., 2004.03004

Optimization

1st order:

Gradient descent method: $\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathscr{L} \Big|_{\theta_t}$

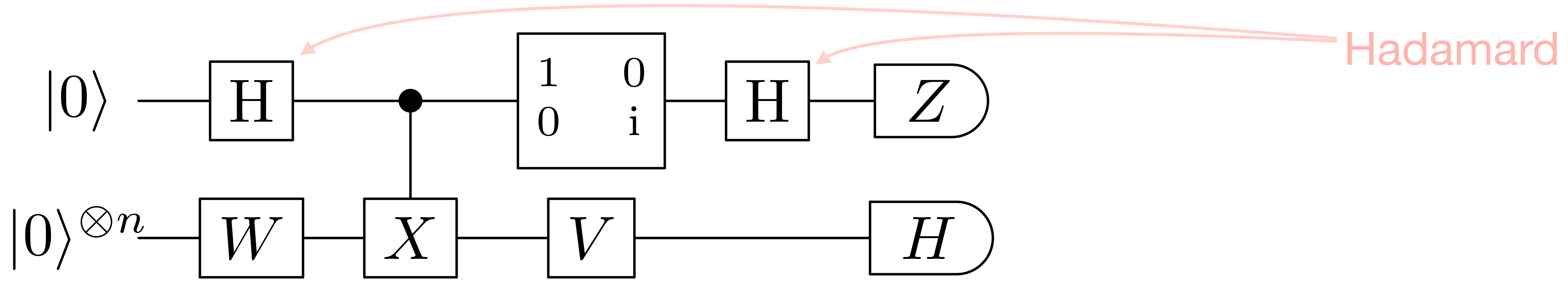


Optimization

1st order:

- gradient is expensive (no-cloning thm, no autograd)
- "exact" gradient is accessible
- gradient as a circuit [1] $Ve^{-\mathrm{i}\theta X}\tilde{W} = VW, \quad |\psi\rangle = VW|0\rangle$

$$\partial_\theta\langle\psi|H|\psi\rangle = 2\,\Im\langle\psi|H\,VXW|0\rangle = 2\,\Im\langle\psi|H\,VXV^\dagger|\psi\rangle$$



- Parameter-shift rule [2,3] $\partial_\theta\langle H\rangle = \langle H\rangle_{\theta+\pi/4} - \langle H\rangle_{\theta-\pi/4}$

[1] G. G. Guerreschi et al., 1701.01450 [2] M. Schuld et al., PRA 2019 [3] L. Banchi et al., 2005.10299

Sources of stochasticity

$x_i \in \mathbb{R}^{3N}$	$y_i \in \{0,1\}$
	0
	1
	0

Some data

$$\mathscr{L}(\theta) = \sum_i \left[\left\langle \sum_j h_j \right\rangle_{(x_i,\, \theta)} - y_i \right]^2$$

1. Measurement shots ← hardware-dependent (shots might actually be cheap)
2. Observable components ← non-commuting components
3. Data
4. Parameter-shift terms ← dependent on circuit architecture (at least factor of 2)

Sources of stochasticity

$$\mathscr{L}(\theta) = \sum_i \left[\left\langle \sum_j h_j \right\rangle_{(x_i,\, \theta)} - y_i \right]^2$$

$$\partial_\theta \mathscr{L} = \sum_i 2 \left[\sum_j \langle h_j \rangle_{(x_i,\, \theta)} - y_i \right] \sum_j \frac{1}{2} \left(\langle h_j \rangle_{(x_i,\, \theta + \frac{\pi}{2})} - \langle h_j \rangle_{(x_i,\, \theta - \frac{\pi}{2})} \right)$$

Fast QAOA optimization

$$|\psi(\boldsymbol{\beta},\boldsymbol{\gamma})\rangle = e^{-i\beta_p X} e^{-i\gamma_p H} \dots e^{-i\beta_1 X} e^{-i\gamma_1 H} \,|+\rangle$$

$$e^{-i\beta_p X} = e^{i\beta_p \sigma_1^x} \dots e^{i\beta_p \sigma_n^x}$$

$2m$ parameter shift terms

$2n$ parameter shift terms

Sampling PS terms alone reduces #(circuit evaluations) by a factor of $\dfrac{2p}{2pn + 2pm} = \dfrac{1}{n+m}$ per optimization step.

Convergence

Problem: $\mathbb{E}(\mathscr{L}(X)) \neq \mathscr{L}(\mathbb{E}(X))$

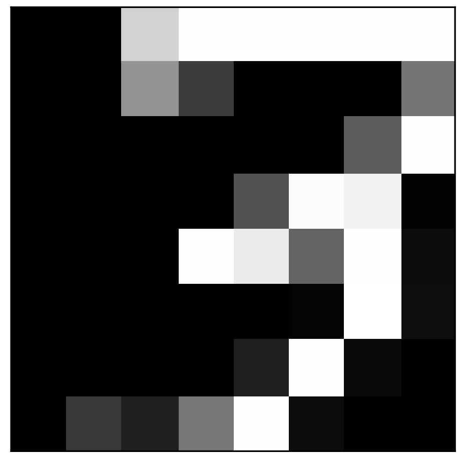
- Solved for polynomial loss functions

$$\mathscr{L}(X) = \sum_{j=0}^{k} a_j X^j = \mathscr{L}(X_1, \ldots, X_k)$$

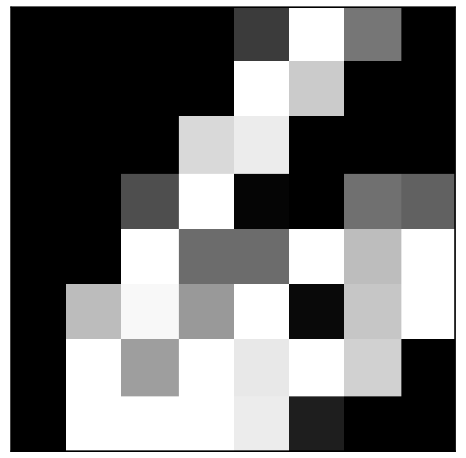
$$e_k\big(\mathscr{L}(X)\big) = \frac{1}{k!} \sum_{i_1,\ldots,i_k \in \mathrm{P}\{1,\ldots,k\}} \mathscr{L}(x_{i_1}, \ldots, x_{i_k})$$

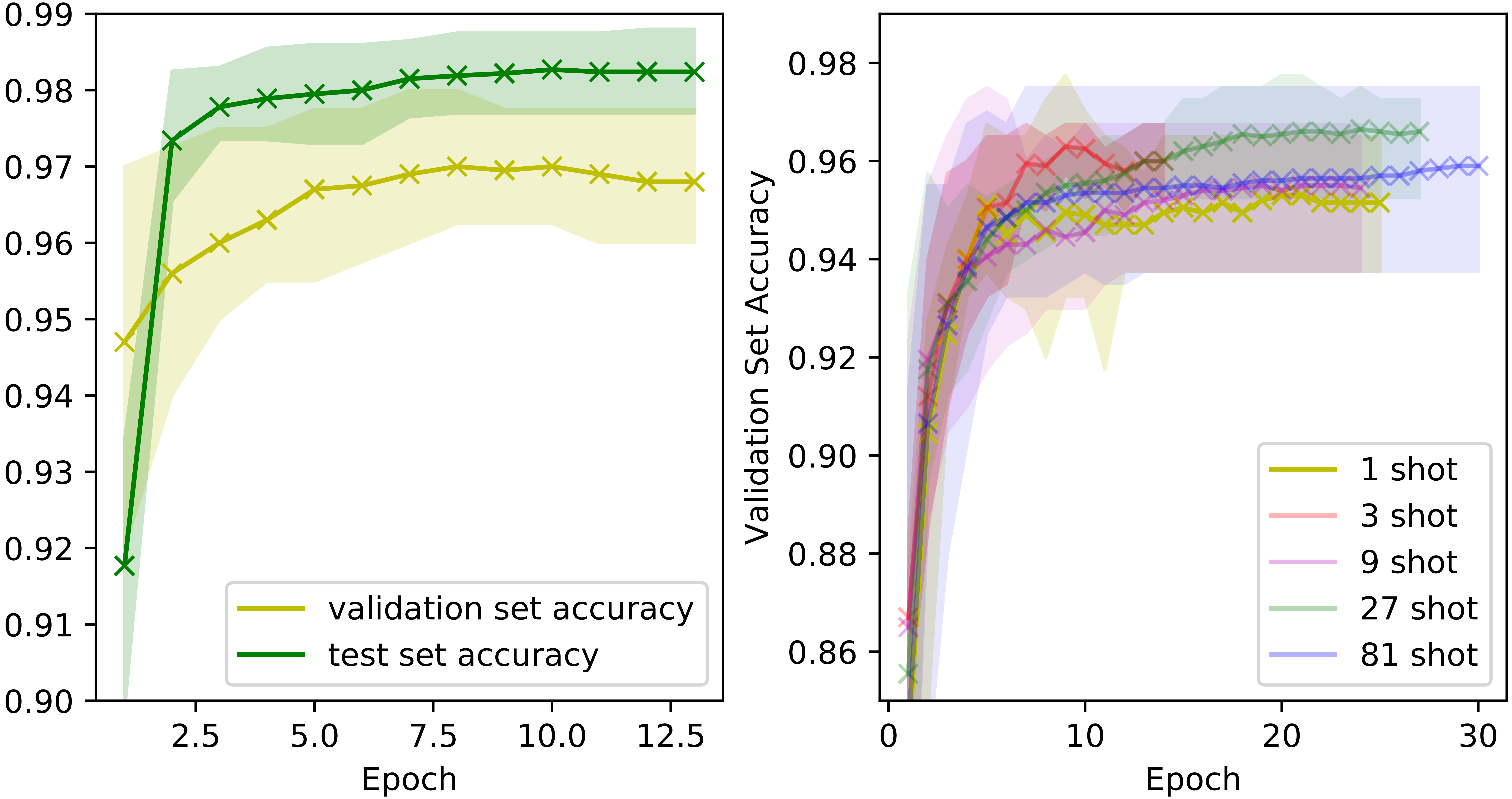
- Provable convergence under certain assumptions about $\mathscr{L}$ [1]
  - Polyak-Lojasiewicz (PL) inequality "no local minima"
  - Lipschitz continuity

[1] M. M. Wolf, "Mathematical foundations of supervised learning"



MNIST Classifier





8 qubits, 400 parameters, batch size = 1

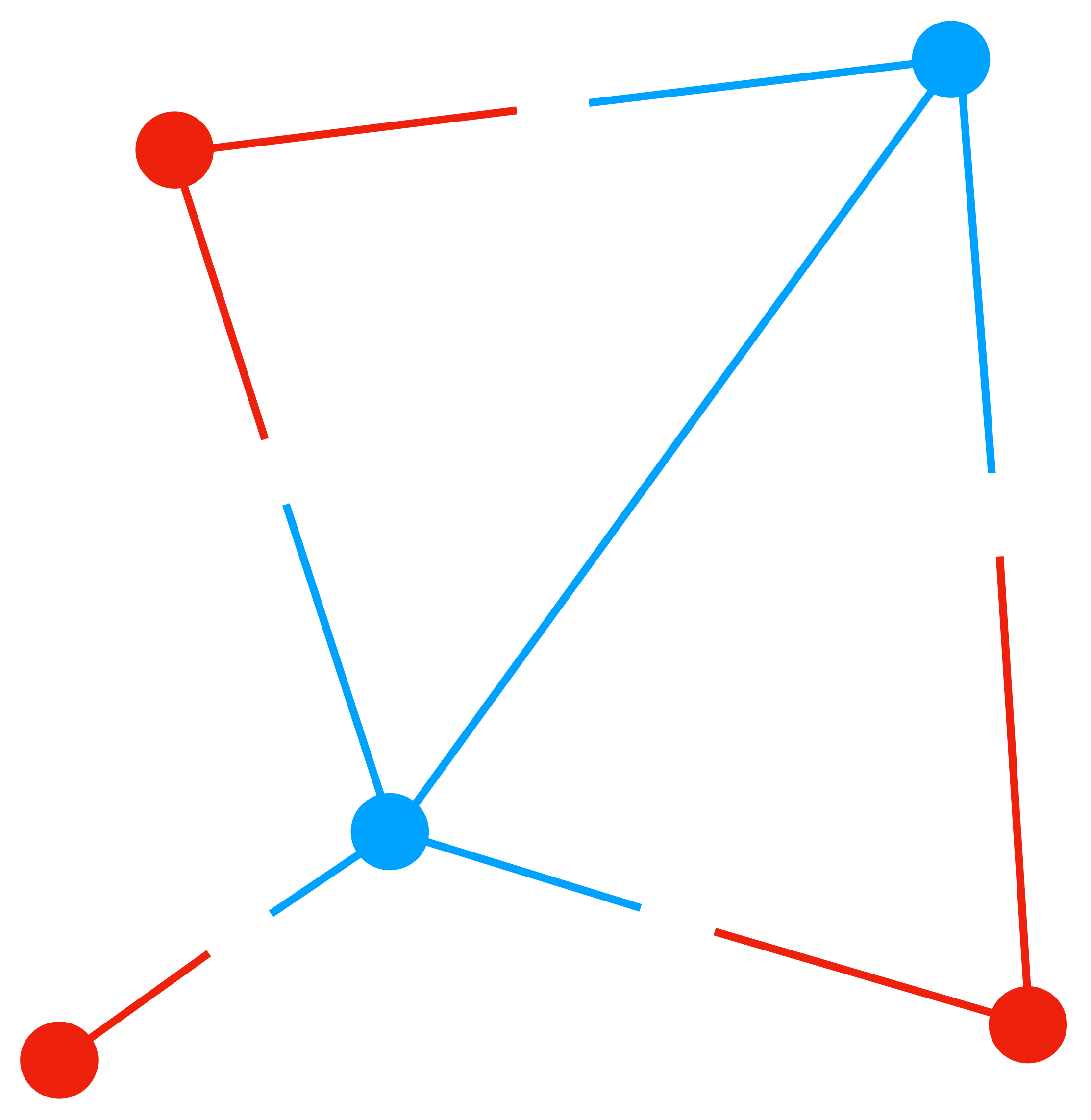
MAXCUT with QAOA

- Given a graph $G = (V, E)$ divide V into two subsets, s.t. the number of edges between those is maximal. $\leftarrow$ NP-hard

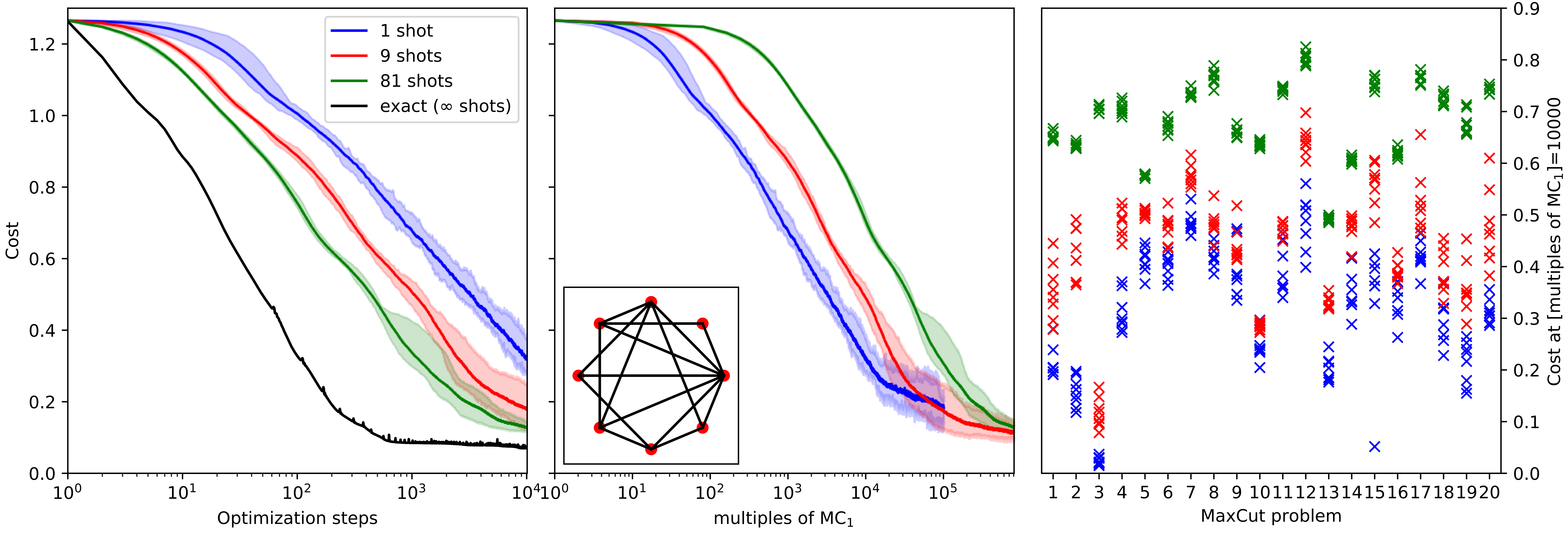
$$H = \sum_{(i,j) \in E} \sigma_i^z \sigma_j^z$$

- QAOA:

$$|\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle = \mathrm{e}^{-\mathrm{i}\beta_p X} \mathrm{e}^{-\mathrm{i}\gamma_p H} \dots \mathrm{e}^{-\mathrm{i}\beta_1 X} \mathrm{e}^{-\mathrm{i}\gamma_1 H} \; |+\rangle$$



MAXCUT with QAOA



p=50, random graphs $|V|=8, |E|=16$

Number of shots is a hyper parameter.

Alternatives:
[Pennylane](#) (default.qubit.tf device)
[Yao.jl](#)
[Tensorflow Quantum](#)

qradient

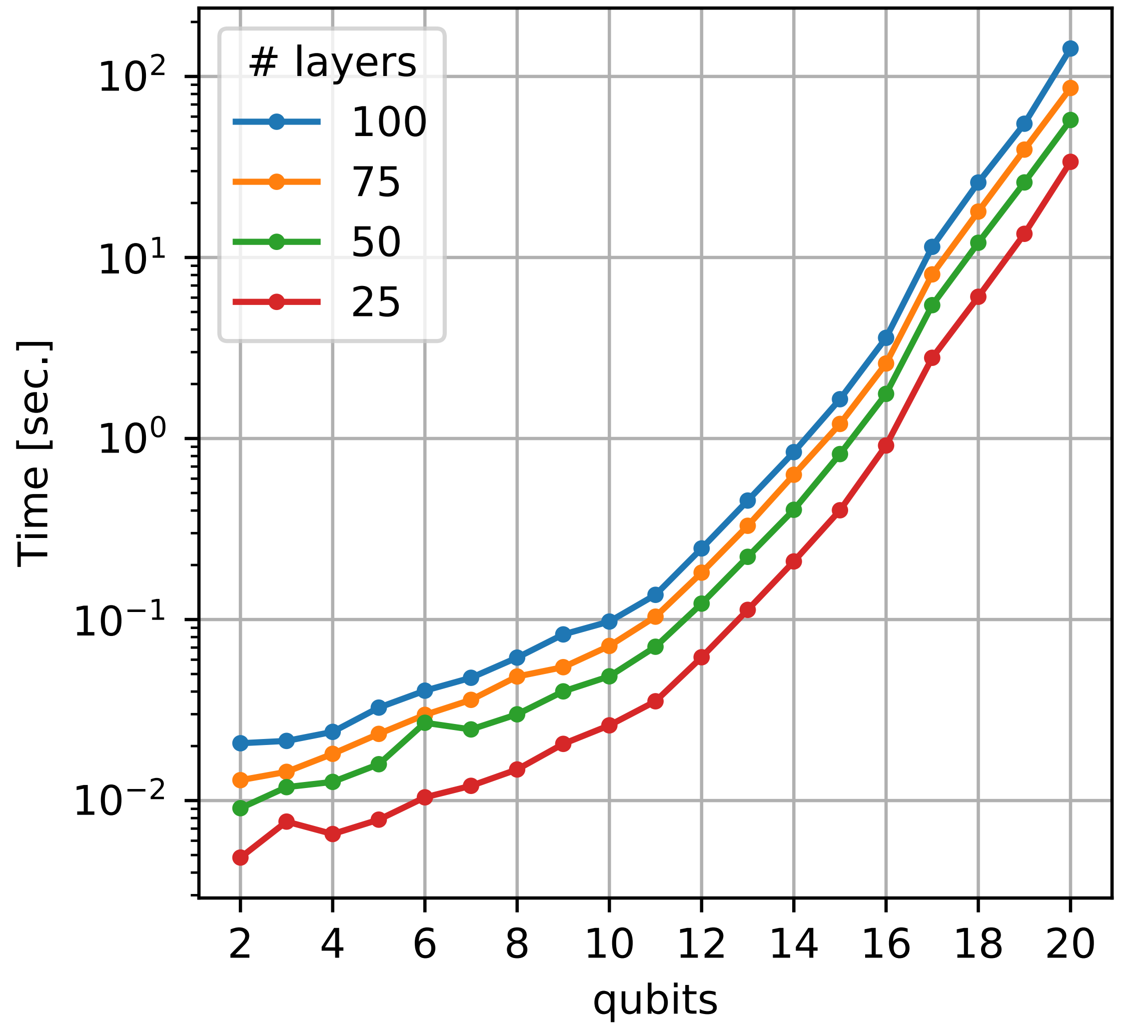
github.com/frederikwilde/qradient

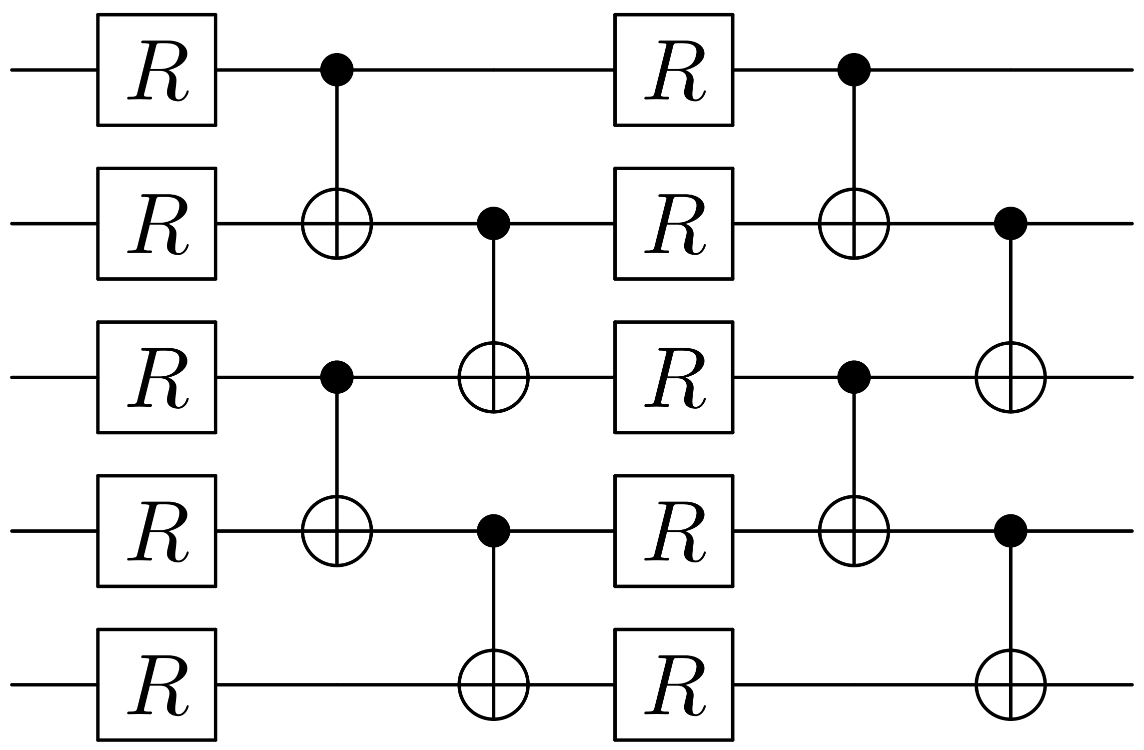
calculate gradients of quantum circuits efficiently (in simulation!)

```
import numpy as np
from qradient.circuit_logic import RandomRotations
```

```
interactions = np.array([[None, 1.  , None],
                         [None, None, -1. ],
                         [None, None, None]])
observable = {'zz': interactions}
axes = np.array([[0, 0, 0],
                 [1, 2, 1]])
```

```
circuit = RandomRotations(observable, axes)
expectation_value, gradient = circuit.gradient(angles)
```





Disclaimer:  Under construction!

currently the above applies to the 'restructure' branch
contact me for any questions :-)

Related work

- Rigorous separation between 0th and 1st order optimization [1]
- iCANS [2]
  individual Coupled Adaptive Number of Shots
  $$\#(\text{shots}) = \frac{2L\alpha}{2 - L\alpha}\frac{\hat{\text{Var}}(g_i)}{g_i^2}$$
- Rosalin [3]
  Cleverly distribute "shot budget" amongst Hamiltonian terms (randomly!) to decrease the estimator's variance.

[1] J. Napp et al., 1901.05374
[2] J. M. Kübler et al., Quantum 2020
[3] A. Arrasmith et al., 2004.06252

What now?

- Non-polynomial loss functions
- Beyond parameter shift rule
- #(measurement shots) and barren plateaus
- Impact of noise [1]
  Robustness through the parameter shift rule? [2]

[1] W. Lavrijsen et al., 2004.03004
[2] J. J. Meyer et al., 2006.06303

Thanks for your attention

Co-authors:

Ryan Sweke
Freie Universität Berlin

Johannes Jakob Meyer
Freie Universität Berlin

Maria Schuld
Xanadu Inc. and University of KwaZulu-Natal South Africa

Paul K. Fährmann
Freie Universität Berlin

Barthélémy Meynard-Piganeau
Ecole Polytechnique Paris

Jens Eisert
Freie Universität Berlin

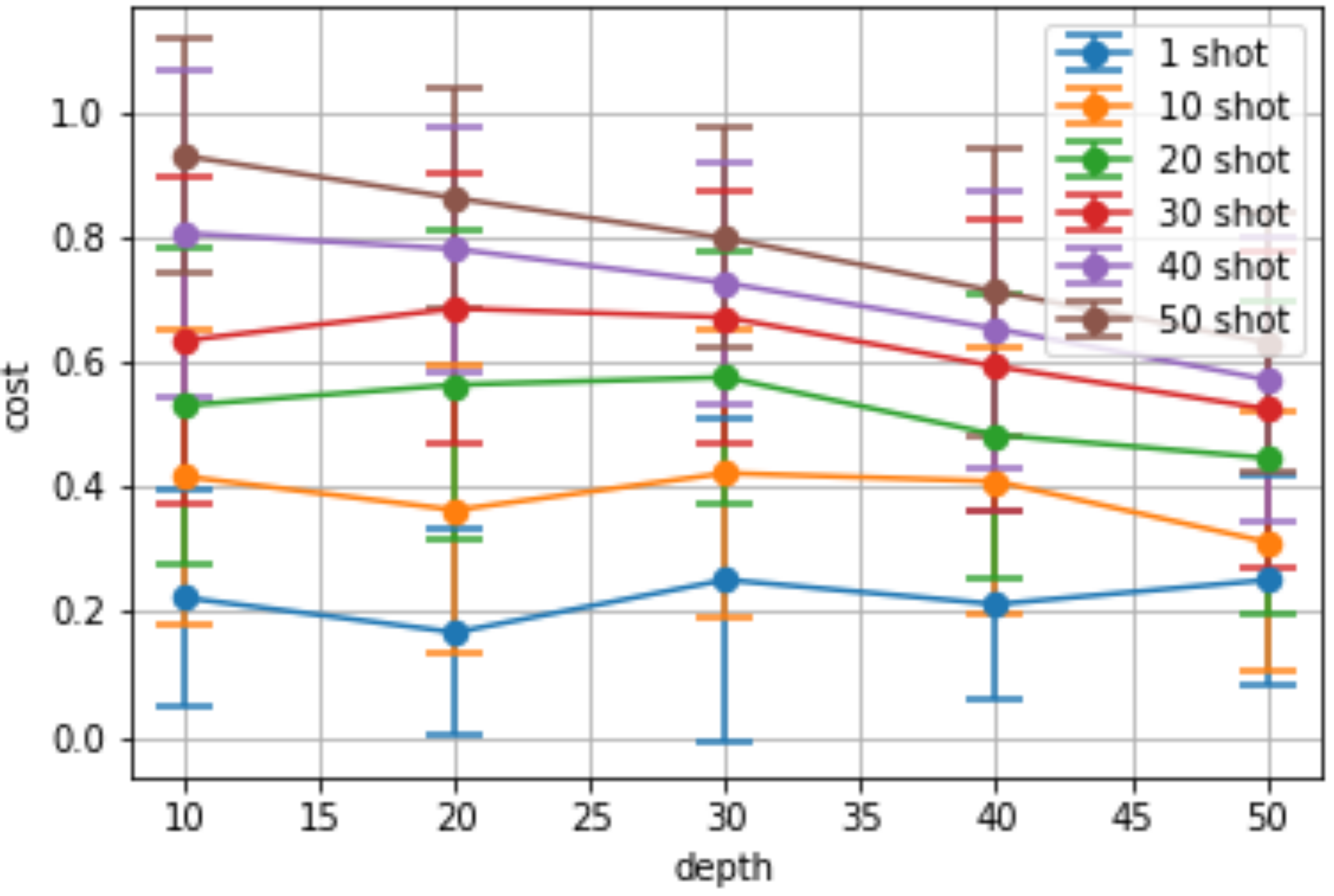
arxiv: 1910.01155

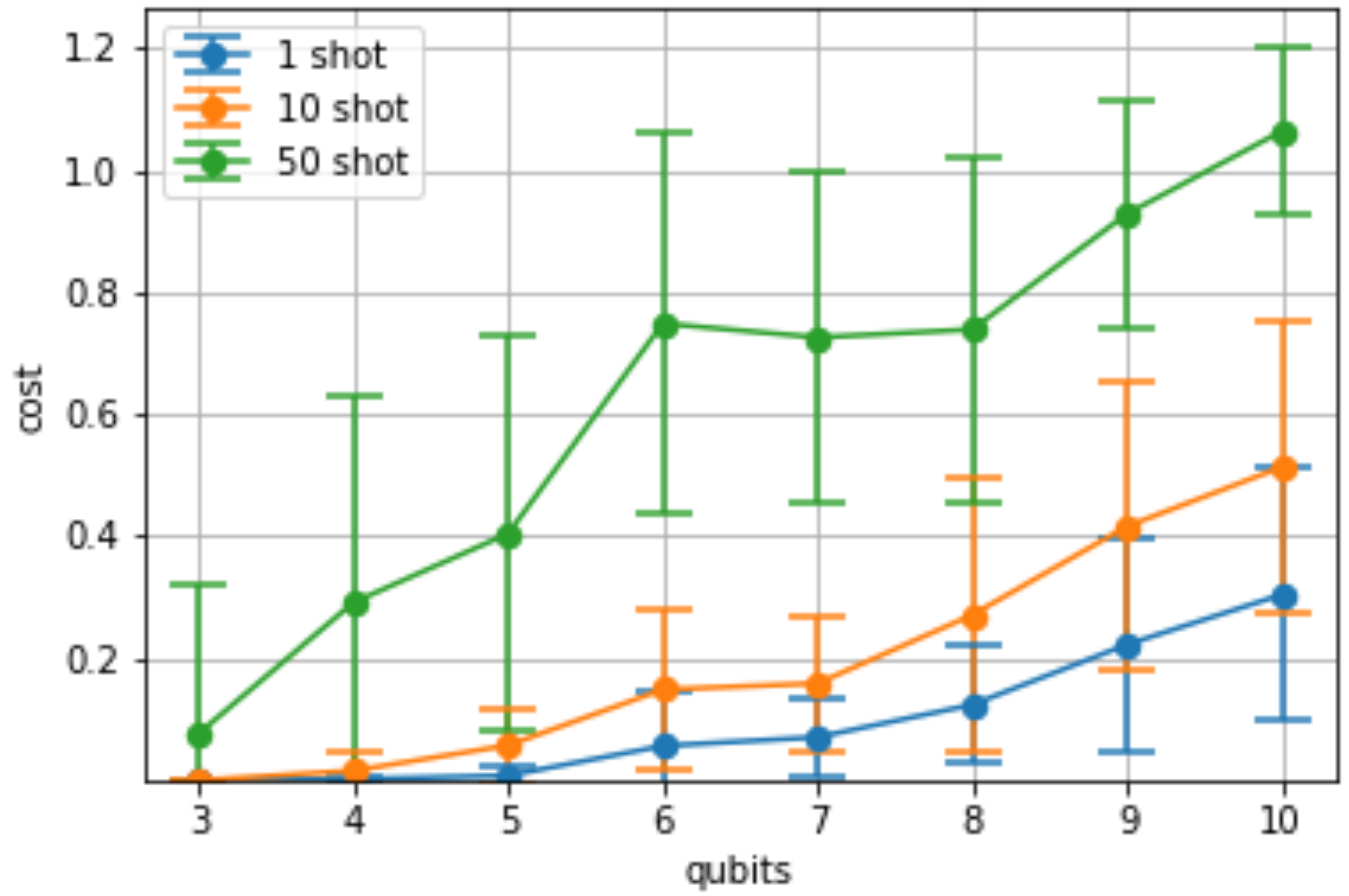
Slides at:
`frederikwil.de/slides/pas2020`

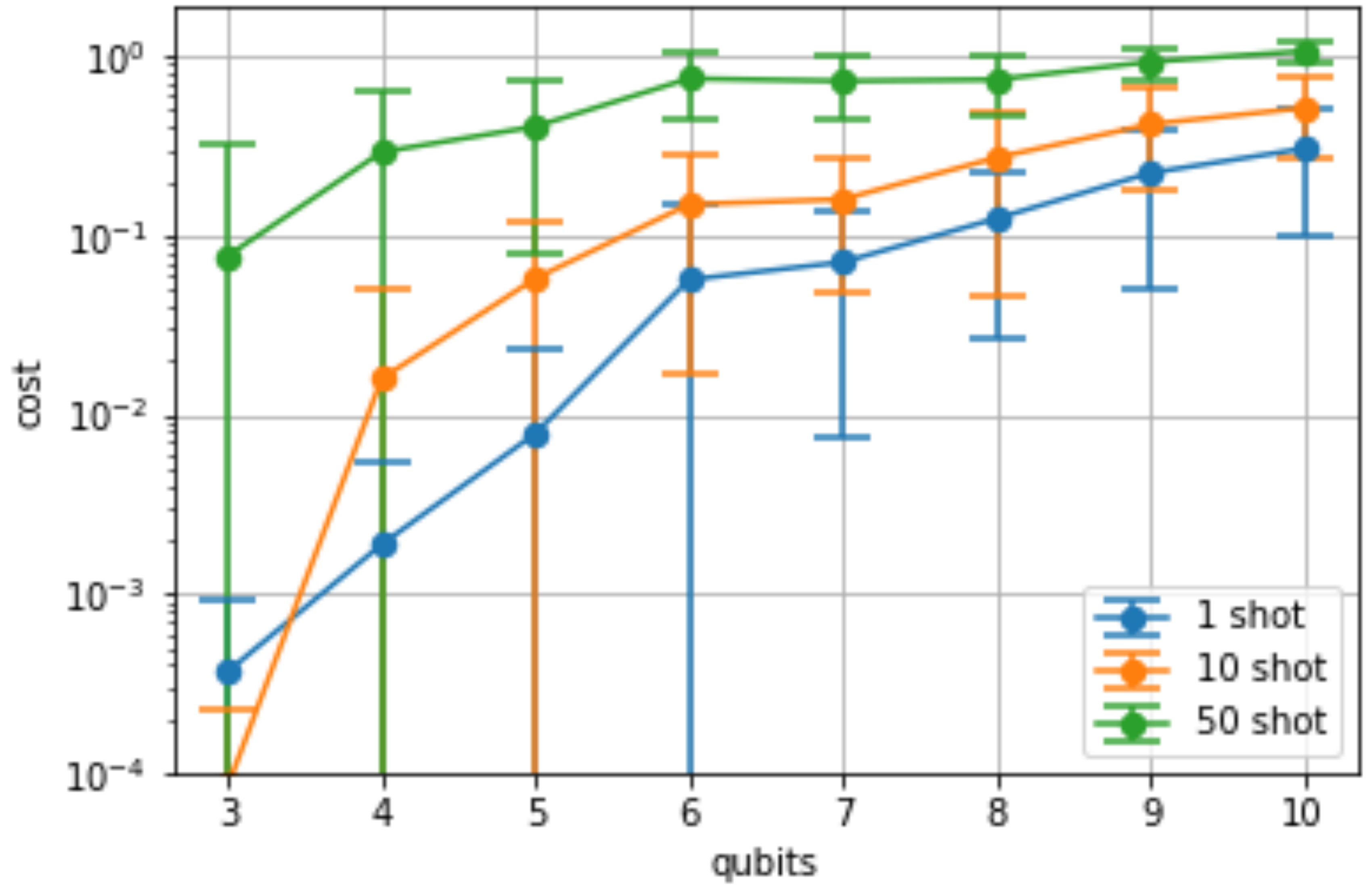
`f.wilde@fu-berlin.de`

Scaling

QAOA on Erdős–Rényi graphs (edge probability 30%) ~20 samples/data-point (unpublished)







Corrections for polynomial loss functions

- Let X represent the measurement
- Expand $\mathscr{L}(X)$ around $\mathbb{E}(X) = x_0$
- $\mathscr{L}(X) = \mathscr{L}(x_0) + \mathscr{L}'(x_0)\,(X - x_0) + \sum_{n=2}^{s} \frac{1}{n!} \mathscr{L}^{(n)}(x_0)\,(X - x_0)^n$
- $\mathbb{E}\left[\mathscr{L}(X)\right] = \mathbb{E}\left[\mathscr{L}(x_0)\right] + \sum_{n=2}^{s} \frac{1}{n!} \mathscr{L}^{(n)}(x_0)\,\mathbb{E}\left[(X - x_0)^n\right]$