Towards Computational Fluid Dynamics with Tensor-Programmable Quantum Circuits

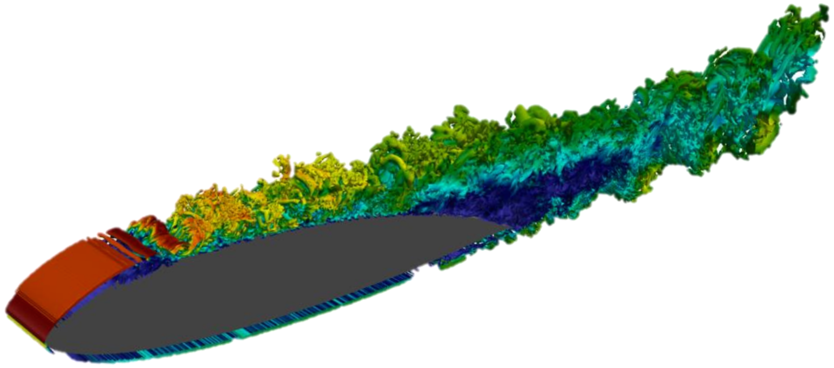
AQMCSE, 2025/10/09

Pia Siegl, Greta Reese, Nis van Hülst, Tomohiro Hashizume, Dieter Jaksch

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

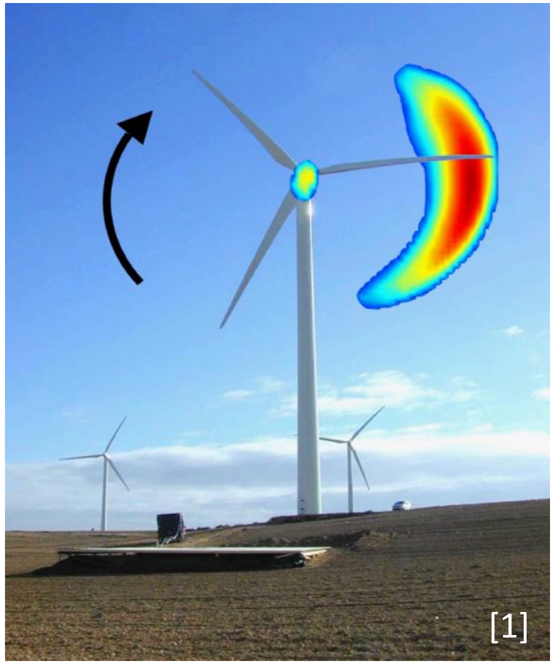
DLR

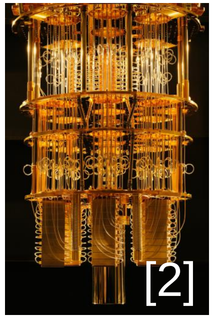
Computational Fluid Dynamics (CFD)



fluid dynamics:

- described non-linear differential equations
- often requires expensive numerical schemes
- discretization in space and time



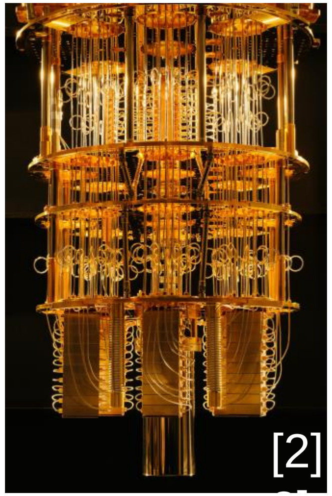


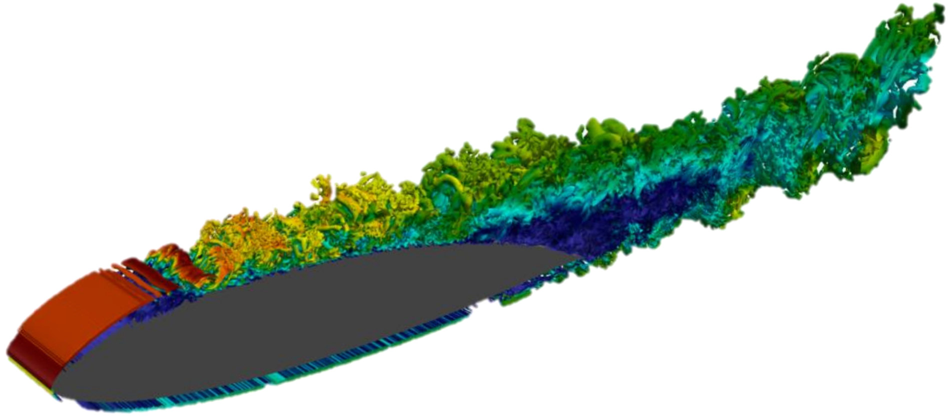
goal on quantum computers:

- encoding field solution in exponentially growing state space
- efficient applications of operators

[1] Oerlemans, et al., J. **Sound Vib. 299**, 869, (2007)
[2] IBM-research_kryostat.jpg (higgs.ch)

Why is CFD Hard on a Quantum Computer?





- normalized state vectors
- unitary (and linear) operations

- non-normalized fields
- non-unitary operators
- non-linear operations

[2] IBM-research_kryostat.jpg (higgs.ch)

Outline

Basic Method:

1. How to encode the field?
2. How to perform time evolution?
3. How to map the operators?

variational quantum algorithm

- parametrized encoding of the field
- “tensor-programmed” operators
- training of next time-step

Application to linear PDEs:

1. advection-diffusion equation
2. training landscapes

- good agreement with classical solution
- well trainable parameters

Extension to non-linear PDEs:

1. circuit extension
2. application to non-linear Burgers’ equation

- weighted combination of linear and non-linear terms
- good agreement with classical solution

Field Encoding

solution at time step j

$$f^{j+1} = \hat{O} f^{j}$$

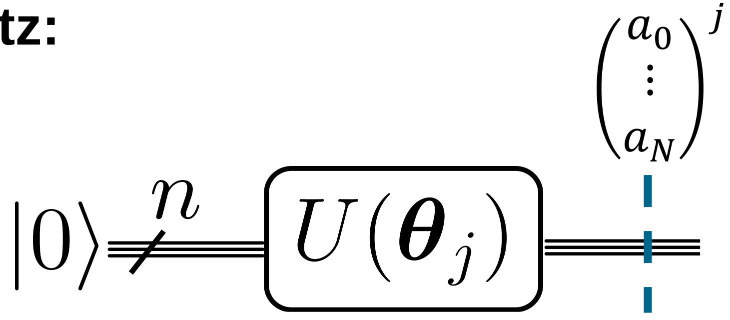
unitary evolution operator

$$f^{j} \rightarrow \begin{pmatrix} f_0 \\ \vdots \\ f_N \end{pmatrix}^{j}, \hat{O} \rightarrow \begin{pmatrix} o_{11} & o_{12} & \cdots & o_{1N-1} & o_{1N} \\ o_{21} & o_{22} & & o_{2N-1} & o_{2N} \\ \vdots & & \ddots & \vdots & \\ o_{N-11} & o_{N-12} & \cdots & & o_{N-1N} \\ o_{N1} & o_{N2} & \cdots & o_{1N-1} & o_{NN} \end{pmatrix}$$

amplitude encoding:

$$f^{j} : \begin{pmatrix} f_0 \\ \vdots \\ f_N \end{pmatrix} = \boxed{\theta_j^0} \begin{pmatrix} a_0 \\ \vdots \\ a_N \end{pmatrix}$$

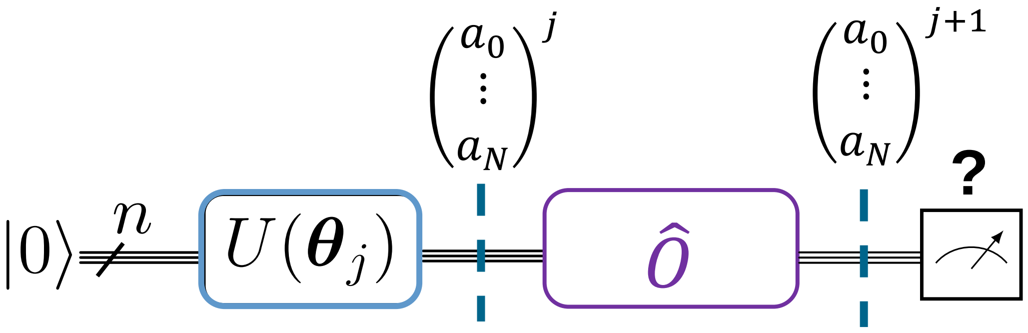
using a **parametrized ansatz:**



Variational Time Stepping

example: one iteration step $f^{j+1} = \hat{O} f^{j}$

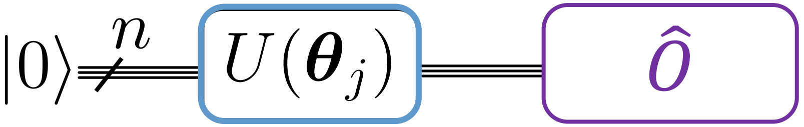
unitary



[3] we do:

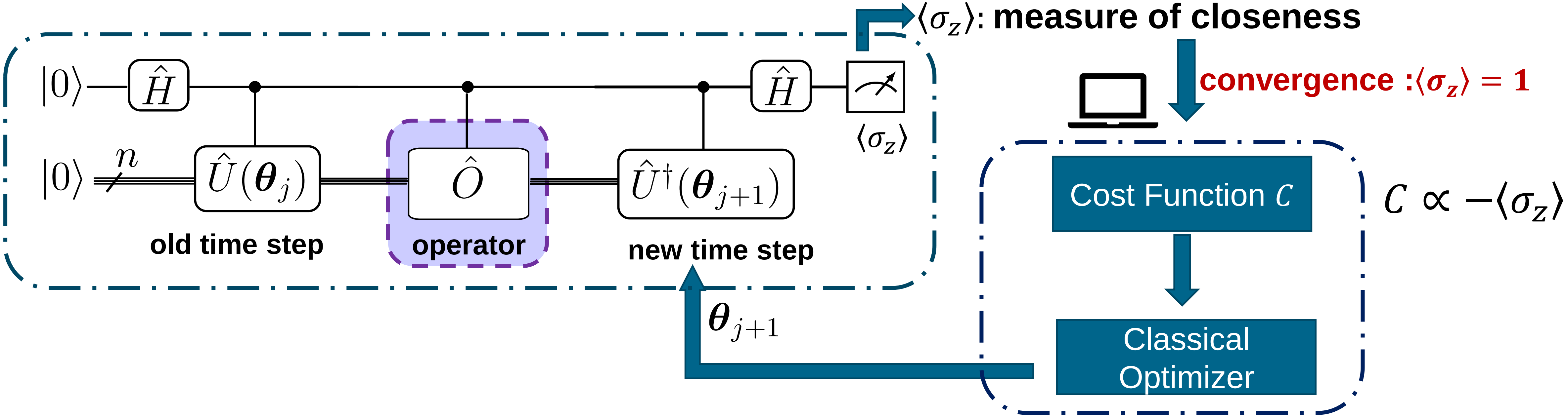
create a **variational ansatz** $\boxed{U^{\dagger}(\boldsymbol{\theta}_{j+1})}$

compare with



[3] M. Lubasch et al., **Phys. Rev. A 101**, (2020)

Hadamard Test for Comparison



➢ **repeat for each time step**

advantages of variational time stepping:

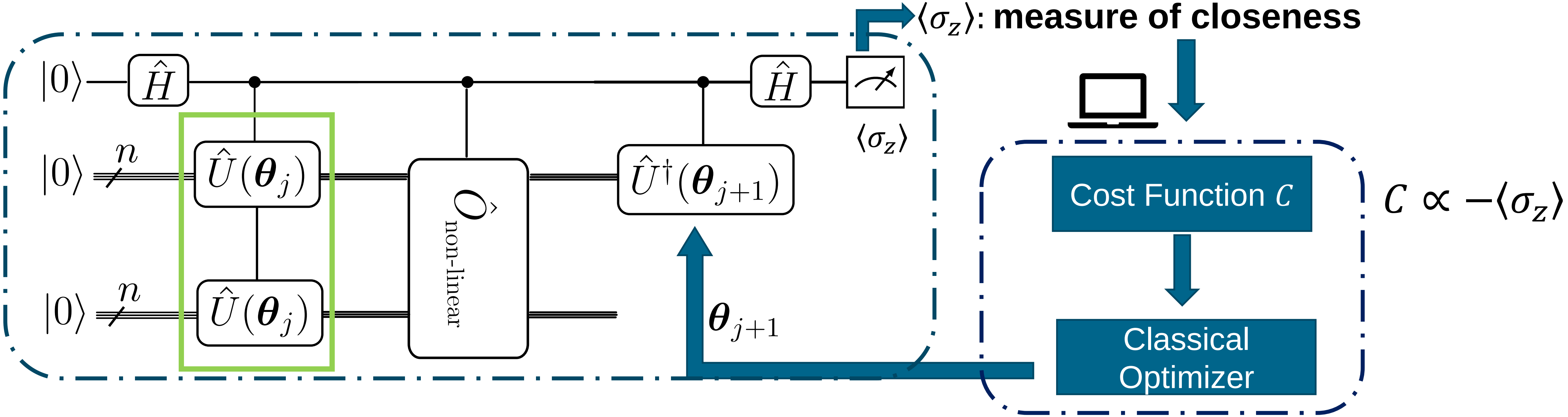
➢ computation of time-dependent observables

➢ shallow circuits

➢ incorporation of non-linearities of type $f(\hat{O}f)$

[3] M. Lubasch et al., **Phys. Rev. A 101**, (2020)

Hadamard Test for Comparison



- **repeat for each time step**

advantages of variational time stepping:
- computation of time-dependent observables
- shallow circuits
- incorporation of non-linearities of type $f(\hat{O}f)$

[3] M. Lubasch et al., **Phys. Rev. A 101**, (2020)

Challenges of Variational Time Stepping

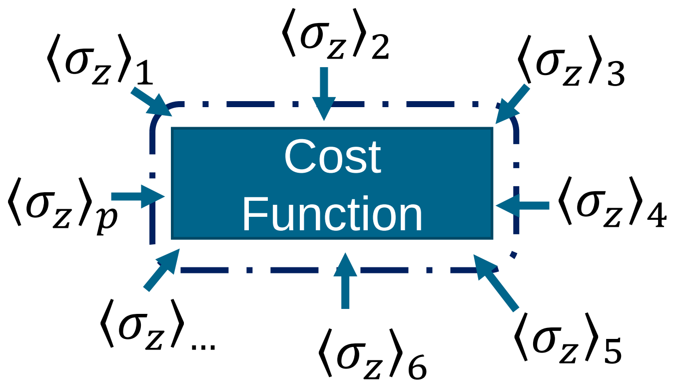
so far : $f^{j+1} = \hat{O} f^{j}$

example: advection-diffusion equation
(with explicit Euler time stepping)

$$f^{j+1} = f^{j} + dt \cdot \vartheta \Delta f^{j} - dt \cdot c \nabla f^{j}$$

if operators are not unitary:

- **hand-design** of quantum circuits
- increasing number of quantum circuits for added complexity
- **no** convergence measure



tensor-programmable quantum operations:

- direct representation of operators
- capable to restrict to one circuit
- allows to define a convergence measure

[3] M. Lubasch et al., **Phys. Rev. A 101**, (2020)
[4] P. Over et al., **Comput. Fluids 299**, 106508, (2025)

Tensor Networks (TN) and “Tensor-Programming”

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG
DLR

compression method:

- compressed representation of fields (vectors)
- compressed representation of operators (matrices)
- applying operations in the compressed subspace

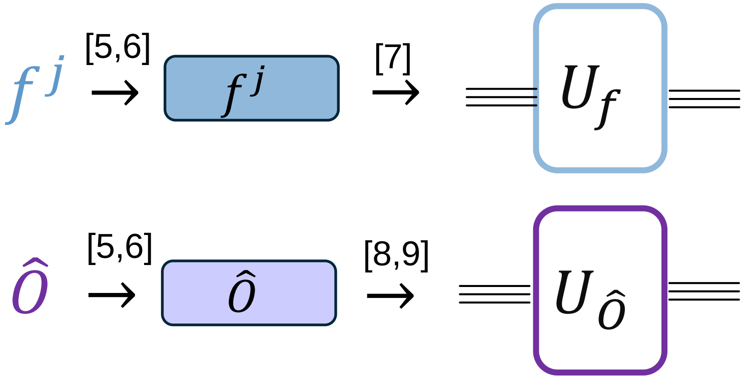
$\boxed{f^j}$

$\boxed{\hat{O}}$

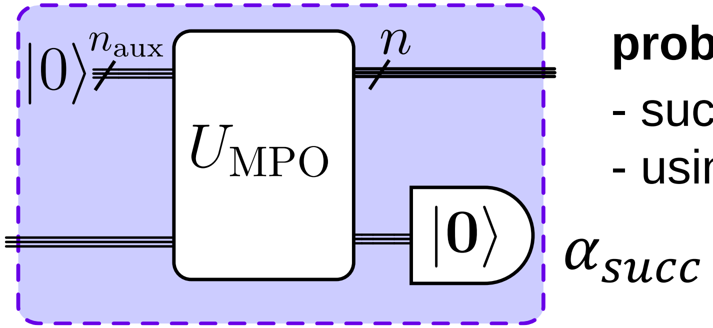
$\boxed{\hat{O}} : \boxed{f^j} \rightarrow \boxed{f^{j+1}}$

$\boxed{\hat{O}_1} + \boxed{\hat{O}_2} \rightarrow \boxed{\hat{O}_{new}}$

mappable on quantum computers:



$\rightarrow$



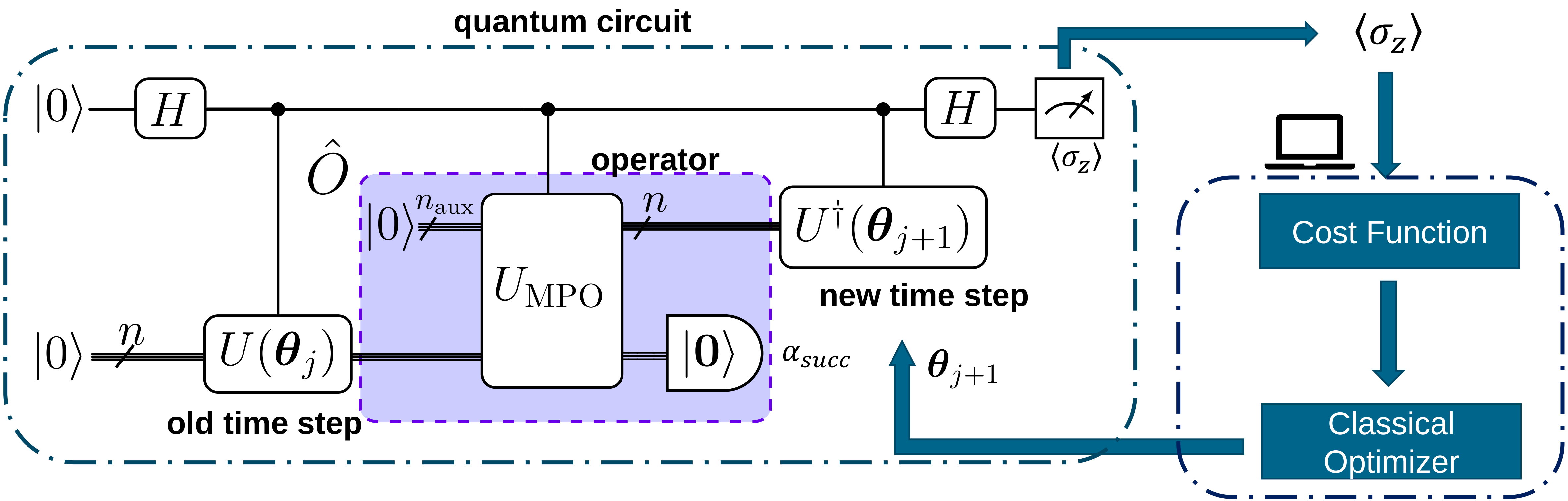
probabilistic application of $\hat{O}$:

- success probability α_{succ}
- using extra qubits + measurements

[5] Fernandez et al., **arxiv:2407.02454** (2024)
[6] Orus et al., **Ann. Phys. 349** (2014)
[7] Ran et al., **Phys. Rev. A 101**, 032310 (2020)
[8] Termanova et al., **New J. Phs. 26,** 123019, (2024
[9] Nibbi et al., Phys. Rev. A **110**, 042427, (2024)

Full Quantum Circuit

[10] combining tensor network based operator encoding with variational time stepping



[10] P. Siegl et al. (2025), https://arxiv.org/abs/2502.04425

Necessary Norm Correction

problem: solution has wrong norm field: $\begin{pmatrix} f_0 \\ \vdots \\ f_N \end{pmatrix}^j = \theta_0^j \begin{pmatrix} a_0 \\ \vdots \\ a_N \end{pmatrix}^j \rightarrow \theta_0^{j+1}$?

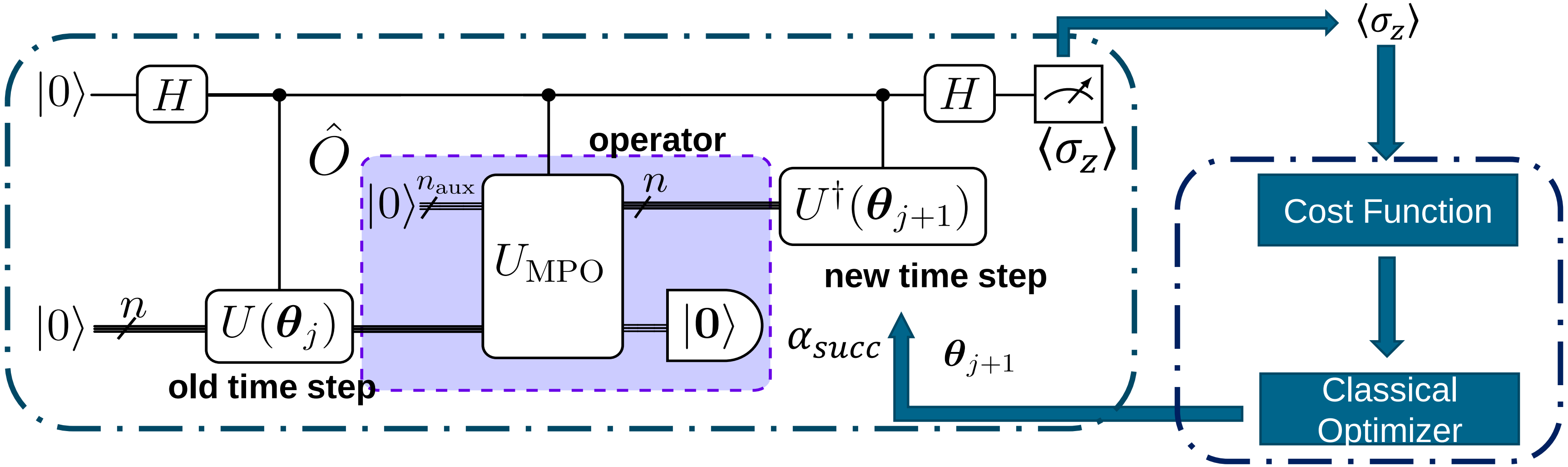
reason:

$\hat{O}_{classic} \begin{pmatrix} a_0 \\ \cdots \\ a_N \end{pmatrix} = \begin{pmatrix} \cdots \\ \cdots \\ \cdots \end{pmatrix}$, $\hat{O}_{quantum} \begin{pmatrix} a_0 \\ \cdots \\ a_N \end{pmatrix} = \begin{pmatrix} \cdots \\ \cdots \\ \cdots \end{pmatrix}$,

$$\boxed{f_{\hat{O}} \theta_0^j \rightarrow \theta_0^{j+1}}$$

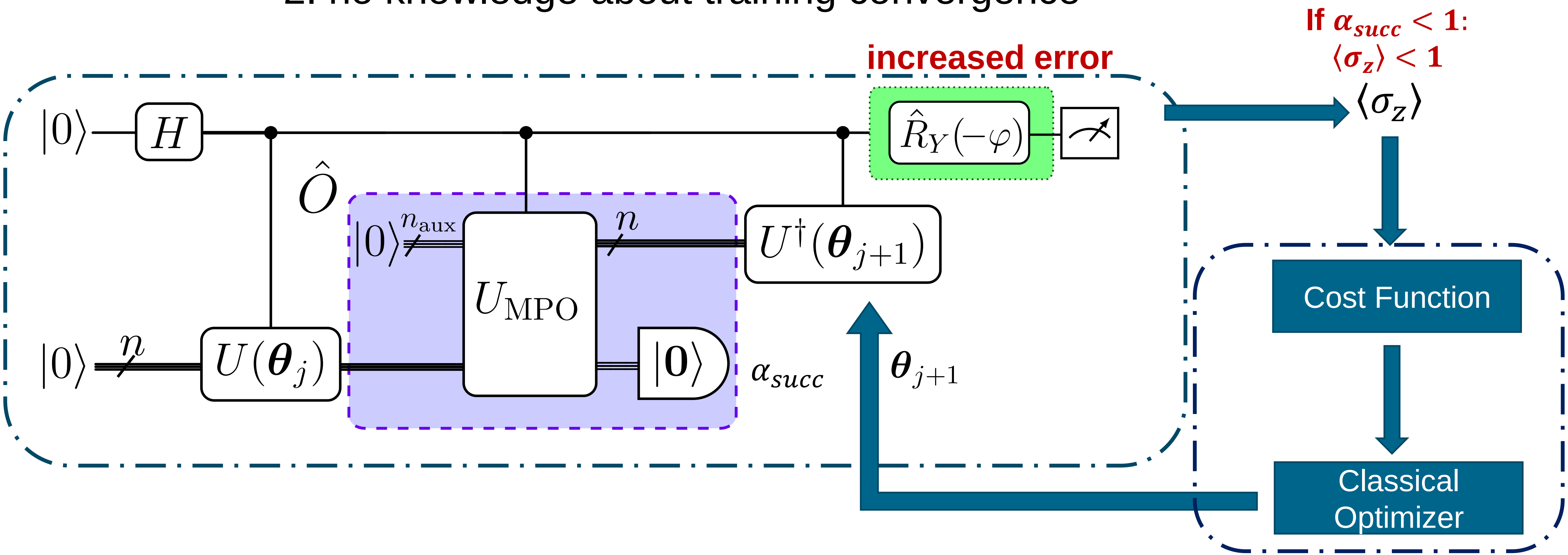
$\begin{pmatrix} \cdots \\ \cdots \\ \cdots \end{pmatrix} = f_{\hat{O}} \begin{pmatrix} \cdots \\ \cdots \\ \cdots \end{pmatrix}$,

can be computed **from α_{succ}**



Beneficial Correction and Convergence Measure

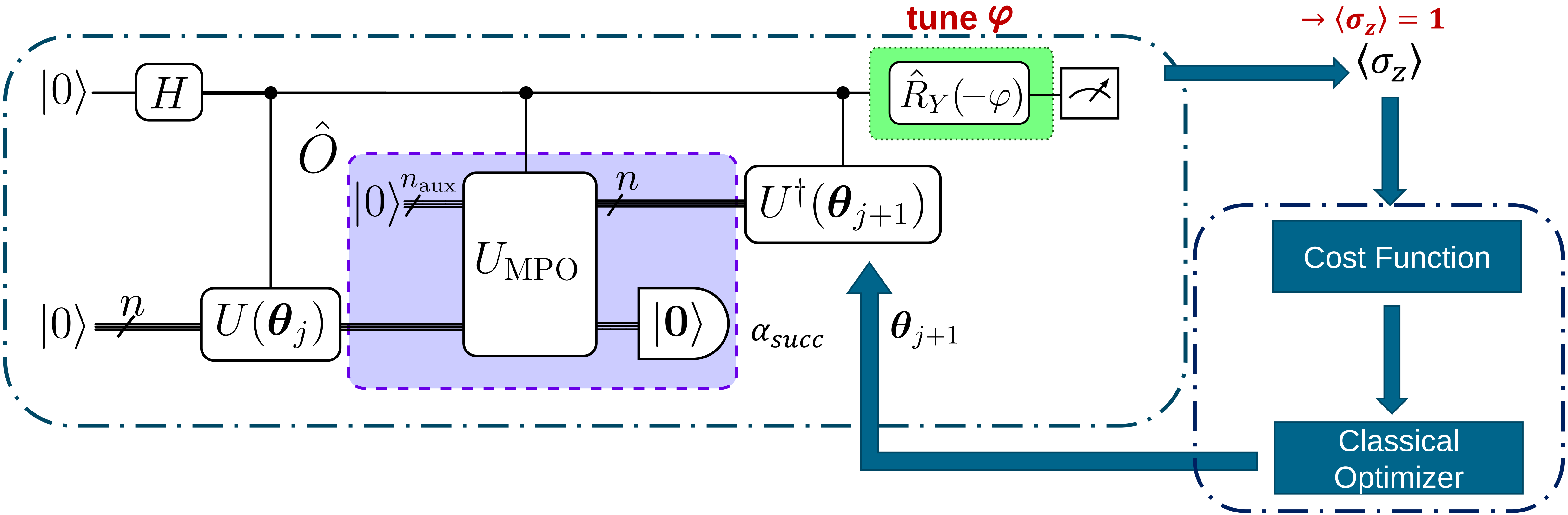
problems: 1. extra measurements $|0\rangle$ increase measurement error of
2. no knowledge about training convergence



[10] P. Siegl et al. (2025), https://arxiv.org/abs/2502.04425

Beneficial Correction and Convergence Measure

problems: 1. extra measurements $|0\rangle$ increase measurement error of
2. no knowledge about training convergence



[10] P. Siegl et al. (2025), https://arxiv.org/abs/2502.04425

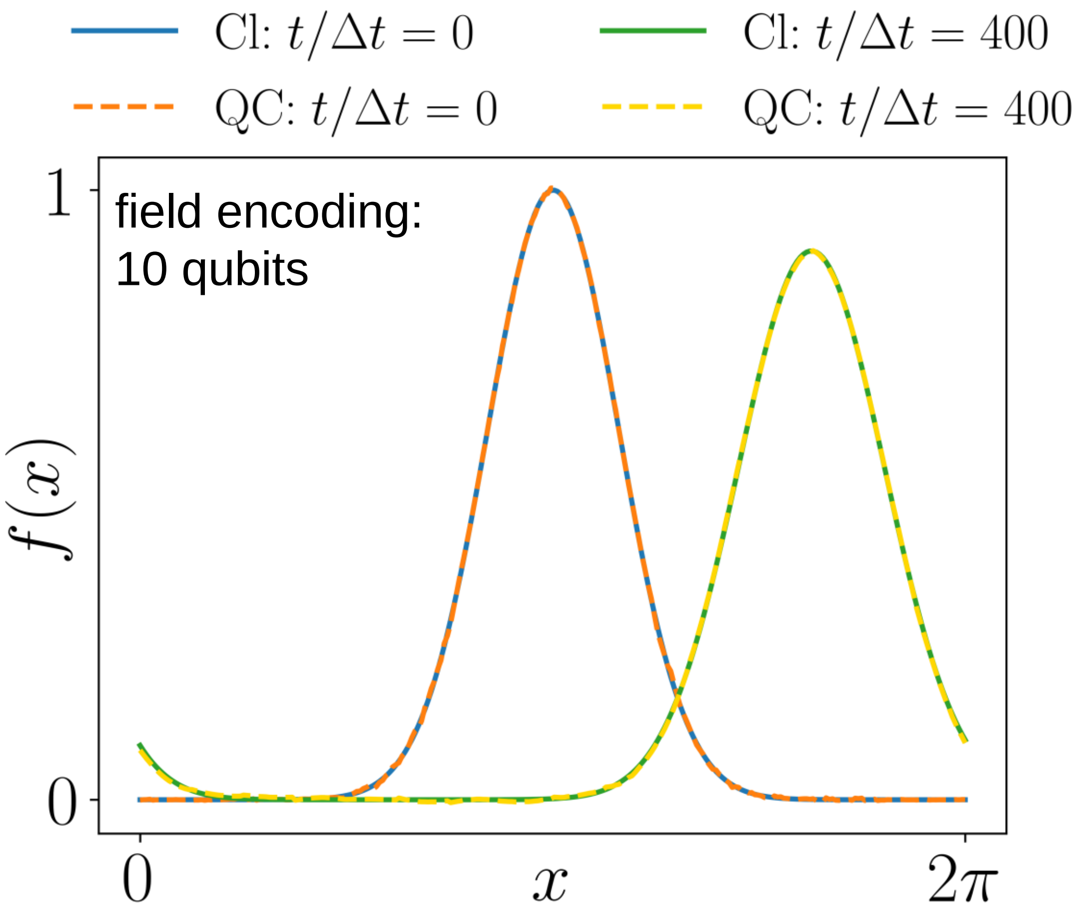
Example: Advection-Diffusion Equation

1D linear advection-diffusion equation

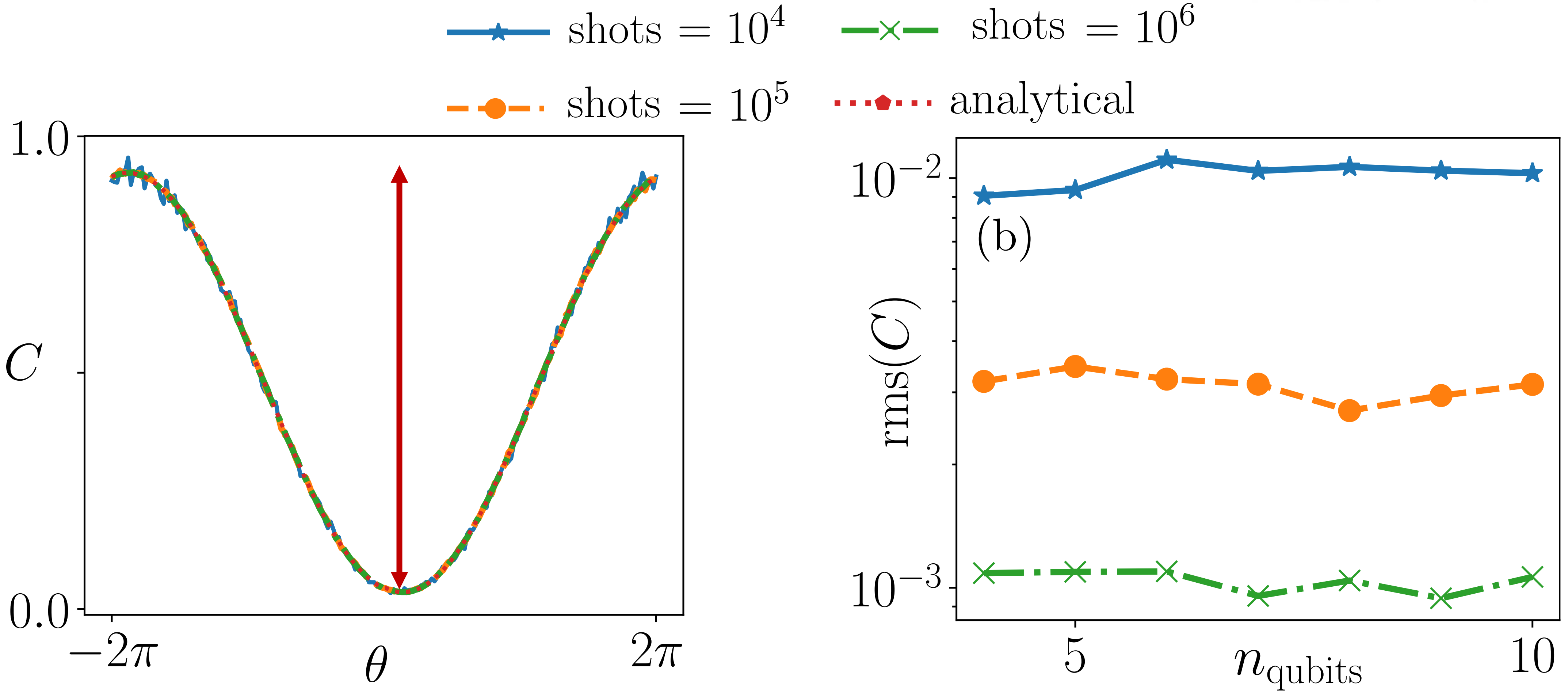
$$\frac{\partial f}{\partial t} = \vartheta \Delta f - c \nabla f$$

$$f(x, 0) = \exp\left(-\frac{(x-\pi)^2}{0.5}\right)$$

- explicit Euler time-stepping
- convergence measure: fidelity
- run on quantum simulator



Training Landscape



- “warm-starts” keep training landscape pronounced
- apparently no barren plateaus.

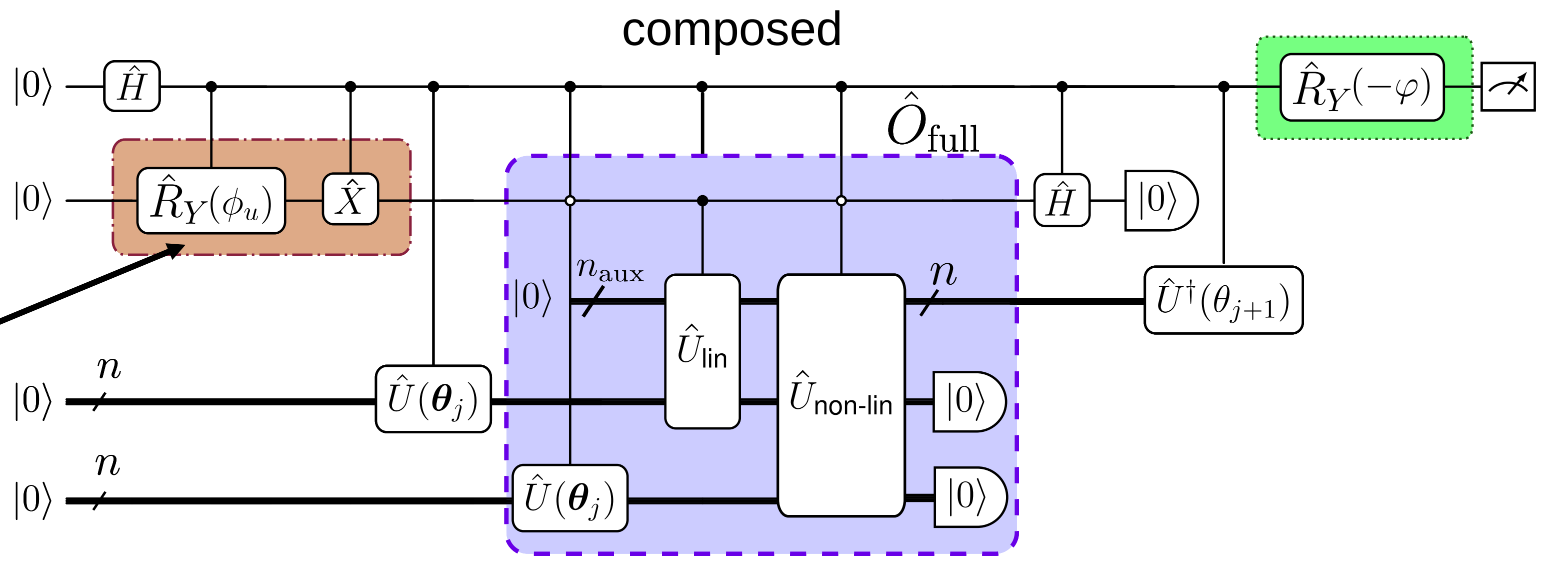
Extension to Non-Liner Problems

problem:

$$\frac{\partial u}{\partial t} = \vartheta \Delta u - u \nabla u \rightarrow$$ one operator depends on $u(t)$

solution:

weighted contribution of linear and non-linear term



depending on $\|u\|$

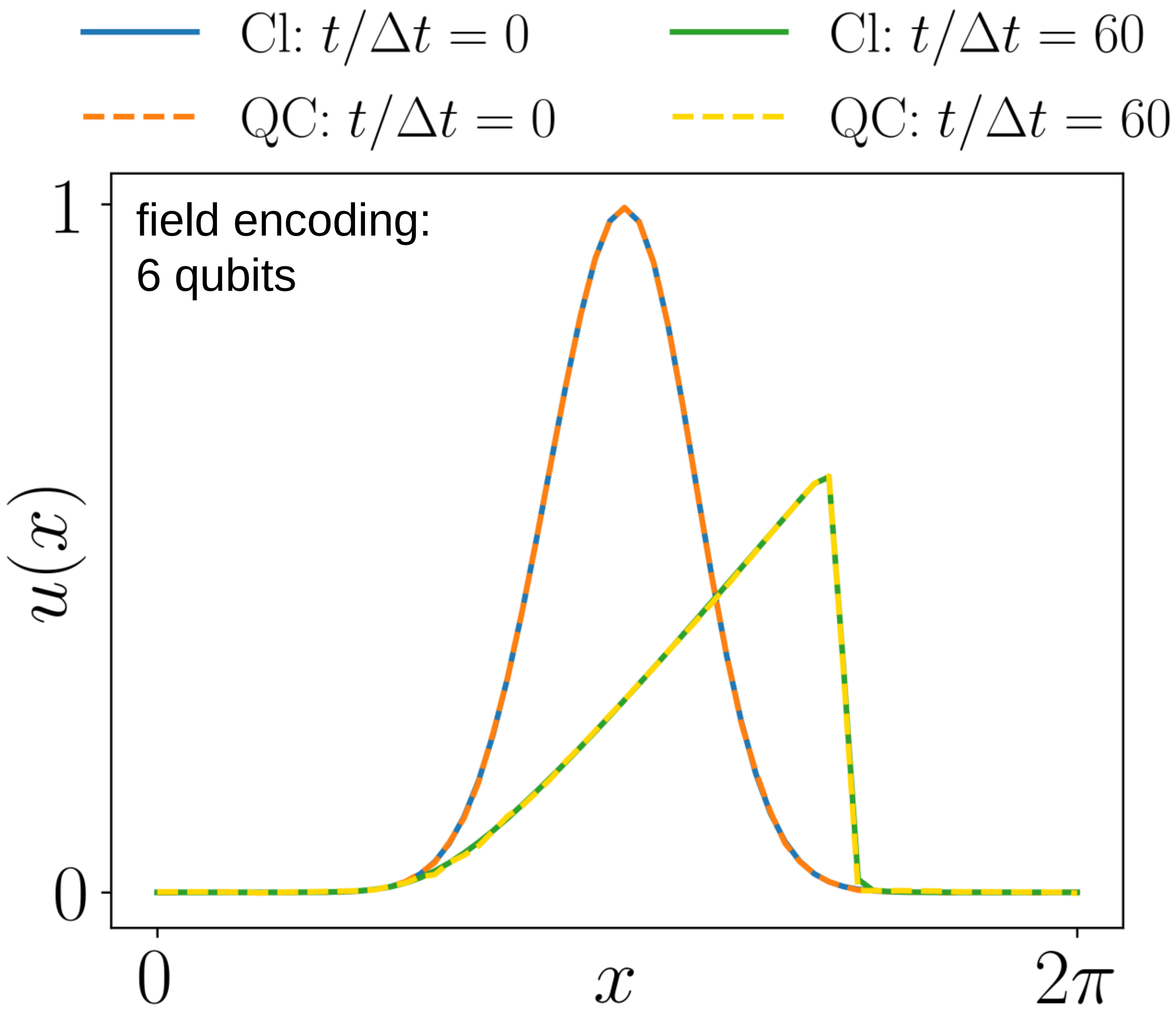
Example: Non-linear Burgers' Equation

1D non-linear Burgers' equation

$$\frac{\partial u}{\partial t} = \vartheta \Delta u - u \nabla u$$

$$f(x, 0) = \exp\left(-\frac{(x-\pi)^2}{0.5}\right)$$

- explicit Euler time-stepping
- convergence measure: fidelity
- run on quantum simulator



Summary and Outlook

variational time-stepping routine with tensor-programmable operators

Summary:

- very easy incorporation of various operators with little additional cost
- allows definition of convergence measure
- extendible to non-linear differential equations
- successful application to various use-cases

Outlook:

- scaling properties
- application to more complex use cases
- implementation of flexible training framework

contact: pia.siegl@dlr.de

arxiv: arxiv:2502.04425

Questions?

Literature

[1] Oerlemans, et al., J. **Sound Vib. 299**, 869, (2007)

[2] IBM-research_kryostat.jpg (higgs.ch)

[3] M. Lubasch et al., **Phys. Rev. A 101**, (2020)

[4] P. Over et al., **Comput. Fluids 299**, 106508, (2025)

[5] Fernandez et al., **arxiv:2407.02454** (2024)

[6] Orus et al., **Ann. Phys. 349** (2014)

[7] Ran et al., **Phys. Rev. A 101**, 032310 (2020)

[8] Termanova et al., **New J. Phs. 26,** 123019**,** (2024)

[9] Nibbi et al., Phys. Rev. A **110**, 042427**,** (2024)

[10] P. Siegl et al. (2025), https://arxiv.org/abs/2502.04425