
Estimation of gradients
and analysis of gradient-based optimizers
for variational quantum algorithms

INAUGURAL-DISSERTATION ZUR
ERLANGUNG DES AKADEMISCHEN GRADES

doctor rerum naturalium (Dr. rer. nat.)

IN THEORETISCHER PHYSIK

der
Mathematisch-Naturwissenschaftlichen Fakultät
der Universität zu Köln

vorgelegt von

DAVID WIERICHS

aus Aachen

Köln, 2022

Gutachter: Prof. Dr. David Gross

PD Dr. Rochus Klesse

Tag der Disputation: 11.01.2023

Abstract

On its path towards computational advantage, quantum computing hardware still is at an early prototypical stage, not yet allowing the use of error correction codes and algorithms that are provably more performant than their classical competitors. Even so, it might be the case that the current noisy quantum devices can be used for relevant computations that are out of reach for current classical computers, if only for a few specific applications, and without any performance guarantees. The question is thus whether the quantum devices in the near future can be promoted to more than a mere bridge technology, which gave rise to the field of research on noisy intermediate-scale quantum (NISQ) algorithms. A substantial part of these research efforts, and also the main topic of this thesis, concerns variational quantum algorithms (VQAs), which aim at using NISQ devices in a hybrid setup together with classical computers. In such a setup, a computational problem is encoded in terms of an observable, typically a Hamiltonian, such that determining the minimal energy (or the corresponding ground state) would yield the solution. With the purpose of generating candidates for the ground state, a classical computer selects a quantum circuit from a family of circuits and the quantum device executes the circuit to prepare the corresponding state. The family of circuits is typically defined in form of a parametrized quantum circuit (PQC) and a particular circuit can be chosen by fixing its parameters. In return the classical computer receives measurement outcomes of selected observables on the candidate states. The algorithm proceeds by optimizing over the parameter space in order to find for (a useful approximation of) the target state.

There are many variants and proposed use cases for VQAs which has led to a modular structure of the algorithms. In this thesis, the first chapter focuses on derivative estimators for objective functions that are based on PQCs, which is a subroutine commonly used in the optimization within VQAs. It begins with a review of (componentwise) differentiation methods for PQC-based objective functions, followed by a detailed comparison of these methods using the example of a ubiquitous class of PQCs. This comparison confirms and complements recent results on the topic and has implications for gradient estimation in practice.

The second chapter continues on the theme of gradient estimation for optimization, and covers so-called parameter-shift rules, a particular class of derivative estimators. It starts with a brief review of the literature, followed by an extension of said estimators to a specific class of gates for quantum chemistry calculations, which was published as part of a larger manuscript. The main part of the chapter is a publication about the generalization of the parameter-shift rule to a larger class of quantum gates. It presents derivative estimators for various gates, some of which have been shown in the literature to be optimal, together with a thorough cost analysis for both classical simulators and quantum hardware.

The third and final chapter gives a short outline of VQAs and provides the context for a second publication, which analyses different algorithms for the optimization task in VQAs.

It compares the quantum natural gradient optimizer (QNG) to two established gradient-based methods from classical non-convex optimization. This is done with numerical experiments, in which QNG shows favourable convergence properties and enhanced robustness against symmetry-breaking POCs for highly structured problems. This work uses the variational quantum eigensolver (VQE), a popular example of a VQA, on spin chain Hamiltonians as its benchmark problem and the experiments are based on extensive classical simulations.

A promising direction for future work is to investigate individual building blocks of the large VQA construct separately and to develop metrics for these blocks that allow to predict their usefulness in practice. This may reduce the complexity of single research efforts and lead to insights that are as modular as VQAs themselves.

Contents

Acronyms	vi
Symbols and Notation	vii
Introduction	ix
1 Parametrized Quantum Circuits	1
1.1 Qubits, circuits and objective functions	1
1.2 Differentiation of PQC-based functions	8
1.2.1 Costs of derivative estimators	9
1.2.2 Finite differences	12
1.2.3 Parameter-shift rules	16
1.2.4 Antisymmetric two-term recipes	20
1.2.5 Linear combination of unitaries	21
1.2.6 Simultaneous perturbation stochastic approximation	28
1.2.7 Differentiation on classical simulators	28
1.2.8 Estimator comparison for $R = 1$	32
2 Parameter-shift rules	45
2.1 Literature discussion	45
2.2 Four-term shift rule for quantum chemistry gates	48
2.2.1 Introduction to the four-term rule	48
2.2.2 Two-term rule and shift tuning	48
2.2.3 Four-term parameter-shift rule	49
2.2.4 Minimizing the variance	52
2.3 Contributions to the first publication	54
2.4 Publication: General parameter-shift rules for quantum gradients	56
3 Variational Quantum Algorithms	83
3.1 Ansätze, initializations and optimizers	83
3.2 Contributions to the second publication	86
3.3 Publication: Avoiding local minima in variational quantum eigensolvers with the natural gradient optimizer	86
Conclusion	105
Acknowledgements	107
Bibliography	109
Formalia	121

Acronyms

ADAPT-VQE	adaptive derivative-assembled pseudo-Trotter ansatz VQE
BFGS	Broyden-Fletcher-Goldfarb-Shanno
DFT	discrete Fourier transform
LCU	linear combination of unitaries
LHS	left hand side
MSE	mean squared error
NISQ	noisy intermediate-scale quantum
PQC	parametrized quantum circuit
QAD	quantum analytic descent
QAOA	quantum approximate optimization algorithm
QML	quantum machine learning
QNG	quantum natural gradient optimizer
QPU	quantum processing unit
RAM	random access memory
SPSA	simultaneous perturbation stochastic approximation
UCC	unitary coupled cluster
VQA	variational quantum algorithm
VQE	variational quantum eigensolver

Symbols and Notation

Symbol	Meaning
a_ℓ	Coefficients of even part of E_k
\tilde{a}_ℓ	Coefficients of even part of $\langle H^2 \rangle_k$
B	Hamiltonian transformed by parts of a PQC
b_ℓ	Coefficients of odd part of E_k
\tilde{b}_ℓ	Coefficients of odd part of $\langle H^2 \rangle_k$
C	(Parametrized) quantum circuit
$C_{[j:k]}$	Contiguous part of a PQC, incl. the j th and excl. the k th gate
c_j	Rescaling factor in a gate group
$c-U$	Unitary/Quantum gate controlled on at least one aux. qubit
E	Objective function stemming from an expectation value of a PQC
E_k	Univariate restriction of E to its k th parameter
$E_k^{(j)}$	j th derivative of E_k
e_k	Canonical basis vector
F	Perturbation operator in a parametrized quantum gate
f	Function from \mathbb{R}^n to \mathbb{R}^p
\mathcal{F}	Set of Fourier coefficients for E_k and $\langle H^2 \rangle_k$
G	Generator of a parametrized quantum gate
g_r	Summand in the derivative computed for a decomposition of G
H	Hamiltonian
H_ℓ	Term from a Pauli decomposition of H , corresponding to I_ℓ
H_p	p th harmonic number
\tilde{H}_t	Approximation of Hessian in the BFGS algorithm
h_i	Coefficient in Pauli decomposition of H
h	Shift parameter in a differentiation recipe
h^*	Optimal shift parameter in a differentiation recipe
I_ℓ	Index subset for a Pauli decomposition of H
\mathbb{I}	Identity matrix in arbitrary number of dimensions
K	Number of terms in a decomposition of H or G
L	Number of parameter positions sampled in a numerical experiment
M	Number of gates in a gate group
m	Number of qubits a generic gate acts on
m_t	Momentum term at step t in Adam optimizer
m_μ	Shift multiplier in a finite difference recipe
N	Number of qubits in a generic qubit register
n	Number of input parameters of a PQC
P	Tensor product of Pauli operators (and \mathbb{I}), short Pauli word
p	Halved number of shifts in a generalized central difference stencil
Q	Generator of a quantum number-preserving gate
\bar{Q}	Traceless generator of a quantum number-preserving gate
q	Number of shifts in a finite difference recipe
R	Number of frequencies in a Fourier series
R_\star	Pauli rotation gate (single qubit for $\star \in \{X, Y, Z\}$)
S_μ	Sum of reciprocals of $\{m_\nu\}_{\nu \neq \mu}$
s	Shot budget
t	Coefficient in perturbed quantum gate or optimization step index

Symbol	Meaning
U	Unitary/Quantum gate
\mathcal{U}	Quantum channel created by unitary U
V	Unitary for a basis change
$V(\mathbf{m})$	Vandermonde matrix of vector \mathbf{m}
V_G	Particular quantum gate built from a decomposition of G
\mathbf{v}_t	Momentum term at step t in Adam optimizer
X	Pauli- X operator, or domain for x
x	Real number (sometimes restricted to X)
x_μ	Shift in a differentiation recipe
Y	Pauli- Y operator
y_μ	Coefficient in a differentiation recipe
y_1^*	Optimal coefficient in a two-term recipe
y_1°	Optimal coefficient on average in a numerical experiment
Z	Pauli- Z operator
$\alpha_\mu / \alpha_{p,\mu}$	Coefficient in a finite (central) difference recipe (with $2p$ shifts)
$\boldsymbol{\alpha}_p$	Coefficient vector in the central difference recipe with $2p$ shifts
γ_r	Absolute value of $\tilde{\gamma}_r$
$\tilde{\gamma}_r$	Coefficient in a decomposition of G
Δ^2	Averaged MSE of an estimator in a numerical experiment
$\partial_{[\cdot]}^{[h]}$	Differentiation recipe, optionally with indicated shift h
ϵ	Regularization parameter in quantum natural gradient
$\epsilon^2[\cdot]$	MSE of an estimator
η (η^*)	(Optimal) Learning rate in a gradient-based optimizer
$\boldsymbol{\theta} / \boldsymbol{\theta}_t$	Parameters of a PQC / at optimization step t
Λ	Eigenvalue spectrum of an operator
λ_k	Eigenvalue of a Hermitian operator
λ^*	Optimal rescaling parameter for parameter-shift rule
ρ	Fraction between one-shot variance and derivative; density matrix
$\sigma^2[\cdot]$	One-shot variance of an estimator
$ \phi\rangle$	Quantum state prepared by part of a PQC
φ	Real-valued phase parameter
χ	Real-valued phase, i.e. $\chi \in \{\pm 1\}$; also $\chi_r = \text{sgn} \tilde{\gamma}_r$
$ \psi\rangle$	Quantum state prepared by a PQC
Ω	Base frequency in a Fourier series with commensurable frequencies
Ω_ℓ	ℓ th frequency in a Fourier series
ω_j	Generator eigenvalue
$ k\rangle$	Computational basis state
$\langle O \rangle$	Expectation value of operator O with respect to the state of a PQC
$\langle O \rangle_k$	Univariate restriction of $\langle O \rangle$ to its k th parameter
\otimes	Kronecker product/power of matrices or vectors
\odot	Elementwise product/power of vectors
\oplus	Direct sum of matrices
\times_k	Cartesian product of spaces labeled by k
$\widehat{\cdot}^k$	Statistical estimator for a quantity
$\mathbb{V}[\cdot]$	Variance of an estimator
$\mathbb{E}[\cdot]$	Expectation value of an estimator
\dots	Average over one-dimensional domain X
\dots_\circ	Quantities averaged over samples in numerics ($\sigma_\circ^2, E_\circ'^2, a_\circ^2, \rho_\circ$)

For differing notation in the publications see Sec. 2.3 and Sec. 3.2.

Introduction

In the research field on quantum computing, most effort originally used to be spent on concepts for fault-tolerant quantum information processing. This includes the development of fault-tolerant algorithms, error correction codes and complexity theoretic proofs to categorize the power of error-correcting quantum computers. The requirements imposed on quantum processing units (QPUs) by this paradigm are harsh, considering today's hardware [1, 2]. Three key metrics commonly used to illustrate this are: first, the number of quantum bits, or *qubits*, that are needed to realize the redundant information storage underlying error correction codes in order to compensate the flaws of each single physical qubit. Second, the largest allowed error rates in the information storage and in the operations that implement the computation before error correction methods lose the upper hand. Third, the connectivity of a QPU architecture, i.e. the number of qubits that are considered close to each other in the network of controllable qubit interactions. These and other requirements are not independent but can be balanced against each other; in any scenario, the realization of a fault-tolerant quantum computer poses a major scientific and technological challenge. This original approach to quantum computing is concerned with algorithms that can be proven mathematically to perform better than their classical (non-quantum, that is) counterparts for sufficiently large problems. The price for this guarantee on increased computational power is the above canon of requirements on the quantum computing hardware. In addition, coprocessing algorithms need to be run on powerful classical computers to realize error correction sufficiently quickly for large QPUs.

By now, the field has seen first experimental realizations of QPUs with a few qubits, based on different physical principles to store and manipulate the quantum information. These principles include the charge, current, spin, polarization or energy of the used physical system, as well as time binning of photon measurements or the photonic occupation number [3]. The corresponding architectures used in the realizations vary widely, too. They include superconducting circuits, trapped ions, quantum dots, spin qubits in silicon, nitrogen vacancies in diamonds and linear optics networks. Following up on these first demonstrators, QPUs with a few more, and few dozens of, qubits are being built and used in research already [4, 5, 6, 7]. This marks the advent of a different paradigm in quantum information processing, which is commonly dubbed the noisy intermediate-scale quantum (NISQ) [8] regime and has received a lot of attention in recent years. NISQ algorithms

attempt to solve computationally hard problems while being realizable on existing or near-future quantum hardware. This strongly limits the resources these algorithms may assume, like the number and connectivity of the qubits, and in particular they cannot make use of full error correction. In addition, noise affects all stages of information processing in a NISQ device, which restricts the number of operations that can be executed within a single computation, demands the algorithms to be robust against said noise, and requires mitigation protocols to reduce the distortion of the results without consuming too many additional resources. Despite these shortcomings, NISQ devices are of interest in various regards. They can of course simply be seen as a necessary step towards large error-corrected QPUs or as tools for basic research on the devices and the exploited physical principles themselves. In this thesis, however, I am concerned with using them for computations that may be realizable while lying out of reach for classical algorithms [9, 10].

As opposed to fault-tolerant quantum algorithms, proving computational advantage is typically neither achievable nor the primary goal for NISQ algorithms. Two reasons that often make such proofs hard in practice are that the computations contain subroutines for which no performance guarantees are known, like non-convex optimization, and that it is difficult to predict and model the impact of noise in the QPU in a general setting. Furthermore, proving computational advantage is usually not the primary goal because NISQ devices cannot be scaled up easily but it is essential to work with whatever devices exist. This means that the constants and sub-leading factors in the computational cost – should an expression thereof be available – might be as relevant as the impact of the leading order in the problem size when comparing NISQ to classical algorithms. This is in contrast to the perspective of complexity theory, which attempts to show that there is *some* problem size at which a certain algorithm performs favourably by making statements about asymptotic behaviour. Furthermore, proofs of computational complexity often are concerned with worst-case or average-case performance across some class of problems, which might not necessarily capture the behaviour for applications. This gap between provable guarantees and behaviour in practice is not a specific feature of quantum algorithms but can also be found in classical computer science.

A substantial part of research on NISQ algorithms is concerned with variational quantum algorithms (VQAs) [11, 12] that attempt to use QPUs in tandem with classical coprocessors to prepare interesting quantum states and extract practically relevant properties of the states afterwards. This includes NISQ algorithms for combinatorial optimization, some variants of quantum machine learning (QML) as well as variational quantum eigensolvers (VQEs), whose purpose is to find certain eigenstates of a – typically quantum – physical system. The “variational” in VQAs stems from the variational method in quantum mechanics in the context of finding energy eigenstates of a quantum system, i.e. from the envisioned application of VQEs, but (by now) also refers more generally to the use of so-called parametrized quantum circuits (PQCs). These circuit families allow for searching a subspace of the full state space, or Hilbert space, that are (expected to be) accessible

at reasonable cost on a NISQ device. The hope then lies in being able to construct such parametrizations that contain the sought-after quantum states – or states that resemble them in some physical quantity and/or significantly overlap with them – and to find the parameters that correspond to those target states. For the latter step typically some objective or cost function depending on the PQC is considered, which can be evaluated by running the circuit on a NISQ device. Using these evaluations, optionally together with some auxiliary quantities from the QPU, an optimization algorithm is then run on a classical computer to minimize the objective. An important class of these algorithms are gradient-based optimization algorithms.

VQAs and the use of gradient-based subroutines in the optimization step make up the context of this thesis. As the title suggests, it treats two topics in particular: estimating the gradient of PQC-based objective functions for the use in optimizers for VQAs and analysing a selection of these gradient-based optimization strategies. The structure of the thesis is as follows:

Chap. 1 introduces PQCs together with relevant basic concepts in Sec. 1.1 and discusses most of the established estimators in the literature for gradients of PQC-based functions in Sec. 1.2. This discussion includes “hardware-ready” methods that can be implemented on QPUs directly as well as algorithms that are tailored to classical simulators of QPUs and cannot be readily run on NISQ devices. It concludes with a comparison of the hardware-ready methods for a particular, simple yet ubiquitous class of PQCs that shows that the budget of circuit repetitions, or shots, strongly influences which gradient estimator should be used in practice. A particular result is that the somewhat naïve finite difference method can outperform the so-called parameter-shift rule, a special purpose estimator, by far – provided it is used with appropriate parameters. This chapter moreover discusses known gradient estimators and results on their properties from the literature, but also new perspectives and variants thereof together with unpublished insights such as the comparison mentioned above, which includes a newly conducted numerical experiment.

Chap. 2 treats the parameter-shift rule in detail, beginning with a literature review in Sec. 2.1. Afterwards Sec. 2.2 presents a generalization of the shift rule, which is part of Ref. [13], to a specific type of operations in the quantum circuit. The core part of the chapter is a full reprint of a publication on further generalizations of the shift rule [14], preceded by a short commentary in Sec. 2.3.

Chap. 3 briefly introduces VQAs in Sec. 3.1 to provide the context for the reprint of a second publication [15], again preceded by a short commentary in Sec. 3.2. As said publication itself introduces the particular PQC architectures, or ansätze, problem Hamiltonians and optimization techniques, there is no separate review, but the reader is referred to corresponding reviews in the literature.

The thesis concludes with a brief summary of the results together with some comments on how to potentially extend the presented work.

Chapter 1

Parametrized Quantum Circuits

In this chapter I introduce parametrized quantum circuits (PQCs) and the objective functions they give rise to. These circuits and objective functions are key ingredients of variational quantum algorithms (VQAs), the main subject of Chap. 3. Here, the focus lies on the differentiation of PQC-based objective functions, on both quantum hardware and classical simulators.

This chapter is structured as follows: Sec. 1.1 contains basic definitions of quantum circuits and the type of parametrized gates we will be concerned with in this work. That is, I will not aim at the most general definitions but only include relevant circuit classes for the presented publications and additional results.

In Sec. 1.2 I discuss the differentiation of functions based on PQCs and complement the publication in Chap. 2 on parameter-shift rules with analyses of other differentiation techniques. This section contains unpublished results that I hope provide a useful overview of differentiation techniques, as well as an introduction to their practical evaluation.

1.1 Qubits, circuits and objective functions

The quantum systems we will discuss are *qubit registers*, i.e. compositions of two-level quantum systems, and we denote the number of qubits in a register as N . The appropriate mathematical description for the state space of such a system is the $2^N - 1$ -dimensional complex projective space. However, it will be sufficient for this thesis to consider the complex vector space \mathbb{C}^{2^N} , i.e. assume states to be normalized (in 2-norm), and “manually” identify states that only differ by a global phase, i.e. a factor $\exp(i\varphi)$ with $\varphi \in \mathbb{R}$.

A *quantum gate* U , or simply a gate, acting on m qubits is a unitary matrix in $\mathbb{C}^{2^m \times 2^m}$, i.e.

$$U \in \mathbb{C}^{2^m \times 2^m} \text{ and } U^\dagger U = \mathbb{I}. \quad (1.1)$$

Such an m -qubit gate acts on a register with $N \geq m$ qubits by forming the Kronecker product, or tensor product, with the identity matrix on the remaining $N - m$ qubits and

performing standard matrix-vector multiplication. This identification is assumed throughout the thesis, including the publications, and we will not denote it explicitly. While the word gate is sometimes used to refer only to the most elementary operations, or matrices, we will not make this restriction here.

A *parametrized gate* U is a function that maps a real-valued parameter x to a gate $U(x)$. For this thesis, we will restrict parametrized gates to the form $U(x) = \exp(i(xG + F))$ with (not necessarily commuting) Hermitian matrices G and F . For most of the gates, F will vanish, leaving us with one-parameter groups $U(x) = \exp(ixG)$. G is called the *generator* of the gate, and F can be understood as a *perturbation*. This definition is more restrictive than it has to be, as gates with multiple arguments or different functional forms could be considered.

Common examples for parametrized gates stem from applications and implement useful operations for specific tasks, or they are of interest because they are native to (some) quantum computing architectures. Examples of the former include the building blocks of the quantum approximate optimization algorithm (QAOA) [16] and gates for applications in quantum chemistry that preserve symmetries, or quantum numbers [13, 17, 18]. Commonly used hardware-friendly parametrized gates are the single-qubit Pauli rotations R_X, R_Y, R_Z and – depending on the hardware – CNOT gates [19], two-qubit rotations like $R_{ZZ}(x) = \exp(-ixZ \otimes Z/2)$ [20] and $R_{XX}(x)$ [21], or parametrized (fermionic) SWAP gates [22].

A *quantum circuit* C is a sequence of quantum gates, each assigned to act on a specified set of qubits in a register. As usual, the qubits a circuit acts on are simply all qubits on which at least one of the gates acts. Moreover, circuits (and parts of circuits) are identified with unitary matrices via matrix multiplication of the constituents (in reverse order, and again by building the tensor product of each gate with the identity if $m < N$). As one might notice, this makes telling circuits from gates conceptually difficult, which is by design: we can easily switch between gates and their *decompositions*, i.e. circuits with typically “more elementary” gates that have the same matrix as the gate. We write $C_{[j:k]}$ for the circuit consisting of the j th (inclusive) to k th (exclusive) gate and if j is the smallest possible or k is the biggest possible index, we skip it¹.

Unless stated otherwise, we will consider a qubit register to be in the state $|0\rangle^{\otimes N}$ before any circuit is applied. We will abbreviate this, and in fact $|0\rangle^{\otimes m}$ for any m , as $|0\rangle$ when the number of qubits is clear from the context. Introducing a certain ambiguity, we will say that a circuit prepares a quantum state, which refers to applying it to the initial state $|0\rangle$.

A *parametrized quantum circuit* is a map C from n real-valued parameters θ to a quantum circuit $C(\theta)$. We do not assume anything about the structure of this map, and in particular allow that a given parameter is fed into multiple gates at any locations in the circuit. Note that in contrast to gates and circuits, a PQC cannot be interpreted as a parametrized gate in general because we restricted those to be of the form $\exp(i(xG + F))$. For some

¹ This is the slicing notation used in Python.

PQCs with a single input ($n = 1$), however, such an equivalence exists and the circuit is a decomposition of the corresponding parametrized gate in this case. We sometimes denote the state prepared by a PQC as $|\psi(\boldsymbol{\theta})\rangle := C(\boldsymbol{\theta})|0\rangle$.

From circuits to objective functions Ultimately, we will be interested in a real-valued function that originates from estimating, or measuring, the (physical) expectation value $\langle H \rangle$ of some observable H in the quantum state prepared by a PQC. This measurement results in a map

$$E : \mathbb{R}^n \rightarrow \mathbb{R} \tag{1.2}$$

$$\boldsymbol{\theta} \mapsto E(\boldsymbol{\theta}) := \langle 0 | C(\boldsymbol{\theta})^\dagger H C(\boldsymbol{\theta}) | 0 \rangle, \tag{1.3}$$

which is called the *objective function* or energy. I first consider the functional form of such objective functions and then briefly discuss the statistical experiments realized on quantum processing units (QPUs) to estimate $E(\boldsymbol{\theta})$.

It can be shown that E is a (perturbed) Fourier series in its n input parameters $\boldsymbol{\theta}$ where the frequency spectrum is determined by the gate generators and locations, and the perturbations (with bounded Fourier spectrum) are caused by the perturbation terms of the gates. A derivation for unperturbed gates is given in Sec. 2.1 of the publication in Chap. 2, also consider the educational example below. The coefficients of the series depend on the circuit structure and on H . This observation has been made repeatedly in the literature and was exploited in various ways: first, it allows to understand PQCs on the footing of established Fourier theory. Second, it enables the design of PQCs [23, 24], differentiation techniques [25, 26, 27] (also see Sec. 1.2 and Chap. 2) and optimization strategies with favourable properties for VQAs [25, 28, 29, 30, 31, 32]. Third, it inspires tactics for noise mitigation in experiments [33], and yields an exact classical model of E that can be constructed (approximately) from function evaluations on a QPU [28, 34, 35]. Differentiation of functions based on PQCs that contain perturbed gates has been studied in [36, 37]. In the discussion here and in Sec. 1.2 we will consider the unperturbed case unless stated otherwise.

We will frequently consider a univariate restriction of E to a single parameter. For ease of notation, we will abbreviate

$$E_k(x) := E(\boldsymbol{\theta} + x\mathbf{e}_k), \tag{1.4}$$

where \mathbf{e}_k is the k th canonical basis vector for \mathbb{R}^n and we omitted $\boldsymbol{\theta}$ entirely on the left hand side, as it usually is clear from the context². We extend this notation to the expectation

² If it is not, we stick to the explicit notation.

value of other observables O and write

$$\langle O \rangle_k(x) := \langle 0 | C(\boldsymbol{\theta} + x\mathbf{e}_k)^\dagger O C(\boldsymbol{\theta} + x\mathbf{e}_k) | 0 \rangle. \quad (1.5)$$

The univariate restriction (in the above sense) of a multi-dimensional Fourier series is a one-dimensional finite Fourier series, for which we use the following notation (in the unperturbed case):

$$E_k(x) = a_0 + \sum_{\ell=1}^R a_\ell \cos(\Omega_\ell x) + b_\ell \sin(\Omega_\ell x). \quad (1.6)$$

The frequencies Ω_ℓ depend on the (combination of) used gates in the PQC and its structure, but not on the measured observable.

In some works, and indeed in the publication in Chap. 2, all gates that depend on a given parameter θ_k are assumed to be executed successively, but it is important to note that the univariate restriction in Eq. (1.4) is a Fourier series even if these gates are distributed across the circuit. Such a setup is relevant in particular for quantum machine learning (QML), e.g. when using data reuploading [34, 38]. If multiple gates are controlled by a parameter, the frequency spectrum $\{\Omega_\ell\}$ of the restriction contains the sums and differences of the frequencies contributed by the separate gates. As the spectrum for each separate gate contains the frequency 0, these sums and differences include the gate frequencies themselves.

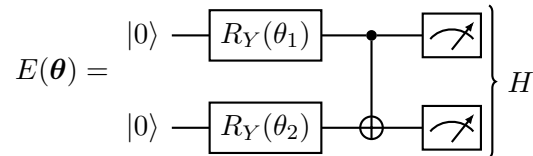
In order to illustrate the concepts introduced above, I give a toy example of a PQC and an objective function arising from it. The PQC we consider is

$$C(\boldsymbol{\theta}) = \text{CNOT}^{(1,2)} R_Y^{(2)}(\theta_2) R_Y^{(1)}(\theta_1) \quad (1.7)$$

where the superscript denotes the qubits the gates act on, $R_Y(x) = \exp(ix(-Y/2))$ is the Pauli- Y rotation gate, and $\text{CNOT} = |0\rangle\langle 0| \otimes \mathbb{I} + |1\rangle\langle 1| \otimes X$. The circuit acts on $N = 2$ qubits and takes an $n = 2$ -dimensional argument $\boldsymbol{\theta}$. The problem Hamiltonian is

$$H = \frac{3}{4}Z^{(2)} + \frac{1}{4}X^{(1)} = \frac{3}{4}\mathbb{I} \otimes Z + \frac{1}{4}X \otimes \mathbb{I}. \quad (1.8)$$

A standard way to denote this setup is via a quantum circuit diagram:



Complementary to the derivation in Chap. 2 mentioned above, we here compute the objective function by directly evaluating the transformation of the quantum state caused

by the circuit:

$$|\psi(\boldsymbol{\theta})\rangle = C(\boldsymbol{\theta}) \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \cos(\theta_1/2) \cos(\theta_2/2) \\ \sin(\theta_1/2) \cos(\theta_2/2) \\ \sin(\theta_1/2) \sin(\theta_2/2) \\ \cos(\theta_1/2) \sin(\theta_2/2) \end{pmatrix}. \quad (1.9)$$

When contracting this state with the Hamiltonian above, we obtain the objective function

$$E(\boldsymbol{\theta}) = \frac{3}{4} \cos(\theta_1) \cos(\theta_2) + \frac{1}{4} \sin(\theta_1) \sin(\theta_2). \quad (1.10)$$

As we can see, the two Pauli- Y rotations yield the frequency 1 and there is no constant term for this particular choice of C and H . The univariate restriction defined in Eq. (1.4) e.g. for $\boldsymbol{\theta} = (\pi/4, \pi/3)^T$ and $k = 2$ reads

$$E_2(x) = E(\boldsymbol{\theta} + x\mathbf{e}_2) = \frac{3 + \sqrt{3}}{8\sqrt{2}} \cos(x) + \frac{1 - 3\sqrt{3}}{8\sqrt{2}} \sin(x). \quad (1.11)$$

To demonstrate how more complicated frequency spectra arise, we also consider a reparametrized version of the above circuit, namely with a single new parameter $\tilde{\theta}$ and the mapping $\boldsymbol{\theta}(\tilde{\theta}) = (\tilde{\theta}, \frac{\tilde{\theta}}{2})^T$. Using trigonometric identities, this gives rise to the objective function

$$\tilde{E}(\tilde{\theta}) = \frac{1}{2} \cos\left(\frac{\tilde{\theta}}{2}\right) + \frac{1}{4} \cos\left(\frac{3\tilde{\theta}}{2}\right). \quad (1.12)$$

On the level of the PQC, the reparametrization means that we replace the two Pauli- Y rotations by the two-qubit gate $U_{YY}(\tilde{\theta}) = \exp\left(i\tilde{\theta}(-Y \otimes \mathbb{I}/2 - \mathbb{I} \otimes Y/4)\right)$ generated by a sum of two rescaled Pauli words. The analysis in Chap. 2 shows that this gate in general could contribute an additional term with the frequency 1, which vanishes in the example above.

Objective function estimation Here I briefly discuss for completeness how to measure the objective function E , sticking to a basic approach. The analysis of derivative estimators in the rest of the chapter treats this measurement process as a “black box” that can be queried for a specified number of measurements. This approach makes sense because the functional dependence of the measurement statistics on the variational parameters, which is the only property we will use in our derivations, is not affected by choosing a measurement recipe. Additional care needs to be taken when using derivative estimators that pose special conditions on the measured observable and thus force us to decompose the Hamiltonian further.

A common method to represent H is to decompose it into a sum of *Pauli words*, i.e. ten-

sor products of the Pauli operators (and identity) $\{X, Y, Z, \mathbb{I}\}$, denoted as

$$H = \sum_{i=1}^K h_i P_i. \quad (1.13)$$

In practice, H is then measured by grouping (pairwise) commuting Pauli words together, according to sets $I_\ell \subset \{1, \dots, K\}$. This allows to measure the terms $\langle H_\ell \rangle := \sum_{i \in I_\ell} h_i \langle P_i \rangle$ of the Hamiltonian independently. For this, the prepared quantum state is rotated into the common eigenbasis of the words $\{P_i\}_{i \in I_\ell}$, the register is measured in the computational basis, and the eigenvalues of H_ℓ corresponding to the obtained basis states are determined (classically). The rotation only requires up to N additional single-qubit gates that can be effected simultaneously, which we assume to be a negligible overhead.

For each set I_ℓ , the circuit with the appended rotations is configured on the device and executed s_ℓ times, obtaining an estimator for $\langle H_\ell \rangle$ by averaging the computed eigenvalues. This means that the number of Pauli word sets, or of bases to measure in, impacts the cost of estimating $\langle H \rangle$ significantly. As an example, consider the single-qubit Hamiltonian $H = h_1 P_1 + h_2 P_2$ with $[P_1, P_2] = 0$, which we could measure simultaneously or separately. Assuming a total budget of s shots, measuring simultaneously yields the variance³

$$\mathbb{V} \left[\hat{E}_{\text{sim}} \right] = \frac{1}{s} \left[h_1^2 (1 - \langle P_1 \rangle^2) + h_2^2 (1 - \langle P_2 \rangle^2) + 2h_1 h_2 (\langle P_1 P_2 \rangle - \langle P_1 \rangle \langle P_2 \rangle) \right], \quad (1.14)$$

whereas separate measurements (with equally allocated shots) result in

$$\mathbb{V} \left[\hat{E}_{\text{sep}} \right] = \frac{1}{s} \left[2h_1^2 (1 - \langle P_1 \rangle^2) + 2h_2^2 (1 - \langle P_2 \rangle^2) \right] \quad (1.15)$$

$$= \mathbb{V} \left[\hat{E}_{\text{sim}} \right] + \mathbb{V} \left[\hat{D}_{\text{sim}} \right] \quad \text{with } D = \langle h_1 P_1 - h_2 P_2 \rangle \quad (1.16)$$

$$\geq \mathbb{V} \left[\hat{E}_{\text{sim}} \right], \quad (1.17)$$

where the last term is non-negative, because it is a variance. This makes the former measurement scheme favourable in terms of precision and tells us to measure two Pauli words simultaneously if we can.

However, for more terms in H the situation quickly becomes complicated. First, there is not one unique grouping into mutually commuting words and finding a grouping into fewest possible sets of words is a computationally hard problem [39]. Second, the optimal measurement grouping depends on the weights h_i , which are known, as well as the (co)variances of the Pauli words with respect to the quantum state, which are not known a priori. Even worse, it turns out that minimizing the number of sets does *not* necessarily minimize the variance of the overall energy measurement [40, 41, 42].

It should also be noted that the number of Pauli words – or other operators that can be measured easily – in H differs strongly between the envisioned applications of PQCs. As

³ See p. 7 for details on the used notation.

extreme examples, quantum embedding kernels use an overlap measurement between pure states as cost function [43], which corresponds to measuring the single projector $|0\rangle\langle 0|$, whereas Hamiltonians arising from electronic structure calculations e.g. for quantum chemistry generically contain $\mathcal{O}(N^4)$ Pauli words for N orbitals [40].

The described strategy to measure H by measuring sets of commuting Pauli words is straightforward and does not inflict any additional cost beyond the rotation gates into the eigenbases of the sets. Other measurement strategies are known and can be shown to have favourable statistical properties [44], drastically reduce the number of separately measured Hamiltonian terms [42] and/or allow for error mitigation protocols [44, 45, 46, 47]. However, these methods come at the cost of additional auxiliary qubits and/or multi-qubit gates, which in turn require enhanced qubit connectivity and longer coherence times. We will not discuss the challenge of grouping the Pauli words of a Hamiltonian for the energy measurement further, but point out the references [39, 42, 48, 49, 50, 51, 52, 53, 54, 55] that suggest and review methods to approach this challenge on different levels of the computational pipeline, in particular for electronic structure calculations. In this work we will assume the simple measurement based on a grouping of the Pauli words.

When measuring the mutually commuting Pauli words in a set $\{P_i\}_{i \in I_\ell}$, the layer of local rotation gates V into the eigenbasis is executed after the PQC and a computational basis state $|k\rangle$ is sampled from the qubit register. The eigenvalue $\lambda_{\ell,k} := \sum_{i \in I_\ell} h_i \langle k| V P_i V^\dagger |k\rangle$ can then be computed classically⁴ and with s_ℓ repetitions of this sampling procedure we obtain the estimator⁵

$$\hat{E}_\ell(\boldsymbol{\theta}) = \frac{1}{s} \sum_{j=1}^s \hat{\lambda}_\ell^{(j)}(\boldsymbol{\theta}), \quad \hat{\lambda}_\ell^{(j)}(\boldsymbol{\theta}) = \lambda_{\ell,k} \text{ with probability } |\langle k| V |\psi(\boldsymbol{\theta})\rangle|^2. \quad (1.18)$$

This means that all single-measurement random variables follow the indicated probability induced by the PQC and are independent. We construct the estimator $\hat{E}(\boldsymbol{\theta})$ for the full energy by summing over the sets of Pauli words, i.e. the different measurement bases. This estimator is unbiased because each $\hat{\lambda}_\ell^{(j)}$ is:

$$\mathbb{E} \left[\hat{\lambda}_\ell^{(j)}(\boldsymbol{\theta}) \right] = \sum_{k=1}^{2^N} \langle \psi(\boldsymbol{\theta}) | V^\dagger |k\rangle \lambda_{\ell,k} \langle k| V |\psi(\boldsymbol{\theta})\rangle \quad (1.19)$$

$$= \sum_{i \in I_\ell} h_i \langle \psi(\boldsymbol{\theta}) | P_i | \psi(\boldsymbol{\theta}) \rangle \quad (1.20)$$

$$= \langle H_\ell \rangle(\boldsymbol{\theta}). \quad (1.21)$$

⁴ Note that V is not unique because there is no canonical eigenstate ordering, and that $\lambda_{\ell,k}$ therefore depends on the choice of V together with k .

⁵ This notation collides with the one for univariate restrictions of $E(\boldsymbol{\theta})$. We will not use it outside of this section.

The variance of the ℓ th term is

$$\mathbb{V} \left[\hat{E}_\ell(\boldsymbol{\theta}) \right] = \frac{1}{s} \left[\langle \psi(\boldsymbol{\theta}) | H_\ell^2 | \psi(\boldsymbol{\theta}) \rangle - E_\ell(\boldsymbol{\theta})^2 \right], \quad (1.22)$$

and their sum is the variance for $\hat{E}(\boldsymbol{\theta})$. Occasionally it will be useful to write

$$\sigma^2 \left[\hat{E}_\ell(\boldsymbol{\theta}) \right] := s \mathbb{V} \left[\hat{E}_\ell(\boldsymbol{\theta}) \right] = \left[\langle \psi(\boldsymbol{\theta}) | H_\ell^2 | \psi(\boldsymbol{\theta}) \rangle - E_\ell(\boldsymbol{\theta})^2 \right], \quad (1.23)$$

separating the variance of the operator from the number of shots (also see Sec. 1.2.1. Note that this variance of $\hat{E}(\boldsymbol{\theta})$ differs from the physical variance $\langle H^2 \rangle - \langle H \rangle^2$ of the full Hamiltonian by covariance terms $\langle H_\ell H_{\ell'} \rangle - \langle H_\ell \rangle \langle H_{\ell'} \rangle$ with $\ell \neq \ell'$. As the measurement procedure essentially samples eigenstates of a specific observable H_ℓ and the corresponding eigenvalues are accumulated classically, an estimate of the variance can be computed during the sampling process, allowing for dynamic methods that adapt the number s of collected samples.

Due to the first term $\langle H_\ell^2 \rangle$ in Eq. (1.22) being an expectation value, it again is a Fourier series with the same frequencies as $\langle H_\ell \rangle$, whereas the second term will contain pairwise sums and differences of the original frequencies and thus extends the frequency spectrum of $\mathbb{V} \left[\hat{E}(\boldsymbol{\theta}) \right]$ compared to $\mathbb{E} \left[\hat{E}(\boldsymbol{\theta}) \right]$.

While the above description in terms of random variables is the correct approach for computations run on quantum computers, we will also frequently come back to exact expectation values, which are attained in the limit $s \rightarrow \infty$. This is the correct setting to derive differentiation rules for E , but also represents the situation encountered in most classical simulators for quantum circuits.

1.2 Differentiation of PQC-based functions

Many optimization routines for VQAs are gradient-based. Besides evaluating the objective function E itself, these optimizers require an estimate of its gradient ∇E . There are multiple estimators for the entries of ∇E , which differ significantly in their statistical properties and the extent to which they exploit the structure of E . For convenience I first list some existing comparisons between these methods in the literature. Afterwards I move on to introduce and discuss the gradient estimators in detail, including methods that are only available on classical simulators of quantum computers. These “software-only” methods differ fundamentally from the quantum “hardware-ready” techniques and are particularly interesting for research based on numerical experiments.

Literature comparing differentiation methods Here I discuss a few other works that investigated different derivative estimators and compared them. Ref. [56] performs a detailed comparison of the forward and central difference (see Sec. 1.2.2) with the two-

term parameter-shift rule (Sec. 1.2.3). The authors discuss the bias-variance tradeoff implemented in the finite difference methods and introduce a rescaled parameter-shift rule, which is the unique derivative estimator based on two shifted evaluations that minimizes the mean squared error (MSE). I will present a rather similar discussion in Sec. 1.2.8 but with a different perspective that is tailored to the trigonometric nature of the objective functions and treats the central difference and the parameter-shift rule as particular choices of a larger family of estimators. Furthermore I include additional derivative estimators that do not use two shifted evaluations, namely higher-order finite differences and the linear combination of unitaries (LCU)-based estimator from Sec. 1.2.5. These turn out to not be favourable for the investigated example circuit, and the optimal estimator I find is the same as the one in [56]. Nonetheless I think that the new perspective presented here might be useful and in particular leads to methods that provide the optimal estimator in practice, given sufficiently “representative” data on the PQC and objective function. If no data is available, my results contradict [56] for small shot budgets.

Shortly before I finished this thesis, Ref. [57] appeared online, introducing a Bayesian learning framework that optimizes the derivative estimator with respect to the MSE during the training loop which typically is used in VQAs. This is complemented with an analysis of the average Fourier coefficients in a PQC that allows to obtain gradient estimators which perform well on average. One of the conclusions is that finite differences are favourable in the regime of few shots per measurement, which also is the result of Sec. 1.2.8.

In [27] the entire family of parameter-shift rules for (unperturbed) PQC-based functions is analysed within a measure theoretic mathematical framework, leading to existence and optimality proofs, as well as computational methods to obtain new shift rules.

1.2.1 Costs of derivative estimators

For most parts of this section, we will consider the gradient entries

$$E'_k(0) = \frac{\partial}{\partial x_k} E(\boldsymbol{\theta} + x_k \mathbf{e}_k) \Big|_{x_k=0} \quad (1.24)$$

separately and therefore consider the univariate restrictions E_k of the objective function. To increase readability, we drop the index k and denote derivative recipes as $\partial_{[\cdot]}$.

Mean squared error In order to evaluate estimators for $E'(0)$, we will discuss their total deviation from the exact value in terms of the *mean squared error (MSE)*

$$\varepsilon^2[\partial_{[\cdot]}\hat{E}(0)] := \mathbb{E} \left[\left(\partial_{[\cdot]}\hat{E}(0) - E'(0) \right)^2 \right] = \mathbb{V} \left[\partial_{[\cdot]}\hat{E}(0) \right] + \left(\mathbb{E} \left[\partial_{[\cdot]}\hat{E}(0) \right] - E'(0) \right)^2, \quad (1.25)$$

which – as usual – is composed of the variance and the squared systematic error, or squared bias. Some estimators will be unbiased, so that

$$\varepsilon^2[\partial_{[\cdot]}\hat{E}(0)] = \mathbb{V}[\partial_{[\cdot]}\hat{E}(0)] = \sigma^2[\partial_{[\cdot]}\hat{E}(0)]/s. \quad (1.26)$$

We focus on the variance of certain schemes of estimators and trigonometric objective functions like those produced by PQCs, and briefly look at their bias afterwards.

Some of the derivative estimators will make use of estimates for E at positions shifted away from 0, so that the variance at these positions influences the analysis:

$$\partial_{[\cdot]}E(0) = \sum_{\mu} y_{\mu}E(x_{\mu}) \Rightarrow \mathbb{V}[\partial_{[\cdot]}\hat{E}(0)] = \sum_{\mu} \frac{y_{\mu}^2}{s_{\mu}} \sigma^2[\hat{E}(x_{\mu})], \quad (1.27)$$

where s_{μ} is the number of shots used to evaluate $\hat{E}(x_{\mu})$. As mentioned before, the variance is a Fourier series similar to E itself but typically has additional frequencies, because it is not only composed of $\langle H^2 \rangle$, which takes the form Eq. (1.6), but also includes E^2 . At x_{μ} it is given by

$$\sigma^2[\hat{E}(x_{\mu})] = \tilde{a}_0 + \sum_{\ell=1}^R \left(\tilde{a}_{\ell} \cos(\Omega_{\ell}x_{\mu}) + \tilde{b}_{\ell} \sin(\Omega_{\ell}x_{\mu}) \right) \quad (1.28)$$

$$- \left[a_0 + \sum_{\ell=1}^R a_{\ell} \cos(\Omega_{\ell}x_{\mu}) + b_{\ell} \sin(\Omega_{\ell}x_{\mu}) \right]^2, \quad (1.29)$$

where we denoted the Fourier coefficients of $\langle H^2 \rangle$ with $\tilde{\cdot}$. We will call a recipe of the form in Eq. (1.27) *antisymmetric* if it uses pairs of shifts $x_{\pm\mu} = \pm x_{\mu}$ together with coefficient pairs $y_{\pm\mu} = \pm y_{\mu}$. For such a recipe it will be useful to know that

$$\sum_{x \in \{\pm 1\}} \sigma^2[\hat{E}(x x_{\mu})] = 2\tilde{a}_0 + 2 \sum_{\ell=1}^R \tilde{a}_{\ell} \cos(\Omega_{\ell}x_{\mu}) \quad (1.30)$$

$$- 2 \left[a_0 + \sum_{\ell=1}^R a_{\ell} \cos(\Omega_{\ell}x_{\mu}) \right]^2 - 2 \left[\sum_{\ell=1}^R b_{\ell} \sin(\Omega_{\ell}x_{\mu}) \right]^2. \quad (1.31)$$

Often we are not interested in the error at a single point, but in its average across all x , because we want to evaluate the cost of the estimators more generally than for a particular parameter position. Therefore we will occasionally replace the local variance $\mathbb{V}[\hat{E}(x_{\mu})]$ by its average σ^2/s_{μ} across the domain of θ_k , where s_{μ} is the number of samples used to estimate $\hat{E}(x_{\mu})$. Effectively this means that we evaluate the derivative recipe averaging over θ_k while keeping all other θ_{ℓ} , $\ell \neq k$, fixed. Note that this approach assumes the MSE of $\partial_{[\cdot]}\hat{E}(0)$ to depend linearly on the individual variances at the shifted positions, which is not true in general. In fact, for finite differences the standard tradeoff between bias

and variance will make the dependency non-linear. We will anyways make use of the replacement $\mathbb{V}[\hat{E}(x_\mu)] \mapsto \sigma^2/s_\mu$, considering it an approximation to the true behaviour (see Sec. 1.2.2 for details). Moreover, this approximation also arises whenever we allocate shots independently of the Fourier coefficients, as explained below. This allocation often makes sense because we set out to measure those coefficients and therefore will not know them beforehand.

Regarding the bias, we observe that $E'(0) = \boldsymbol{\Omega} \cdot \mathbf{b}$ (c.f. Eq. (1.6)) and that an antisymmetric derivative recipe yields

$$\partial_{[\cdot]}E(0) = \sum_{\mu>0} 2y_\mu \sum_{\ell=1}^R b_\ell \sin(\Omega_\ell x_\mu), \text{ so that} \quad (1.32)$$

$$\partial_{[\cdot]}E(0) - E'(0) = \sum_{\ell=1}^R b_\ell \left(\sum_{\mu>0} 2y_\mu \sin(\Omega_\ell x_\mu) - \Omega_\ell \right). \quad (1.33)$$

This means that the magnitude of the Fourier coefficients b_ℓ , as compared to those of the variance, will be a decisive quantity when evaluating bias-variance tradeoffs.

Shot allocation For derivative estimators composed of multiple estimators $\hat{E}(x_\mu)$, the total budget of s shots needs to be allocated appropriately. The optimal allocation follows the absolute value of the weighted standard deviations:

$$s_\mu = s \frac{|y_\mu| \sigma[\hat{E}(x_\mu)]}{\sum_\nu |y_\nu| \sigma[\hat{E}(x_\nu)]}, \quad (1.34)$$

which can be shown e.g. via Lagrange multipliers. The resulting variance is

$$\mathbb{V}[\partial_{[\cdot]}\hat{E}(0)] = \frac{1}{s} \left(\sum_\mu |y_\mu| \sigma[\hat{E}(x_\mu)] \right)^2. \quad (1.35)$$

A crucial disadvantage of the optimal shot allocation is its dependence on the unknown variances, and therefore on the sought-after Fourier coefficients, at the shifted positions, which prevents us from computing the allocation beforehand. Three approaches to alleviate this problem are commonly used: first, one can decide to only account for the influence of the known weights y_μ and choose $s_\mu = s|y_\mu|/\|\mathbf{y}\|_1$. This allocation, which we call *practical allocation*, also minimizes the variance *on average* across all (univariate) parameter positions if we use a fixed derivative recipe, and we will frequently make use of it as it is a feasible allocation strategy. The resulting total variance is

$$\mathbb{V}[\partial_{[\cdot]}\hat{E}(0)] = \frac{\|\mathbf{y}\|_1}{s} \sum_\mu |y_\mu| \sigma^2[\hat{E}(x_\mu)]. \quad (1.36)$$

Second, the variances could be computed using other, approximate methods, e.g. on a classical computer [42]. This makes sense whenever we want to measure the energy derivative to a very high precision and the auxiliary methods deliver coarse variance estimates at low computational cost. A third approach is to first measure the objective function values at the shifted positions with a fraction of the total shot budget, estimate their variances based on these preliminary samples, and then use the estimates for the optimal shot allocation in Eq. (1.34) [58, 59, 60].

1.2.2 Finite differences

The first derivative estimators we consider are the forward and central differences with shift parameter h :

$$\partial_{\text{forw}}E(0) := \frac{E(h) - E(0)}{h}, \quad (1.37)$$

$$\partial_{\text{cent}}E(0) := \frac{E(h) - E(-h)}{2h}. \quad (1.38)$$

This technique likely is the most straightforward and simple approach; it is easily applied as numerical differentiation method for arbitrary (well-behaved) functions and does not make any assumptions about the structure of the function E . However, it therefore also cannot exploit additional information we have about E . The recipes require one and two evaluations of E per parameter, respectively, and the forward difference additionally reuses the value $E(\theta)$ for all entries of the gradient. Importantly, these finite differences are not exact even when we evaluate E without uncertainty; their systematic error and hence the bias of the corresponding estimators is typically characterized to leading order in h , assuming that $h \ll 1$:

$$\partial_{\text{forw}}E(0) - E'(0) = \frac{h}{2}E''(0) + \mathcal{O}(h^2), \quad (1.39)$$

$$\partial_{\text{cent}}E(0) - E'(0) = \frac{h^2}{6}E^{(3)}(0) + \mathcal{O}(h^3). \quad (1.40)$$

Consequently, using an infinitesimally small shift h would be optimal if we were able to evaluate E to infinite precision. However, in practice we always use a representation with finite precision, leading to a tradeoff between rounding errors and the above systematic error, which implies an optimal shift $h^* > 0$. On classical computers the numerical precision is very high, which leads to $h^* \ll 1$ and hence justifies the assumption made for the expansion of the bias to leading order in h . On the contrary, the precision with which we measure E on a quantum computer is much lower, increasing the variance contribution to the MSE and the optimal shift h^* , so that it might not satisfy $h^* \ll 1$ any longer. As also discussed in [56], we are therefore interested in the regime $h \not\ll 1$ as well. Lifting the assumption $h \ll 1$ additionally enables us to discuss the central difference in the same

framework as the two-term parameter-shift rule (see Sec. 1.2.3). However, it also means that the leading order term of the bias will not approximate it well, so that we instead have to look at its full expression. For the forward and central difference we compute them to be

$$\partial_{\text{forw}} E(0) - E'(0) = \frac{1}{h} \sum_{\ell=1}^R a_{\ell} (\cos(\Omega_{\ell} h) - 1) + b_{\ell} (\sin(\Omega_{\ell} h) - h\Omega_{\ell}) \quad (1.41)$$

$$\partial_{\text{cent}} E(0) - E'(0) = \frac{1}{h} \sum_{\ell=1}^R b_{\ell} (\sin(\Omega_{\ell} h) - \Omega_{\ell} h), \quad (1.42)$$

and observe that the leading-order description in general⁶ will *overestimate* them (in magnitude).

Higher-order stencils The generalization of the forward and central difference above leads to so-called finite difference stencils that take the form

$$\partial_{\text{FD}} \hat{E}(0) = \sum_{\mu=1}^q \frac{\alpha_{\mu}}{h} E(m_{\mu} h), \quad (1.43)$$

with constant coefficients $\alpha_{\mu} \in \mathbb{R}$ and shift multipliers $m_{\mu} \in \mathbb{R}$. This differentiation rule is of the form in Eq. (1.27) with $y_{\mu} = \alpha_{\mu}/h$ and $x_{\mu} = m_{\mu} h$, which yields the variance

$$\mathbb{V} \left[\partial_{\text{FD}} \hat{E}(0) \right] = \sum_{\mu=1}^q \frac{\alpha_{\mu}^2}{s_{\mu} h^2} \sigma_{\mu}^2 = \begin{cases} \frac{1}{sh^2} \left(\sum_{\mu=1}^q |\alpha_{\mu}| \sigma_{\mu} \right)^2 & \text{optimal allocation} \\ \frac{\|\alpha\|_1}{sh^2} \sum_{\mu=1}^q |\alpha_{\mu}| \sigma_{\mu}^2 & \text{practical allocation,} \end{cases} \quad (1.44)$$

where we again abbreviated $\sigma_{\mu} = \sigma \left[\hat{E}(m_{\mu} h) \right]$.

What is the structure of the stencil coefficients α ? Assume we are given q unique shifts m . Then we want the first-order term of the Taylor expansion of E to be reproduced exactly, i.e. $m \cdot \alpha = 1$, and we want the zeroth order to vanish, like as many higher-order terms as possible. This can be written as $\sum_{\mu} \alpha_{\mu} = 0$ and $m^{\odot j} \cdot \alpha = 0$ for as many consecutive $j \geq 2$ as possible, with $m^{\odot j}$ denoting the j th elementwise power. As we have q coefficients, we may hope that this works up to $j = q - 1$, so that α satisfies q linear equations. A short expression for these conditions is

$$V(m)\alpha = e_2, \quad (V(m))_{j\mu} = m_{\mu}^j, \quad 0 \leq j \leq q - 1, \quad (1.45)$$

where e again denotes a canonical basis vector and V is the (square) Vandermonde matrix of m ⁷. Indeed $V(m)$ is invertible (because the shifts are unique) so that a solution for

⁶ In the sense of "in many realistic scenarios".

⁷ Some definitions of the Vandermonde matrix start with the power $j = 1$, this one starts with $j = 0$. Yet other definitions consider the transpose of this definition.

α satisfying all q conditions exists. The column $V(\mathbf{m})^{-1}\mathbf{e}_2$ we are interested in is given by [61]

$$\alpha_\mu = (V^{-1}\mathbf{e}_2)_\mu = \left(\prod_{\substack{\tau=1 \\ \tau \neq \mu}}^q \frac{1}{m_\tau - m_\mu} \right) \sum_{\substack{\nu=1 \\ \nu \neq \mu}}^q \prod_{\substack{\tau=1 \\ \mu \neq \tau \neq \nu}}^q m_\tau \quad (1.46)$$

That means we do not need a general matrix inversion algorithm but the coefficients are known explicitly⁸. The leading order term of the bias for this stencil is $\mathcal{O}(h^{q-1})$ instead of $\mathcal{O}(h^q)$, because of the prefactor $\frac{1}{h}$ in Eq. (1.43), but we will not restrict ourselves to the domain of small shifts, as discussed for the forward and central difference above.

There is a small bonus the stencil has to offer, still. Consider the sum of reciprocals of all m_ν , but one, i.e.

$$S_\mu := \sum_{\substack{\nu=1 \\ \nu \neq \mu}}^q \frac{1}{m_\nu}. \quad (1.47)$$

If one of the shifts in \mathbf{m} ⁹ vanishes, say w.l.o.g. m_1 , only S_1 is well-defined. In this case, we have

$$\alpha_\mu = \begin{cases} S_1 & \text{for } \mu = 1 \\ -\frac{1}{m_\mu} \prod_{\substack{\tau=2 \\ \mu \neq \tau}}^q \frac{m_\tau}{m_\tau - m_\mu} & \text{for } \mu > 1 \end{cases} \quad (1.48)$$

and therefore α_μ is finite for all $\mu > 1$. If S_1 vanishes, so does α_1 and we do not actually use the function evaluation at $m_1 = 0$ for the derivative. If it is not contained in the set of shifts already, we may include the shift $m_{q+1} = -S_1^{-1}$ as additional term in the recipe, which explicitly sets $S_1 + m_{q+1}^{-1} = 0$ and therefore does not increase the number of used shifts while raising the order of the stencil by 1.

If none of the shifts are zero, we may rewrite

$$\alpha_\mu = S_\mu \prod_{\substack{\tau=1 \\ \tau \neq \mu}}^q \frac{m_\tau}{m_\tau - m_\mu} \quad (1.49)$$

and see that α_μ vanishes exactly if S_μ does. If one particular S_μ , say S_1 , vanishes, we know that no other S_ν can, because that would imply $\frac{1}{m_1} = \frac{1}{m_\nu}$, i.e. that two shifts were equal in the first place. If none of the S_μ vanish, we again may add another shift m_{q+1} in an attempt to increase the precision of the stencil without increasing the number of used shifts. This will succeed either if $\sum_\nu m_\nu^{-1} = 0$ (any m_{q+1} will have $\alpha_{q+1} = 0$) or if one of the potential new shifts $\{-S_\mu^{-1}\}_{\mu=1}^q$ is not contained in \mathbf{m} already.

⁸ Albeit in a somewhat cumbersome expression.

⁹ And exactly one, by uniqueness of the shifts.

In summary, we know that one coefficient in α vanishes for suitably chosen shifts, effectively reducing the number of shifts q by one. Conversely, again for suitable m , we can “silently” add a shift m_{q+1} without increasing the number of used shifts, because it will have the coefficient $\alpha_{q+1} = 0$ or set the coefficient for one of the other shifts to zero. The resulting stencil will have leading order bias $\mathcal{O}(h^q)$ (instead of $\mathcal{O}(h^{q-1})$).

A particular choice of shifts is $m_\mu = \mu$ for $-p \leq \mu \leq p$, where we changed the summation range to $\{-p, -p+1, \dots, p\}$, with the coefficients

$$\alpha_0 = 0, \quad \alpha_\mu = \frac{(-1)^{\mu+1}(p!)^2}{(p+\mu)!(p-\mu)!\mu}, \quad \forall -p \leq \mu \leq p, \mu \neq 0. \quad (1.50)$$

The shifts satisfy $\alpha_0 = S_0 = 0$ and therefore we effectively use only $2p$ shifts. This generalized central difference stencil¹⁰ is an antisymmetric derivative recipe as introduced in Sec. 1.2.1 and therefore incurs the bias

$$\partial_{\text{FD},p} E(0) - E'(0) = \frac{1}{h} \sum_{\ell=1}^R b_\ell \sum_{\mu=1}^p 2\alpha_\mu (\sin(\mu h \Omega_\ell) - \mu h \Omega_\ell), \quad (1.51)$$

for a Fourier series with R frequencies, where we used $\alpha \cdot m = 1$. To quantify the variance, as discussed in Sec. 1.2.1, we will need the norm

$$\|\alpha\|_1 = \sum_{\mu=-p}^p |\alpha_\mu| \quad (1.52)$$

$$= 2 \sum_{\mu=1}^p \frac{(p!)^2}{(p-\mu)!(p+\mu)!\mu} \quad (1.53)$$

$$= \frac{2}{\binom{2p}{p}} \sum_{\mu=1}^p \binom{2p}{p+\mu} \frac{1}{\mu} \quad (1.54)$$

$$= H_p, \quad (1.55)$$

with the p th harmonic number H_p . We compute at the end of this section. We can interpret $\partial_{\text{FD},p}$ as linear combination¹¹ of copies of ∂_{cent} with different shifts μh , using coefficients that sum to 1:

$$\partial_{\text{FD},p} = \sum_{\mu=1}^p 2\mu\alpha_\mu \partial_{\text{cent}}^{[\mu h]}. \quad (1.56)$$

From this perspective it is clear that the bias of the stencil is just the same linear combination of the biases of $\partial_{\text{cent}}^{[\mu h]}$ and that the variances are summed together with prefactors $\frac{4s}{s_\mu} (\mu\alpha_\mu)^2$ where s_μ is the shot budget allocated to the μ th central difference in the combi-

¹⁰ For $p = 1$ we get back ∂_{cent} with $\alpha_{\pm 1} = \pm 1/2$.

¹¹ The linear combination of gradient recipes is defined canonically: It is applied to a function by applying each of its rules and summing the results weighted with the coefficients of the combination.

nation.

If the variance is the dominating source of error, it will be crucial to minimize it by allocating all shots at the optimal shift size h^* rather than splitting them among (the optimal and) suboptimal shift sizes $\{h, 2h, \dots, ph\}$ in order to reduce the bias. This means that higher-order stencils hardly offer an advantage in this domain, and the optimal shot allocation would mostly sample the shifts $\pm h^*$, effectively ignoring the other terms of a stencil. However, the optimal allocation typically is not available and the practical shot allocation does not lead to this concentration on evaluations at h^* . Consequently, higher-order stencils will allocate too many shots to the unfavourable shifts $\{2h^*, 3h^* \dots\}$ and perform poorly compared to the simpler central difference at the optimal shift. This effect will also become apparent in the numerical experiments in Sec. 1.2.8.

To finalize this section we compute the norm $\|\alpha\|_1$ as announced above, using induction in p . For this we explicitly write $\alpha = \alpha_p$ and note $\|\alpha_1\|_1 = 1 = H_1$, so that the statement in Eqs. (1.52-1.55) holds for $p = 1$. Then we compute

$$\|\alpha_{p+1}\|_1 - \|\alpha_p\|_1 = \frac{2}{\binom{2p+2}{p+1}} \left[\frac{1}{p+1} + \sum_{\mu=1}^p \frac{1}{\mu} \left(\binom{2p+2}{p+1+\mu} - \binom{2p}{p+\mu} \frac{\binom{2p+2}{p+1}}{\binom{2p}{p}} \right) \right] \quad (1.57)$$

$$= \frac{2}{\binom{2p+2}{p+1}} \left[\frac{1}{p+1} + \sum_{\mu=1}^p \binom{2p+2}{p+1+\mu} \frac{1}{\mu} \left(1 - \frac{(p+1+\mu)(p+1-\mu)}{(p+1)^2} \right) \right]$$

$$= \frac{2}{\binom{2p+2}{p+1} (p+1)^2} \sum_{\mu=1}^{p+1} \binom{2p+2}{p+1+\mu} \mu \quad (1.58)$$

$$= \frac{1}{p+1}, \quad (1.59)$$

which together with the induction hypothesis implies that

$$\|\alpha_{p+1}\|_1 = \|\alpha_p\|_1 + \frac{1}{p+1} = H_p + \frac{1}{p+1} = H_{p+1} \quad (1.60)$$

and consequently $\|\alpha_p\|_1 = H_p$ for all $p \in \mathbb{N}$.

1.2.3 Parameter-shift rules

The second differentiation method we consider is the so-called *parameter-shift rule*. This technique makes use of the fact that the objective function $E(\theta)$ is a (finite) Fourier series, and in essence executes a discrete Fourier transform (DFT) of E in each parameter direction separately. The frequency spectrum of the series, which is required for the DFT, can be computed efficiently from the parametrized gates used in the PQC, assuming that these gates act only on logarithmically many qubits. For larger gates, either the resulting frequency spectrum (or a superset thereof) can be obtained from a structural analysis of the gate, or the gate can be decomposed and differentiated via the product rule. Note that in order

to apply a gate that is not native to the used hardware, such a decomposition is required anyways, so that the latter approach usually does not pose any additional constraints on the used gates. For digital quantum computers we may assume that native gates do not act on many qubits and hence are easy to analyse with respect to the resulting frequency spectrum.

The publication enclosed in Chap. 2 focuses on the parameter-shift rule and its generalization to unitaries of the form $U(x) = \exp(i(xG + F))$ with arbitrary Hermitian operators G and F . The perturbed case $F \neq 0$ is treated using the concept of a *stochastic shift rule* introduced in [36], leading to “improper” shift rules¹² that modify the PQC, not just its parameters. For derivations and detailed computations I refer the reader to the paper. In addition, Sec. 2.1 discusses the coverage of parameter-shift rules in the literature. Here we give a few results for completeness and discuss the relation to the central difference introduced above. As this section is an introductory summary, the contents are taken from the literature, from the publication in Chap. 2, or are immediate conclusions based on the former.

Consider a gate $U(x) = \exp(ixG)$ whose generator satisfies $G^2 = \mathbb{I}$ (but is not the identity). In this case, the restriction of E to the parameter corresponding to U takes the form

$$E(x) = a_0 + a_1 \cos(x) + b_1 \sin(x), \quad (1.61)$$

where a_0 , a_1 and b_1 are unknown real-valued coefficients. The main example of this class of gates is a Pauli rotation, i.e. G is a Pauli word. For this cost function the original parameter-shift rule [62] reads

$$\partial_{\text{PS}} E(0) = \frac{1}{2} \left[E\left(\frac{\pi}{2}\right) - E\left(-\frac{\pi}{2}\right) \right], \quad (1.62)$$

which is an exact expression. It can also be used with shift values $\pm h$ other than $\pm \frac{\pi}{2}$, and for any gate whose generator has two unique eigenvalues. In this case the rule is

$$\partial_{\text{PS}} E(0) = \frac{\Omega}{2 \sin(\Omega h)} [E(h) - E(-h)], \quad (1.63)$$

where $\Omega := |\omega_2 - \omega_1|$ is the (positive) difference of the two eigenvalues of G . For arbitrary generators G of the parametrized gate (and keeping $F = 0$), the restricted objective function E has the form

$$E(x) = a_0 + \sum_{\ell=1}^R a_\ell \cos(\Omega_\ell x) + b_\ell \sin(\Omega_\ell x), \quad (1.64)$$

where $\{\Omega_\ell\}_\ell$ are the unique, positive differences of eigenvalue pairs of G , and R is the

¹² As opposed to the “proper” shift rules in [37].

Gate class	$ \Lambda(G) $	R
$R_P(x)$	2	1
$\mathbf{c}\text{-}R_P(x)$	3	2
$\prod_{j=1}^M R_{P_j}(x)$	$M + 1$	M
$\prod_{j=1}^M R_{P_j}(c_j x), c_j \neq 0$	$M + 1 \leq \Lambda(G) \leq 2^M$	$M \leq R \leq 2^M - 1$
$U(x) = \exp(ixG) \in \mathbb{C}^{2^m}$	$1 \leq \Lambda(G) \leq 2^m$	$0 \leq R \leq \frac{4^m - 2^m}{2}$

Table 1.1: Some classes of gates, the spectrum size $|\Lambda(G)|$ of their generator G and the number of frequencies R in the resulting Fourier series when using the gate in an objective function based on an expectation value. R_P denotes a Pauli rotation, i.e. a gate $R_P(x) = \exp(ixP)$ with P a Pauli word, and $\mathbf{c}\text{-}R_P$ is such a rotation controlled by an arbitrary number of other qubits. Products of Pauli rotations typically occur in form of layers of single-qubit rotations on distinct qubits, but for rows three and four we only assume that none of the rotations can be combined into an operation with reduced frequency spectrum; $|\Lambda(G)|$ and R may be smaller, should this be possible. If all Pauli words involved in such a product of rotations commute, the generator of the unitary is the sum of the words. This is not true if the $\{P_j\}$ do not commute, but the frequency spectrum will remain the same.

number of these differences. Both the structure of these frequencies Ω_ℓ and R depend heavily on details of G and will be relevant when assessing the cost of the parameter-shift rule. For U acting on m qubits, we have $0 \leq R \leq \frac{4^m - 2^m}{2}$.

In the following we discuss a few examples of gates and the corresponding number R of frequencies in the spectrum; some classes of gates are summarized in Tab. 1.1. While exponentially large frequency spectra (in m) are possible in principle, most practically relevant gates result in much smaller R : Pauli rotations, as mentioned above, yield $R = 1$ and controlled Pauli rotations contribute $R = 2$ frequencies while both may act on any number $m \leq N$ of qubits. In section 5.1 of the publication in Chap. 2 we discuss parametrized layers of quantum gates in QAOA, which act on all N qubits but obey $R_{\text{even}} = N$ and $R_{\text{odd}} \leq \lfloor \frac{N^2}{4} \rfloor$ for the considered optimization problem.

While these gates result in polynomially many frequencies, it is easy to construct examples that actually have exponentially many frequencies: A layer of single-qubit Pauli- Z rotations with rescaled inputs $2^j x$

$$U(x) = \bigotimes_{j=1}^m R_Z^{(j)}(2^j x) \quad \text{is generated by} \quad G = \sum_{j=1}^m 2^j Z^{(j)}. \quad (1.65)$$

G has 2^m eigenvalues $\left\{ \sum_{j=1}^m k_j 2^j \mid k_j \in \{\pm 1\} \forall 1 \leq j \leq m \right\}$, from which we can generate $2^m - 1$ positive differences

$$\{\Omega_\ell\}_\ell = \left\{ \sum_{j=1}^m k_j 2^j \mid k_j \in \{0, 2\} \forall 1 \leq j \leq m, \mathbf{k} \neq \mathbf{0} \right\}. \quad (1.66)$$

Gates that saturate the bound $R \leq 2^{2^m - 1} - 2^m$ cannot be constructed from products of

Pauli rotations¹³. An example of such a gate would be the one generated by an operator with spectrum $\Lambda(G) = \{2^j | 1 \leq j \leq 2^m\}$.

For the general objective function in Eq. (1.64), the parameter-shift rule from the publication in Chap. 2 reads

$$\partial_{\text{PS}} E(0) = \sum_{\mu=1}^R y_{\mu} [E(x_{\mu}) - E(-x_{\mu})], \quad (1.67)$$

where $\pm x_{\mu}$ are the used shifts and y_{μ} are the coefficients e.g. resulting from the DFT that is underlying the shift rule for equidistant frequencies. In principle, the shifts do not have to be chosen symmetrically around 0, but doing so reduces the number of shifts by one. In [27] the author proves that optimal shift rules for any frequency spectrum with $2R$ shifts exist. It is not shown that they need to be antisymmetric as in Eq. (1.67) above, but we can, should this provide an advantage, antisymmetrize a parameter-shift rule at the cost of using more unique shifts, by subtracting the rule with $(\{-x_{\mu}\}, \{y_{\mu}\})$ from the original one that uses $(\{x_{\mu}\}, \{y_{\mu}\})$. While this will not increase the average MSE of the rule, the resulting number of shifts will of course be larger than $2R$. As parameter-shift rules are exact, the MSE of the above estimator consists of the variance alone:

$$\varepsilon^2 [\partial_{\text{PS}} \hat{E}(0)] = \mathbb{V} [\partial_{\text{PS}} \hat{E}(0)] \quad (1.68)$$

$$= \sum_{\mu=1}^R y_{\mu}^2 [\mathbb{V} [\hat{E}(x_{\mu})] + \mathbb{V} [\hat{E}(-x_{\mu})]] \quad (1.69)$$

$$= \sigma^2 \sum_{\mu=1}^R 2y_{\mu}^2 \left(\frac{1}{s_{\mu,+}} + \frac{1}{s_{\mu,-}} \right) \quad (1.70)$$

$$= \frac{4 \|\mathbf{y}\|_1^2 \sigma^2}{s}. \quad (1.71)$$

Here we again averaged the variance of $\hat{E}(x)$ over the domain of E and assumed the optimal shot allocation to the $2R$ function evaluations. The former step is justified because the variances at the shifted positions contribute linearly to $\varepsilon^2 [\partial_{\text{PS}} \hat{E}(0)]$ and we aim at a variant of the parameter-shift rule that performs best on average, but maybe not at a specific parameter position. The optimal shot allocation for this average rule is the same as the practical allocation so that it can be implemented in practice because it only depends on the known coefficients \mathbf{y} and no longer on σ . When treating shift rules as (Borel) measures as in [27, 37], the variance arises from the total variation norm of the considered shift rule.

As the frequencies Ω_{ℓ} and the shifts x_{μ} in general do not have any particular structure, there is no closed form expression for the coefficients y_{μ} or their norm $\|\mathbf{y}\|_1$. However, for the special case of equidistant frequencies $\Omega_{\ell} = \ell\Omega$ with some base frequency Ω and

¹³ This can be shown by looking at the combinations of eigenvalues of rescaled Pauli operators directly, which are too regular to create $\mathcal{O}(4^m)$ distinct frequencies.

equidistant shifts $x_\mu = \frac{2\mu-1}{2R\Omega}\pi$, the DFT coefficients are known and we have

$$y_\mu = \frac{(-1)^{\mu-1}\Omega}{4R \sin^2\left(\frac{2\mu-1}{4R}\pi\right)}, \quad (1.72)$$

which can be shown to be optimal [27]. In App. A.4 of the publication in Chap. 2 we compute¹⁴ $\|\mathbf{y}\|_1 = R\Omega$ for this recipe and arrive at

$$\varepsilon^2[\partial_{\text{PS}}\hat{E}(0)] = \frac{\sigma^2 R^2 \Omega^2}{s}. \quad (1.73)$$

As we can see, the error grows linearly with the largest frequency $R\Omega$. In contrast to the order of the finite difference stencil, neither R nor Ω are freely chosen parameters, but are dictated by the gates used in the PQC and their generators. It can be shown that this is optimal, and that the largest frequency also dictates the cost of an optimal shift rule if the frequency spectrum is not equidistant as above, and even if the frequencies are incommensurable [27]. Eq. (1.73) therefore holds for the optimal rule more generally, even though there need not be a closed-form expression for the shifts $\{x_\mu\}$ or coefficients $\{y_\mu\}$.

We leave all further details on the parameter-shift rule, as well as extensions to higher-order derivatives in one and two parameter dimensions, to the publication in Chap. 2 and references therein. More details on the literature presented a variety of perspectives and differentiation rules can be found in Sec. 2.1. The following section presents a perspective on the central difference and the two-term shift rule as special choices from a larger family of differentiation recipes, and Sec. 1.2.8 contains a numerical comparison of differentiation techniques regarding their MSE.

1.2.4 Antisymmetric two-term recipes

In this section we look at derivative estimators based on antisymmetric recipes using two shifts, allowing us to analyse the central difference and the two-term parameter-shift rule as subsets of a two-parameter family of derivative recipes, as anticipated e.g. in [56]. One might notice that the central difference recipe in Eq. (1.38) and the two-term parameter-shift rule with variable shift in Eq. (1.63) have the same functional form, which more generally can be written as

$$\partial_{\text{two-term}}E(0) := y_1(E(h) - E(-h)), \quad (1.74)$$

i.e. they are antisymmetric recipes with two terms for a particular relation between y_1 and h , namely $y_1(h) = 1/(2h)$ and $y_1(h) = \Omega/(2 \sin(\Omega h))$ for ∂_{cent} and ∂_{PS} , respectively. The

¹⁴The factor Ω was set to 1 in the publication, here we write it out explicitly.

MSE of these antisymmetric two-term recipes is

$$\varepsilon^2 \left[\partial_{\text{two-term}} \hat{E}(0) \right] = \left(\sum_{\ell=1}^R b_\ell (2y_1 \sin(\Omega_\ell h) - \Omega_\ell) \right)^2 + y_1^2 \left(\frac{\sigma^2 [\hat{E}(h)]}{s_+} + \frac{\sigma^2 [\hat{E}(-h)]}{s_-} \right). \quad (1.75)$$

The central difference is unique in this family in that it computes the lowest-order contribution of the derivative correctly, whereas the parameter-shift rule is the unique unbiased choice for cost functions with a single frequency in the Fourier series.

In [56] the authors discuss the comparison between these two recipes as well, and consider an additional tradeoff which reduces its variance at the cost of introducing a bias. To this end, the parameter-shift rule is rescaled and the authors continue to show that the optimal parameters are¹⁵

$$y_1(h) = \frac{1}{2} \frac{\sin(\Omega h)}{\sin^2(\Omega h) + \frac{\sigma^2}{sb_1^2}} \quad \text{and} \quad h = \frac{\pi}{2\Omega}, \quad (1.76)$$

assuming for the optimal shift that the variance of \hat{E} is constant, or alternatively that we use the practical shot allocation strategy introduced above. We denote this optimal anti-symmetric two-term recipe as ∂_{opt} . If we investigate the MSE of the recipe *on average* over the parameter domain, we indeed obtain the same result as if we had replaced the local variance by its average right away. For the parameter-shift rule both shot allocation strategies are equivalent so that the analysis can be transferred to the optimal allocation, but for any other $\left(h, y_1(h) \neq \frac{\Omega}{2 \sin(\Omega h)} \right)$ the strategies differ¹⁶. We will compare the anti-symmetric two-term recipes to higher-order finite differences and the LCU-based estimator (see next section) for the case of a single frequency ($R = 1$) in E in Sec. 1.2.8.

1.2.5 Linear combination of unitaries

In the previous sections we discussed finite differences, which do not take the functional form of the differentiated function into account. We also discussed parameter-shift rules, which make use of the fact that E is a Fourier series with known frequency spectrum but do not consider or modify the structure of the PQC. There is another differentiation method that is more specific to quantum circuits and is based on a generalization of the Hadamard test to compute quantum state overlaps [63, 64, 65]. We will refer to this method as linear combination of unitaries (LCU). To understand this method, we take a step back and look

¹⁵ The appropriate translation from [56] to our notation is $2N \mapsto s$, $s \mapsto h$ and $g_j^2 \mapsto b_1^2 \Omega^2$, where we inserted the frequency Ω in the required places.

¹⁶ This means that we choose to analyse the MSE as encountered in practice where we do not know the Fourier coefficients. However, should there be a way to estimate the coefficients as discussed e.g. in [57], this analysis might be slightly too pessimistic.

at how the objective function E came to be:

$$E(\boldsymbol{\theta}) = \langle 0 | C(\boldsymbol{\theta})^\dagger H C(\boldsymbol{\theta}) | 0 \rangle. \quad (1.77)$$

For the previous methods, it did not make a difference whether a given parameter θ_k contributes to the circuit in a single unitary of the form $\exp(ixG)$ or in multiple such gates, because we only used the functional form of the resulting objective function E . Here we assume the former, and thus each gate carries a unique variable θ_k and is differentiated separately. Assuming this structure, we have

$$E_k(x) = \langle 0 | C_{[:k+1]}^\dagger U(x)^\dagger C_{[k+1:]}^\dagger H C_{[k+1:]} U(x) C_{[:k+1]} | 0 \rangle \quad (1.78)$$

$$= \langle \phi | U(x)^\dagger B U(x) | \phi \rangle, \quad (1.79)$$

where $|\phi\rangle := C_{[:k+1]} | 0 \rangle$ is the state prepared by all gates up to (including) $U(\theta_k)$ and $B := C_{[k+1:]}^\dagger H C_{[k+1:]}$ is the Hermitian obtained by applying all gates after U to H in the Heisenberg picture. Then the derivative is

$$E'_k(0) = -2\Im [\langle \phi | B G | \phi \rangle]. \quad (1.80)$$

We now drop the index k as we did before. Consider the following expression using $N + 1$ qubits

$$\partial_{\text{LCU}} E(0) := -2 \langle + | \langle \phi | \text{c-}G^\dagger (Y \otimes B) \text{c-}G | + \rangle | \phi \rangle \quad (1.81)$$

$$= i (\langle \phi | B G | \phi \rangle - \langle \phi | G B | \phi \rangle) \quad (1.82)$$

$$= E'(0). \quad (1.83)$$

Here we wrote $\text{c-}G$ for the controlled generator, i.e. $\text{c-}G = |0\rangle\langle 0| \otimes \mathbb{I} + |1\rangle\langle 1| \otimes G$.

This estimator can be realized in the following way (see Fig. 1.1): add a single auxiliary qubit to the circuit, which is prepared in the state $|+\rangle$. The original circuit is executed on the N -qubit register as for the computation of E itself, but after executing the differentiated gate $U(\theta_k)$, the generator is applied controlled by the auxiliary qubit, i.e. $\text{c-}G$ is inserted after $U(\theta_k)$. The auxiliary qubit is then measured in the Pauli- Y basis along with the Hamiltonian terms on the main register. This method was proposed e.g. in [66]. Alternatively we may apply the Hamiltonian H to the N -qubit register, controlled on the auxiliary qubit, once the original circuit is completed and only measure the auxiliary qubit [67, 68]. This is equivalent because

$$(\mathbb{I} \oplus H)(Y \otimes \mathbb{I})(\mathbb{I} \oplus H) = (Y \otimes \mathbb{I})(H \oplus H) = Y \otimes H, \quad (1.84)$$

which means applying the controlled gate $\text{c-}H$ before measuring Y on the auxiliary qubit is the same as measuring $Y \otimes H$ on all $N + 1$ qubits. Considering a grouping of H into sets

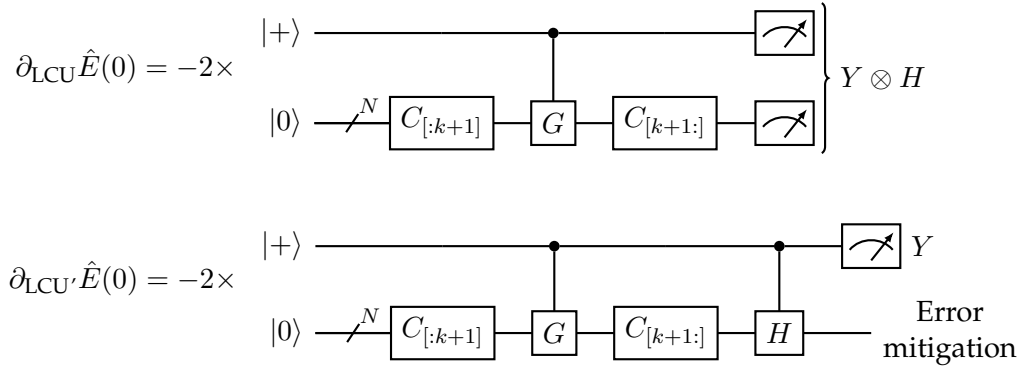


Figure 1.1: Quantum circuits to compute the derivative of an objective function based on a PQC C , using an auxiliary qubit and the controlled application of the generator, $c\text{-}G$. For the first approach, only one controlled operation is required, and the observable $Y \otimes H$ is measured in the end. For the second approach, the measurement is preceded by a controlled application of the terms in an LCU of the Hamiltonian H and only Y is measured on the auxiliary qubit, leaving room for error mitigation protocols on the main register.

of commuting Pauli words (c.f. Sec. 1.1), the first implementation allows us to measure the derivative using the same grouping. The second approach, as it requires H to be unitary, only supports one Pauli word (or another unitary from an LCU of H) to be measured per circuit execution. Alternatively, a grouping into sets of anticommuting Pauli words may be used to construct a decomposition of H into fewer unitaries. Another disadvantage of the second implementation of the Hadamard test is the need to apply the controlled gate $c\text{-}H$, or an LCU thereof, which increases the multi-qubit gate count and poses additional hardware constraints in terms of connectivity; the first method only requires some auxiliary qubit to be connected to the qubits on which the differentiated gate acts. To implement $c\text{-}H$ for the second method, however, the qubits of each unitary term in the Hamiltonian need to be reachable from such an auxiliary qubit in addition.

An advantage of the second method is that the main N -qubit register can be used for error mitigation protocols, such as echo verification, or verified phase estimation [45, 46, 47]. As an alternative to a grouping into Pauli words, other decompositions of H into unitaries are possible, and indeed favourable for the statistics of LCU-based measurement strategies [44].

The circuits for both approaches are shown in Fig. 1.1. There, we do not explicitly indicate the grouping of H into commuting sets of Pauli words or unitaries for the first and second approach respectively, but implicitly assume that H is measured or $c\text{-}H$ is applied via a suitable decomposition, respectively. Both methods for the LCU-based derivative only yield a valid circuit if G is unitary, or equivalently $G^2 = GG^\dagger = \mathbb{I}$ (due to Hermiticity and unitarity). Note that scalar prefactors can simply be accounted for in classical post-processing and a global phase in U does not matter physically. With these two relations the condition $G^2 = \mathbb{I}$ becomes equivalent to $|\Lambda(G)| = 2$, the condition for the original

parameter-shift rule to hold (see Sec. 1.2.3). We will discuss possible extensions of the estimator to more general G further below.

The expression in Eq. (1.81) is exact, and therefore the estimator $\partial_{\text{LCU}}\hat{E}(0)$ is unbiased. Its variance can be calculated via

$$\mathbb{V}\left[\partial_{\text{LCU}}\hat{E}(0)\right] = \frac{4}{s} \langle + | \langle \phi | \mathbf{c} \cdot G^\dagger (\mathbb{I} \otimes B^2) \mathbf{c} \cdot G | + \rangle | \phi \rangle - \frac{1}{s} E'(0)^2 \quad (1.85)$$

$$= \frac{2}{s} \left(\langle \phi | \oplus \langle \phi | G^\dagger \right) (B^2 \oplus B^2) (|\phi\rangle \oplus G|\phi\rangle) - \frac{1}{s} E'(0)^2 \quad (1.86)$$

$$= \frac{2}{s} \left(\langle 0 | C(\boldsymbol{\theta})^\dagger H^2 C(\boldsymbol{\theta}) | 0 \rangle + \langle 0 | C_G(\boldsymbol{\theta})^\dagger H^2 C_G(\boldsymbol{\theta}) | 0 \rangle \right) - \frac{1}{s} E'(0)^2 \quad (1.87)$$

$$= \frac{1}{s} \langle H^2 \rangle''(0) + \frac{4}{s} \langle H^2 \rangle(0) - \frac{1}{s} E'(0)^2, \quad (1.88)$$

where C_G denotes the original PQC with the generator G inserted after U , and we used $G^2 = \mathbb{I}$ in the last step. Note that there is only a single circuit to be executed per Hamiltonian term, so that the shots for the derivative do not need to be split up further than for measuring E itself.

The constraint for G (together with its Hermiticity, and excluding the global phase gate with $G \propto \mathbb{I}$) moreover implies that $\Omega = 2$, i.e. the two generator eigenvalues differ by 2. Therefore we know that $\langle H^2 \rangle(x) = \tilde{a}_0 + \tilde{a}_1 \cos(2x) + \tilde{b}_1 \sin(2x)$ and that E takes the same functional form with coefficients a_0, a_1 and b_1 . In practice we are interested in the point $x = 0$ when computing a single derivative, but like before we here want to evaluate the derivative estimator *on average* over the domain of θ_k , and therefore write

$$\mathbb{V}\left[\partial_{\text{LCU}}\hat{E}(x)\right] = \frac{4}{s} \left(-\tilde{a}_1 \cos(2x) - \tilde{b}_1 \sin(2x) + \tilde{a}_0 + \tilde{a}_1 \cos(2x) + \tilde{b}_1 \sin(2x) \right) - \frac{1}{s} (-2a_1 \sin(2x) + 2b_1 \cos(2x))^2 \quad (1.89)$$

$$= \frac{4}{s} \left[\tilde{a}_0 - (a_1 \sin(2x) - b_1 \cos(2x))^2 \right]. \quad (1.90)$$

Averaging over the domain $[0, \pi]$, we get a variance of

$$\varepsilon^2 \left[\partial_{\text{LCU}}\hat{E}(0) \right] = \frac{4}{s} \left[\tilde{a}_0 - \frac{a_1^2 + b_1^2}{2} \right] \quad (1.91)$$

for the LCU-based derivative. For gates with $\Omega \neq 2$ that need to be rescaled in order to admit this differentiation technique we get an additional factor of $\Omega^2/4$ in this variance, i.e.

$$\varepsilon^2 \left[\partial_{\text{LCU}}\hat{E}(0) \right] = \frac{\Omega^2}{s} \left[\tilde{a}_0 - \frac{a_1^2 + b_1^2}{2} \right] = \frac{\Omega^2}{s} (\sigma^2 + a_0^2) \geq \varepsilon^2 \left[\partial_{\text{PS}}\hat{E}(0) \right], \quad (1.92)$$

with the average variance σ^2/s of $\hat{E}(x)$ itself.

For the second variant of the LCU-based estimator the shot budget needs to be split among the unitary terms that compose H . The resulting variance differs from the variance

of the first LCU-based approach by covariances, and by the deviating prefactors due to the shot allocation. Furthermore, it depends on the decomposition of H which need not be the same as the grouping into sets of commuting Pauli words. All this makes an analytic comparison to the other estimators difficult. Should H be unitary itself, the variances of the two LCU-based approaches become the same:

$$\mathbb{V} \left[\partial_{\text{LCU}} \hat{E}(0) \right] = \frac{4}{s} \left(\langle + | \langle \phi | \mathbf{c} \cdot G^\dagger C_{[k+1:]}^\dagger \mathbf{c} \cdot H^\dagger \right) (Y \otimes \mathbb{I})^2 (\mathbf{c} \cdot H C_{[k+1:]} \mathbf{c} \cdot G | + \rangle | \phi \rangle) - \frac{1}{s} E'(0)^2 \quad (1.93)$$

$$= \frac{1}{s} (4 - E'(0)^2) \quad (1.94)$$

$$= \frac{4}{s} \left(\langle + | \langle \phi | \mathbf{c} \cdot G^\dagger \right) (\mathbb{I} \otimes B^2) (\mathbf{c} \cdot G | + \rangle | \phi \rangle) - \frac{1}{s} E'(0)^2 \quad (1.95)$$

$$= \mathbb{V} \left[\partial_{\text{LCU}} \hat{E}(0) \right]. \quad (1.96)$$

As an alternative to using an auxiliary qubit, there is the possibility to use a quantum state within the original register as reference state, as long as it is not used in the computation itself [42, 69]. This is of relevance in circuits that leave certain parts of the state space untouched because they respect symmetry sectors of the Hamiltonian, e.g. in applications for quantum chemistry.

As discussed, the LCU-based derivative estimator described above only is applicable to gates with $G^2 = \mathbb{I}$. However, there is a straightforward extension using a decomposition of G into a sum of Hermitian unitaries¹⁷

$$G = \sum_{r=0}^{K-1} \tilde{\gamma}_r P_r, \quad (1.97)$$

and the fact that Eq. (1.80) is linear in G (ignoring that G is used to prepare $|\phi\rangle$, of course). The estimator in Eq. (1.81) is then applied to each P_r and the terms are combined classically including the (real-valued) weights $\tilde{\gamma}_r$. The available shot budget has to be allocated to the different estimators according to their variance and the weights. As discussed before, the variance is not known beforehand. For shifted evaluations of E , averaging over the parameter space removed this problem, because then all evaluations have the same mean variance. For the different terms in G , however, this is not possible and we are not able to predict the relation between the average variances that we try to balance with the shot allocation. Therefore, we allocate the shots in proportion to the weights $\gamma_r = |\tilde{\gamma}_r|$ alone

¹⁷ This again is an LCU, but not the one that gives this estimator its name.

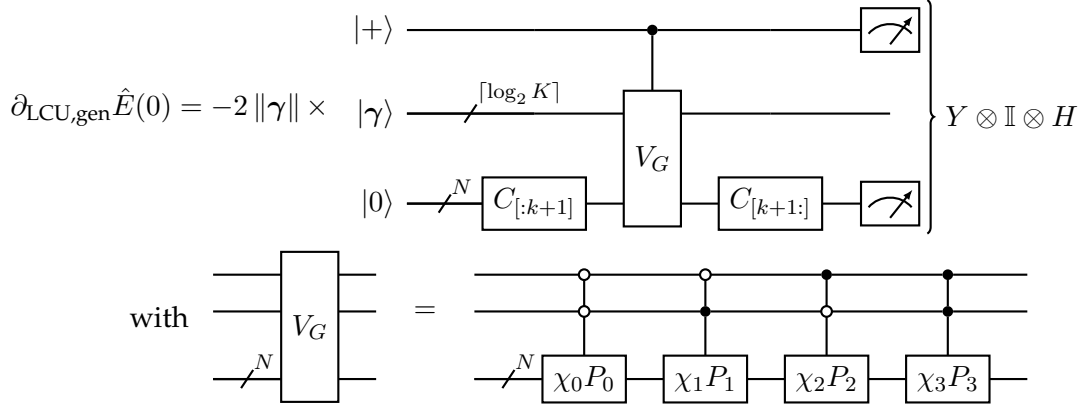


Figure 1.2: Quantum circuit similar to the first circuit in Fig. 1.1, but for nonunitary G that can be decomposed as $G = \sum_{r=1}^K \gamma_r \chi_r P_r$ with $\gamma_r > 0$, $\chi_r \in \{\pm 1\}$ and unitaries P_r . For K terms in the decomposition $\lceil \log_2 K \rceil$ additional qubits and K multi-controlled operations are needed; the operation V_G is shown for $K = 4$. Similar to ∂_{LCU} for unitary G , the measurement of $Y \otimes \mathbb{I} \otimes H$ can be replaced by applying the operation $c\text{-}(\mathbb{I} \otimes H)$ and measuring Y on the first auxiliary qubit alone.

and obtain the variance

$$\mathbb{V} [\partial_{\text{LCU}} \hat{E}(0)] = \frac{\|\gamma\|_1}{s} \sum_{r=0}^{K-1} \gamma_r \left[2 \left(\langle H^2 \rangle + \langle 0 | C_{P_r}(\theta)^\dagger H^2 C_{P_r}(\theta) | 0 \rangle \right) \right. \quad (1.98)$$

$$\left. - (-2\Im [\langle \phi | B P_r | \phi \rangle])^2 \right] \quad (1.99)$$

$$= \frac{2\|\gamma\|_1^2}{s} \langle H^2 \rangle + \frac{2\|\gamma\|_1}{s} \sum_{r=0}^{K-1} \gamma_r \left(\langle H^2 \rangle_{P_r} - 2\Im [\langle \phi | B P_r | \phi \rangle]^2 \right), \quad (1.100)$$

where $\langle \cdot \rangle_{P_r}$ abbreviates the expectation value with respect to the modified circuit C_{P_r} .

Another option is to implement the decomposition of G on the quantum computer itself, using additional $\lceil \log_2(K) \rceil$ auxiliary qubits. The additional qubits are used to implement the summands $\{P_r\}$ in proportion to $\{\sqrt{\gamma_r}\}$, and the first auxiliary qubit implements the generalized Hadamard test itself as before. This construction is shown in Fig. 1.2 and can be found in a similar form e.g. in [70, Fig. 3]. It does not decompose G as in Eq. (1.97) but via

$$G = \sum_{r=0}^{K-1} \gamma_r \chi_r P_r, \quad \text{with } \chi_r = \text{sgn}(\tilde{\gamma}_r). \quad (1.101)$$

The multi-controlled operations as well as the auxiliary weights state are summarized by

$$V_G := |0\rangle\langle 0| \otimes \mathbb{I} \otimes \mathbb{I} + |1\rangle\langle 1| \otimes \sum_{r=0}^{K-1} (|r\rangle\langle r| \otimes (\chi_r P_r)), \quad (1.102)$$

$$|\gamma\rangle := \sqrt{\|\gamma\|_1^{-1}} \sum_r \sqrt{\gamma_r} |r\rangle, \quad (1.103)$$

where the tensor product in V_G is between the auxiliary qubit for the Hadamard test, the ‘‘term selection qubits’’, and the original qubit register. The gradient estimator realized in this circuit, multiplied by $-2 \|\gamma\|_1$, is

$$\mathbb{E} \left[\partial_{\text{LCU,gen}} \hat{E}(0) \right] = -2 \|\gamma\|_1 \langle + | \langle \gamma | \langle \phi | V_G^\dagger (Y \otimes \mathbb{I} \otimes B) V_G | + \rangle | \gamma \rangle | \phi \rangle \quad (1.104)$$

$$= i \sum_{r=0}^{K-1} \sqrt{\gamma_r^2} \left[\langle \phi | B (\chi_r P_r | \phi \rangle) - \left(\langle \phi | \chi_r P_r^\dagger \right) B | \phi \rangle \right] \quad (1.105)$$

$$= -2\Im \left[\langle \phi | B G | \phi \rangle \right]. \quad (1.106)$$

Both the preparation of the weighted state $|\gamma\rangle$ and the K multi-controlled gates in V_G may pose serious challenges for near-term quantum computing hardware, depending on K and the structure of γ . In this approach, the shot budget is allocated like for the energy measurement, leading to the variance

$$\mathbb{V} \left[\partial_{\text{LCU,gen}} \hat{E}(0) \right] = \frac{4 \|\gamma\|_1^2}{s} \langle + | \langle \gamma | \langle \phi | V_G^\dagger (\mathbb{I} \otimes \mathbb{I} \otimes B^2) V_G | + \rangle | \gamma \rangle | \phi \rangle \quad (1.107)$$

$$- \frac{1}{s} E'(0)^2 \quad (1.108)$$

$$= \frac{2 \|\gamma\|_1^2}{s} \langle H^2 \rangle + \frac{2 \|\gamma\|_1}{s} \sum_{r=0}^{K-1} \gamma_r \langle H^2 \rangle_{P_r} - \frac{1}{s} E'(0)^2. \quad (1.109)$$

We observe that the first two terms are identical to the variance of the classically combined estimators for each P_r in Eq. (1.100) but the third terms differ. To compare them, we write

$$g_r := -2\Im \left[\langle \phi | B \chi_r P_r | \phi \rangle \right] \quad (1.110)$$

$$\Rightarrow E'(0) = \gamma \cdot \mathbf{g} \quad (1.111)$$

$$\mathbb{V} \left[\partial_{\text{LCU,gen}} \hat{E}(0) \right] - \mathbb{V} \left[\partial_{\text{LCU}} \hat{E}(0) \right] = \frac{1}{s} (\|\gamma\|_1 \gamma \cdot \mathbf{g}^{\odot 2} - (\gamma \cdot \mathbf{g})^2) \quad (1.112)$$

$$= \frac{1}{s} \sum_{r < q} \gamma_r \gamma_q (g_r - g_q)^2 \geq 0, \quad (1.113)$$

where the inequality holds because $\gamma_r > 0$, $\forall 0 \leq r \leq K-1$. We therefore get a larger (or equal) variance from the generalized LCU-based estimator, which also comes at the cost of additional auxiliary qubits and potentially very NISQ-unfriendly multi-control operations and connectivity requirements. As any decomposition that can be implemented in

this way also can be computed from separate circuits for each P_r together with classical postprocessing, we conclude that this approach (Fig. 1.2) is not favourable for measuring gradients.

1.2.6 Simultaneous perturbation stochastic approximation

Another estimator for derivatives of PQC-based functions can be implemented using the simultaneous perturbation stochastic approximation (SPSA) [71], which – similar to finite differences – does not use the structure of the objective function. However, this estimator differs from the ones discussed above in that it aims at estimating the multivariate derivative as a single object instead of its separate components. The core idea is to repeatedly shift (or perturb, considering the method’s name) all input parameters simultaneously according to some suitable probability distribution, and to compute correlated estimates for all components of the gradient from the evaluations at these shifted positions. We will not review this method in detail but note that SPSA has been employed in the context of VQAs repeatedly [72, 73, 74] and has been extended to the computation of the Fubini-Study metric, allowing an SPSA-based computation of the quantum natural gradient [75]. In our analysis in Sec. 1.2.8 we will not include SPSA either, because there we will focus on elementwise estimation of the gradient.

1.2.7 Differentiation on classical simulators

In addition to the methods discussed in Sec. 1.2.2 to 1.2.6, which mostly can be executed on quantum machines with no or minor additional hardware requirements, there are additional methods for differentiating PQCs when simulating them on classical computers.

Simulating PQCs and variational algorithms is still a major part of NISQ research. Therefore, while these differentiation methods on simulators are unlikely to be of interest for quantum computing at larger scales, they offer relevant performance improvements in today’s numerical experiments.

I will first present two simulator-only differentiation techniques that are computationally cheaper than simulating the hardware-compatible methods directly. For this we assume a (to numeric precision) exact calculation of all quantities. Afterwards I discuss the emulation of shot-noisy gradients on classical simulators, which poses an interesting challenge for near-term simulations.

Automatic differentiation A powerful technique on classical computers is automatic differentiation, sometimes abbreviated as autodiff. It can be understood as the computer implementation of symbolic differentiation in non-symbolic programming languages and mimics differentiation on paper by using the chain rule iteratively. As such, it does not introduce truncation errors beyond the numerical precision cutoff present in straightfor-

ward function evaluations and often exhibits much better performance than finite difference schemes.

Autodiff is an essential tool in most programs in machine learning and one of the core functionalities of major machine learning programming libraries like Tensorflow [76], PyTorch [77], JAX [78] and autodiff (C++) [79]. This wide availability made it easy for quantum simulation frameworks to integrate autodiff functionalities early on, prominent examples include PennyLane [80], Yao.jl [81], Tensorflow Quantum [82] and Qiskit [83].

There are two generic *modes* to compute the full derivative of a function $f : \mathbb{R}^n \rightarrow \mathbb{R}^p$ with autodiff, dubbed *forward* and *reverse mode*. In essence they differ by the order in which the iterative chain rule is resolved, which makes their computational cost dependent on the number of inputs n and outputs p .

The forward mode sweeps through the computational graph in the same order as the original computation of $f(\theta)$. This has to be done once for each input dimension, calculating one column of the Jacobian $\partial f_i / \partial \theta_j$ per sweep. In contrast, the reverse mode starts at the output value of f and resolves the chain rule while moving towards the input nodes of the computational graph. This corresponds to computing the Jacobian row by row, and in particular, it requires only one pass through the computational graph of f if $p = 1$, i.e. for scalar output functions like the objective function $E(\theta)$.

While this makes reverse mode a fast technique for PQC-based functions, it requires a significant amount of additional memory in order to store intermediate results of the computation. This is particularly problematic for simulators of full quantum registers because these intermediate results are complex-valued state vectors with 2^N components. This significantly limits the system size for which a typical gradient computation fits into the fast random access memory (RAM) of a classical computer. The memory overhead typically scales with n , the number of (input) parameters [84], so that we may expect a memory requirement of $\mathcal{O}(n2^N)$. For reversible computations, to which quantum circuits belong, the memory overhead can be reduced to a constant overhead [85], which is exploited e.g. in the library Yao.jl [81] and leads us to the following method for simulators:

Adjoint method A second differentiation method that is available on state vector simulators but not on quantum computing hardware is the so-called adjoint method [81, 86]. Due to its relation to autodiff techniques for reversible computations, it is also called “reverse mode”, but here we adapt the name “adjoint method” in order to separate it from the method in the paragraph above. The adjoint method makes use of the reversibility of quantum circuits and the structure of PQC-based functions, achieving computational cost and memory requirements similar to those for evaluations of the objective function itself¹⁸.

The core idea of the method is as follows: first, compute the state $|\psi(\theta)\rangle$ prepared

¹⁸ At the highest density of parametrized gates in the circuit and when combined optimally with the computation of E itself, it requires leading-order additional cost of $3n$ applications of gate-like matrices, as well as 2 additional state vectors of memory.

by the original PQC – or retrieve it from cache if $E(\boldsymbol{\theta})$ itself was computed before – and copy it. Second, apply the Hamiltonian to one of the copies of $|\psi(\boldsymbol{\theta})\rangle$, obtaining $|\phi\rangle = H|\psi(\boldsymbol{\theta})\rangle$. If E was not computed yet, it can be obtained at this point, at the cost of a single inner product between $|\phi\rangle$ and the second copy of $|\psi(\boldsymbol{\theta})\rangle$. Third, iteratively apply the inverses of the gates from the PQC in reverse order to both $|\psi(\boldsymbol{\theta})\rangle$ and $|\phi\rangle$. Whenever a parametrized gate is encountered during this process, copy one of the states, apply the generator of the parametrized gate to it, and compute the inner product with the other state. This inner product corresponds to the gradient entry up to a constant that depends on notation conventions and the choice of state to which the generator was applied. The copy may be discarded/overwritten after computing the inner product. Ref. [86] includes a pedagogic demonstration of the idea behind the adjoint method and discusses a series of extensions to more general operations, observables and to density matrix simulators. Interested readers may refer to this technical note and its references for details on the method.

To the best of my knowledge, the adjoint method is the fastest differentiation method on classical simulators of exact PQC-based objective functions that does not require impractical amounts of memory, in particular for the domain with more than ~ 20 qubits [76, 80, 81]. Not only the asymptotic scaling of the adjoint method but also its costs for small numbers of qubits and parameters are competitive to other methods. Nonetheless, differentiation methods based on shifted evaluations of the unmodified objective function like the parameter-shift rule or finite differences may be advantageous for small N and n , depending on details of the implementation, programming language and use case. For example just-in-time compilation, the usage profile of computing E and ∇E respectively, and the cost of manipulating state vectors outside of a valid quantum circuit structure may have a significant impact on the performance in practice.

Computing gradients with shot noise State vector simulators usually compute the (numerically) exact probability distribution associated with the quantum state that is prepared by the simulated PQC, and produce a statistical estimator by sampling from the obtained distribution. That is, we obtain exact quantities and have to do additional work in order to get a realistic shot-noisy estimator of these quantities. There are exceptions to this behaviour of simulators, such as the tensor network-based simulator *TNI* by AWS Braket [87]. On a real QPU, the situation is reversed and we only ever have access to noisy estimates for the quantities of interest, so that we require larger shot budgets to obtain more accurate approximations.

When investigating noisy intermediate-scale quantum (NISQ) algorithms, optimally both the statistical uncertainties and the noise inflicted by the quantum device should be considered and included in simulations. In particular the latter source of noise often is excluded in research when examining full optimization workflows because it requires a full density matrix simulation, which costs much more time and memory than state vec-

tor simulations¹⁹. Effectively, this separates investigations of higher-level algorithms from research on error mitigation and similar protocols. While it is not guaranteed that there are no strong interactions between, say, an optimization algorithm and noise mitigation techniques, this separation makes numerical experiments of relevant size possible in the first place.

The statistical error, or shot noise, is an essential property of the estimators obtained from quantum devices and the required precision to which objectives are measured crucially determine the cost of quantum algorithms. Consequently an analysis using numerically exact expectation values differs from one that considers realistic statistical properties of the used estimators: On one hand, statistical fluctuations make the output of e.g. the objective function less precise and allocating measurements optimally to different (steps of) subroutines of a NISQ algorithm is a difficult task but crucially influences the runtime (scaling) [60]. The number of required samples is considered one of the major challenges e.g. for variational quantum eigensolvers (VQEs) [12, 42, 51, 52]. Therefore, running simulations without any statistical noise may drastically overestimate the performance of NISQ algorithms and in particular the accuracy of the final result. On the other hand, fluctuations of computed quantities can be beneficial e.g. for optimization routines [88], which are essential in many VQAs, so that shot-free simulation might actually underestimate convergence properties or the stability with respect to initial conditions of the optimization. The impact of these two discrepancies between shot-free simulations and shot-based estimators, simulated or from a QPU, is unknown and accordingly it is desirable to include shot noise whenever feasible. Nonetheless, it is frequently omitted in application-oriented research that spans full optimization workflows because of the additional computational cost, as is the case in the publication attached in Chap. 3.

But why are shot-based simulations so much more costly than numerically exact experiments? For algorithms involving the differentiation of PQCs, the answer lies in the shortcuts offered by the simulator-only differentiation techniques introduced in this section: For (numerically) exact experiments, or equivalently for very large numbers of shots, the expectation values of the estimators become the only relevant quantity. The derivatives in this case can be computed via automatic differentiation or the adjoint method and thus faster than by simulating any of the hardware-ready methods. However, if we are interested in the shot-noisy estimator, these simulator-only methods do not give us access to the underlying probability distributions, which would be necessary for a fast simulation of the statistical estimator at finite shot numbers. This forces numerical experimentalists to use hardware-ready methods on classical simulators and increases the costs for simulations of shot-noisy derivatives.

We could relax our goal of reproducing the full statistics and just demand that we obtain the correct expectation value and variance for each estimator, thus merely “emulating” the shot noise. This is reasonable for sufficiently large shot numbers because the

¹⁹The additional factor for naïve dense matrix simulators is 2^N both in time and space.

central limit theorem tells us to treat the estimator of the mean, which is typically the form our objective function (derivative) comes in, as a normal random variable in that regime. Even for this relaxed goal, the task remains difficult and to the best of my knowledge there is no method to emulate shot-noisy derivatives at (approximately) the same cost as computing their numerically exact counterparts while reproducing the variance of hardware-ready methods.

In conclusion, simulating gradient-based quantum algorithms is more expensive with shot noise than without. Finding a method to compute the shot-noisy gradient at the cost of, say, the adjoint method would allow researchers to include shot noise in larger scale investigations, which in turn might be very helpful in understanding and characterizing NISQ algorithms and subroutines.

1.2.8 Estimator comparison for $R = 1$

In this section I will present a detailed comparison of the hardware-ready derivative estimators introduced in the previous sections for single-frequency ($R = 1$) objective functions. More precisely, we will consider the central difference ∂_{cent} (Eq. (1.37)), higher-order central differences $\partial_{\text{FD},p}$ (Eq. (1.43)) with $p \in \{2, 4, 10\}$, the parameter-shift rule ∂_{PS} (Eq. (1.63)), the optimal antisymmetric two-term recipe ∂_{opt} (Eq. (1.76)) and the LCU-based estimator ∂_{LCU} (Eq. (1.81)). As also analysed in [56], the optimal recipe ∂_{opt} will be a rescaled variant of the parameter-shift rule. We will assume throughout this section that the practical shot allocation is used, as it does not require any knowledge that we actually try to obtain (also see Sec. 1.2.4).

We begin with an analysis of antisymmetric two-term recipes for single-frequency functions. The variance at a shifted point in this case is (c.f. Eq. (1.28))

$$\sigma^2 \left[\hat{E}(\pm h) \right] = \tilde{a}_0 + \tilde{a}_1 \cos(\Omega h) \pm \tilde{b}_1 \sin(\Omega h) \quad (1.114)$$

$$- (a_0 + a_1 \cos(\Omega h) \pm b_1 \sin(\Omega h))^2. \quad (1.115)$$

Now assume that we fix the shift parameter h , i.e. we do not adapt it dynamically to the computation, but at most to a specific parameter index in the circuit. As we consider the practical rather than the optimal shot allocation, we distribute the shots evenly between the two shifted evaluations so that $s_{\pm} = s/2$. It makes sense to look at the average MSE across the univariate domain $X = [0, 2\pi/\Omega]$. While averaging across the entire parameter space $\times_k [0, 2\pi/\Omega_k]$ requires knowledge of the full circuit and hence seems unfeasible analytically²⁰, the simple structure of E as a function of one parameter makes it possible to determine the univariate average. Given the Fourier coefficients a_0, a_1, b_1 at 0, E takes the same functional form at any other $x \in X$, with $a_0(x) = a_0, a_1(x) = a_1 \cos(\Omega x) + b_1 \sin(\Omega x)$

²⁰ Some progress in this direction is nonetheless made in [57].

and $b_1(x) = b_1 \cos(\Omega x) - a_1 \sin(\Omega x)$, so that the average variance becomes

$$\overline{\sigma^2 \left[\hat{E}(\pm h) \right]} = \frac{\Omega}{2\pi} \int_0^{2\pi/\Omega} \left\{ \tilde{a}_0 + \left(\tilde{a}_1 \cos(\Omega x) + \tilde{b}_1 \sin(\Omega x) \right) \cos(\Omega h) \right. \quad (1.116)$$

$$\left. \pm \left(\tilde{b}_1 \cos(\Omega x) - \tilde{a}_1 \sin(\Omega x) \right) \sin(\Omega h) \right. \quad (1.117)$$

$$\left. - \left[a_0 + (a_1 \cos(\Omega x) + b_1 \sin(\Omega x)) \cos(\Omega h) \right. \right. \quad (1.118)$$

$$\left. \left. \pm (b_1 \cos(\Omega x) - a_1 \sin(\Omega x)) \sin(\Omega h) \right]^2 \right\} dx \quad (1.119)$$

$$= \tilde{a}_0 - a_0^2 - \frac{a_1^2 + b_1^2}{2} \quad (1.120)$$

$$=: \sigma^2. \quad (1.121)$$

Furthermore the average (squared) bias of the antisymmetric two-term recipe is

$$\overline{(\partial_{\text{two-term}} E(0) - E'(0))^2} = \frac{\Omega}{2\pi} \int_0^{2\pi/\Omega} (b_1 \cos(\Omega x) - a_1 \sin(\Omega x))^2 (2y_1 \sin(\Omega h) - \Omega)^2 dx \quad (1.122)$$

$$= (2y_1 \sin(\Omega h) - \Omega)^2 \frac{a_1^2 + b_1^2}{2}, \quad (1.123)$$

so that the average MSE for a function E with $R = 1$ reads

$$\varepsilon_{\text{two-term}}^2 = \begin{cases} \frac{\sigma^2}{s} \frac{\Omega^2}{\sin^2(\Omega h)} & \text{parameter-shift rule} \\ \frac{\sigma^2}{sh^2} \left(1 + \frac{s}{\rho} (\sin(\Omega h) - h\Omega)^2 \right) & \text{central difference} \\ \frac{\sigma^2}{s} \frac{\Omega^2}{\sin^2(\Omega h) + \rho/s} & \text{optimal two-term recipe} \\ \frac{\sigma^2}{s} \left(4y_1^2 + \frac{s}{\rho} (2y_1 \sin(\Omega h) - \Omega)^2 \right) & \text{general two-term recipe.} \end{cases} \quad (1.124)$$

Here we defined the fraction $\rho := \frac{2\sigma^2}{(a_1^2 + b_1^2)}$, which does not depend on h, Ω or s , skipped E and the evaluation point 0 in the notation on the LHS, and used the shifts

$$y_1(h) = \begin{cases} \Omega / (2 \sin(\Omega h)) & \text{parameter-shift rule} \\ 1 / (2h) & \text{central difference} \\ \Omega \sin(\Omega h) / (2 \sin^2(\Omega h) + 2\rho/s) & \text{optimal two-term recipe} \\ y_1 & \text{general two-term recipe.} \end{cases} \quad (1.125)$$

There is an additional simplification if we consider the parameter position θ to be sampled from a translation invariant probability (as we will do in our numerical experiment); sampling $\theta + \frac{\pi}{2\Omega} e_k$ instead of θ yields the replacement $(a_1, b_1) \mapsto (b_1, -a_1)$ so that $\frac{a_1^2 + b_1^2}{2} = b_1^2$ and $\sigma^2 = \tilde{a}_0^2 - a_0^2 - b_1^2$ on average. With this identification the fraction introduced above becomes $\rho = \frac{\sigma^2}{b_1^2}$ and the coefficient for the optimal two-term recipe in the third row of

Eq. (1.125) is the same as in Eq. (1.76), the optimal coefficient from [56]. Note that the optimal shift for both the parameter-shift rule and the optimal two-term recipe is $\pi/(2\Omega)$, whereas the optimal shift for the central difference does not have a closed-form expression but satisfies

$$(x - \sin(x)) (\sin(x) - x \cos(x)) = \frac{\rho}{s}, \text{ with } x = \Omega h_{\text{cent}}^*. \quad (1.126)$$

In summary, the parameter-shift rule and finite difference depend on the shift value h alone, with an optimum at $\pi/(2\Omega)$ and some implicit h_{cent}^* , respectively, whereas the optimal two-term recipe additionally depends on ρ via its coefficient, but again has its optimum at the constant shift $h_{\text{opt}}^* = \pi/(2\Omega)$. This dependence on ρ , which is given by the Fourier coefficients of E , prevents us from using the optimal recipe in practice, unless an estimate for ρ is available.

In addition to the two-term recipes above, we consider the generalized central difference introduced in Sec. 1.2.2

$$\partial_{\text{FD},p} E(0) = \sum_{\mu=1}^p \frac{\alpha_{\mu}}{h} (E(\mu h) - E(-\mu h)) \text{ with } \alpha_{\mu} = \frac{(-1)^{\mu+1} \binom{2p}{p-\mu}}{\mu \binom{2p}{p}}, \quad (1.127)$$

which has the average MSE

$$\varepsilon_{\text{FD},p}^2 = \frac{\sigma^2}{s h^2} \left(H_p^2 + \frac{s}{\rho} \left(\sum_{\mu=1}^p 2\alpha_{\mu} (\sin(\mu\Omega h) - \mu\Omega h) \right)^2 \right). \quad (1.128)$$

For this we combined Eq. (1.44) with Eq. (1.51) for $R = 1$ and the practical shot allocation $s_{\mu} = s|\alpha_{\mu}|/\|\alpha\|_1$ and used the previously computed average σ^2 and squared bias in Eqs. 1.116 and 1.122. We stress once again that for all two-term shifts, and the finite difference stencils, we assumed the practical shot allocation, which is proportional to the known coefficients $|y_1|$ and $\{|\alpha_{\mu}|\}_{\mu=1}^p$, respectively. Note that this only is the same as the optimal allocation (on average) for the parameter-shift rule.

In the comparison of the estimators, the fraction $\rho/s = \sigma^2/(s b_1^2)$ will play an important role, as it determines the relation between the variance and the bias contributions to the MSE of all shift-based derivative estimators. We observe some limiting behaviour in this parameter, using the fixed optimal shift $h_{\text{PS}}^* = \pi/(2\Omega)$ of the parameter-shift rule:

$$\frac{\varepsilon_{\text{PS}}^2}{\varepsilon_{\text{cent}}^2} \xrightarrow{\rho/s \rightarrow \infty} (h_{\text{cent}} \Omega)^2, \quad \frac{\varepsilon_{\text{PS}}^2}{\varepsilon_{\text{cent}}^2} \xrightarrow{\rho/s \rightarrow 0} \left(1 - \frac{\sin(h_{\text{cent}} \Omega)}{h_{\text{cent}} \Omega} \right)^{-2} > 1. \quad (1.129)$$

As we can see, the central difference becomes favourable compared to the parameter-shift rule for $h > 1/\Omega$ in the limit of a large variance (including the factor $1/s$) but has a strictly larger MSE in the limit of a small variance. The crossing point between the best parameter-shift rule and the best central difference does not have a closed-form expression, but can

be located numerically at

$$h_{\text{cent}}^* \approx \frac{1.242}{\Omega}, \quad s_{\times} \approx 6.211\rho. \quad (1.130)$$

While the methods have the same precision at this configuration, they do not have the same shift parameter or coefficient y_1 . As also discussed in [56], the best parameter-shift rule equals the best two-term recipe up to the factor $(1 + \rho/s)^{-1}$ in the coefficient, which approaches 0 and 1 in the limits of large and small variances, respectively. As for the comparison of the central difference and the optimal two-term recipe (with optimal shifts $h_{\text{opt}}^* = \pi/(2\Omega)$ and $h_{\text{cent}}^* = x/\Omega$ satisfying Eq. (1.126)) we obtain

$$\frac{\varepsilon_{\text{cent}}^2}{\varepsilon_{\text{opt}}^2} = \frac{1}{x^2} \left[1 + (x - \sin(x)) \left(x - x \cos(x) + \frac{1}{\sin(x) - x \cos(x)} \right) \right], \quad (1.131)$$

which has a unique minimum at $(\pi/2, 1)$, i.e. the two rules coincide if $h_{\text{cent}} = h_{\text{opt}}^* = \pi/(2\Omega)$ and $s_u = \frac{2\rho}{\pi-2}$.

Furthermore we find, as anticipated on p. 16, that higher-order finite differences have a strictly larger MSE than the central difference in the low shot (i.e. high variance) regime. This is because the higher-order stencils are linear combinations of the central difference at different shifts $\{\mu h\}_{\mu=1}^p$, including h itself, which prevents allocating all shots to the optimal shift. Instead, the stencils fuse in contributions of the central difference at suboptimal h . Even worse, the variance is increased further because $\|\alpha_p\|_1 = H_p$, creating a large contribution to the MSE while the bias reduction is irrelevant in the low-shot regime:

$$\frac{\varepsilon_{\text{FD},p}^2}{\varepsilon_{\text{cent}}^2} \xrightarrow{\rho/s \rightarrow \infty} H_p^2 \underset{p>1}{>} 1. \quad (1.132)$$

Finally, the MSE of the LCU-based estimator is (see Sec. 1.2.5)

$$\varepsilon_{\text{LCU}}^2 = \frac{\Omega^2 \sigma^2}{s} \left(1 + \frac{a_0^2}{\sigma^2} \right), \quad (1.133)$$

which is larger or equal to the (minimal) parameter-shift MSE. We include it in the following analysis and find that the MSE nonetheless is not necessarily a reason to discard this estimator²¹.

Numerical experiment Here I numerically investigate the hardware-compatible derivative estimators discussed above. The performance of the various estimators in general may depend on the details of the PQC-based objective function, including the circuit ansatz C , the problem Hamiltonian H , the parameter position θ , and the measurement strategy. In the present experiment, we choose a generic combination of circuit ansatz and Hamilto-

²¹ While other aspects like the additional hardware requirements might be very good reasons.

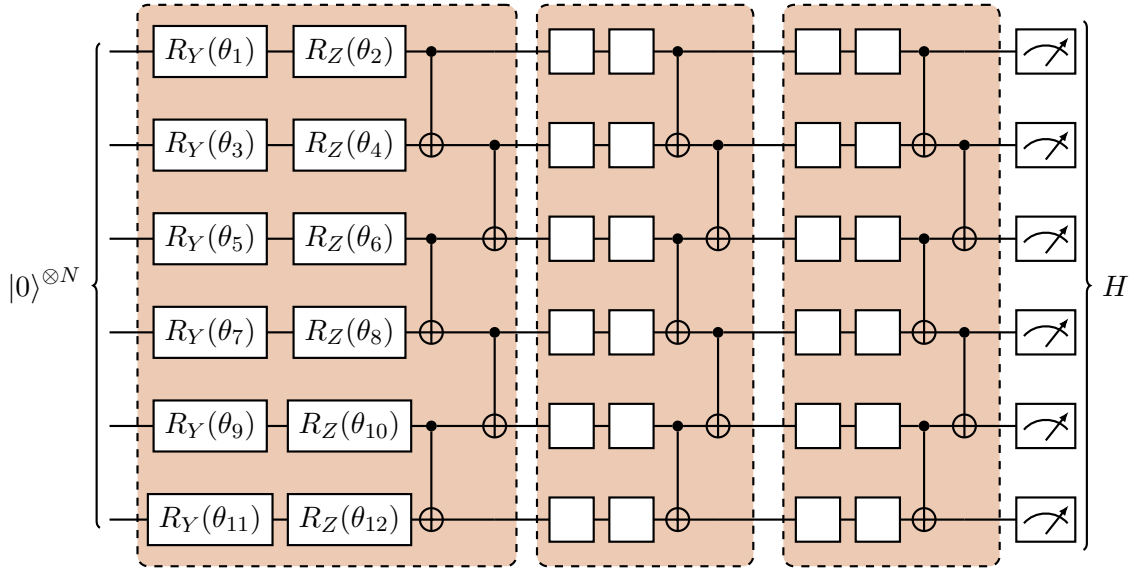


Figure 1.3: Quantum circuit used in the numerical experiment, for $N = 6$. The block of gates marked with the dashed line is repeated $N/2$ times with independent parameters in the single-qubit gates, and the Hamiltonian H measured at the end is sampled randomly as described in the main text.

nian, average across a number of random circuit parameter positions and treat the measurement setup as a black box. In addition, we choose circuits that allow for a particularly convenient treatment using the analyses in the previous sections, namely circuits with Pauli rotations as parametrized gates that lead to univariate objective functions E_k with a single frequency. I do *not* aim for a statistically exhaustive study for the particular circuit and Hamiltonian but consider this numerical experiment a demonstration of the results above which illustrates the situation for a practical example.

We start by constructing a PQC template, depicted in Fig. 1.3, which has $n = N^2$ parameters for N qubits and will lead to single-frequency objective functions with $\Omega = 1$ in each parameter. In addition we sample an observable

$$H = \sum_{i=1}^K h_i P_i, \quad (1.134)$$

with K random Pauli words P_i and coefficients $h_i \in [0, 1]$ as well as L random positions $\theta \in [-\pi, \pi]^n$. All used distributions for these sampling processes are uniform and we note that this does not necessarily represent the distributions encountered in applications. Random sampling of initial parameters, for example, makes VQAs prone to the vanishing gradient problem, also called barren plateaus [89], and Hamiltonians in applications usually possess more structure than reflected in the random samples used here. After sampling all required quantities as described, we compute the Fourier coefficients of $E_k(x)$ and $\langle H^2 \rangle_k(x)$ via a numerically exact state vector simulation using PennyLane [80]. This

is done for each sampled parameter position $\boldsymbol{\theta}$ and each parameter index k , and we obtain nL sets of Fourier coefficients $\mathcal{F}(\boldsymbol{\theta}, k) = \{a_0, a_1, b_1, \tilde{a}_0, \tilde{a}_1, \tilde{b}_1\}(\boldsymbol{\theta}, k)$. In the following we will consider the average MSE over these nL coefficient sets as the objective by which we evaluate the estimators:

$$\Delta_{[\cdot]}^2 := \sum_{j=1}^L \sum_{k=1}^n \varepsilon^2 \left[\partial_{[\cdot]} \hat{E}(\boldsymbol{\theta}_j + x \mathbf{e}_k) \Big|_{x=0} \right]. \quad (1.135)$$

This means we discard information about the parameter values $\boldsymbol{\theta}$ and the parameter index k that determines e.g. the position of the respective gate in the circuit. For applications, one could consider parameters that differ, say, by the position of the associated gate in the circuit separately. To obtain the average MSEs, we compute the average variance, squared derivative²² and squared offset

$$\sigma_{\circ}^2 := \sum_{j,k} \left[\tilde{a}_0(\boldsymbol{\theta}_j, k) - a_0(\boldsymbol{\theta}_j, k) \right]^2 - \frac{1}{2} \left(a_1(\boldsymbol{\theta}_j, k)^2 + b_1(\boldsymbol{\theta}_j, k)^2 \right), \quad (1.136)$$

$$E'_{\circ}{}^2 := \frac{1}{2} \sum_{j,k} a_1(\boldsymbol{\theta}_j, k)^2 + b_1(\boldsymbol{\theta}_j, k)^2, \quad (1.137)$$

$$a_{\circ}^2 := \sum_{j,k} a_0(\boldsymbol{\theta}_j, k)^2. \quad (1.138)$$

Recall that the optimal two-term recipe depends on E via the fraction ρ , which leads to an individual optimal coefficient $y_{1\text{opt}}^*(h, \boldsymbol{\theta}, k)$ for each of the parameter positions and indices. In practice it is likely that we want to minimize the MSE on average and that an estimate for the constituents of ρ is given on average as well. Therefore it is useful to consider the two-term recipe that minimizes the objective Δ^2 , i.e. the average MSE, with a fixed $y_1^{\circ}(h) \neq y_1^{\circ}(\boldsymbol{\theta}, k)$ that can be computed from the fraction of averages $\rho_{\circ} := \sigma_{\circ}^2 / E'_{\circ}{}^2$. Note that $\sum_{\boldsymbol{\theta}, k} y_1^*(h, \boldsymbol{\theta}, k) \neq y_1^{\circ}(h)$ due to the involved nonlinear dependencies. However, in the described numerical experiment, the achieved Δ^2 of the individual $y_1^*(h, \boldsymbol{\theta}, k)$ and of y_1° are very similar. This is an important insight, because it tells us that choosing one $y_1^{\circ}(h)$ based on the average data $(\sigma_{\circ}^2, E'_{\circ}{}^2)$ for a circuit is almost as good as choosing $y_1^*(h, \boldsymbol{\theta}, k)$ for each specific parameter k and position $\boldsymbol{\theta}$. The former not only yields a much simpler, static, recipe. In addition, the required Fourier information for the latter is not available in practice, whereas estimators for the average σ_{\circ} and $E'_{\circ}{}^2$ might be given. Thus, we will skip the impractical recipe with y_1^* and only include the one with y_1° in the following analysis.

From here on, the average data across the $\boldsymbol{\theta}$ and k will be all the information we need about the investigated circuit, and we will only make use of σ_{\circ}^2 , $E'_{\circ}{}^2$ and a_{\circ}^2 . Except for the LCU-based derivative all estimators have a shift parameter h , which we sweep across the range $(0, \pi)$. At each combination of (h, y_1, s) we calculate the MSE for each method.

²² As discussed above, we will have $a_1^2 = b_1^2$ on average across the parameter space, therefore we compute the average derivative using both a_1 and b_1 .

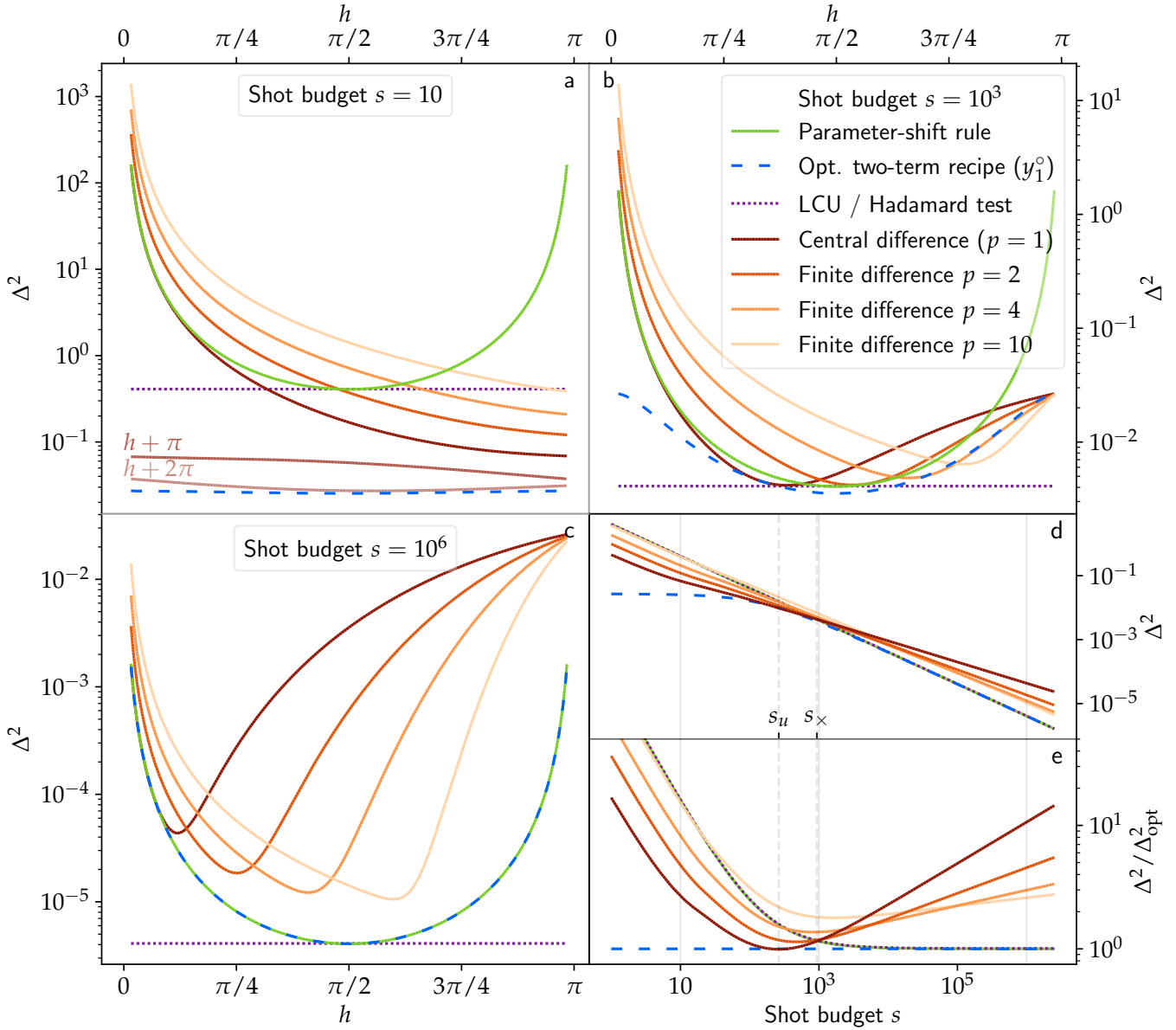


Figure 1.4: mean squared errors Δ^2 for the discussed derivative estimators (different colors) applied to the selected circuit for $N = 6$ qubits, averaged over $L = 20$ parameter positions and the $n = 36$ parameters. The first three panels show the dependence of the MSEs on the shift parameter h for a low (a), medium (b) and large (c) shot budget s . Δ_{LCU}^2 does not have a shift parameter and accordingly is depicted as a constant. In (a) the central difference with shifts increased by π and 2π are depicted as shaded red lines. The lower right panels show the dependence of the MSEs on the shot budget s after minimizing over $h \in (0, \pi)$, for each recipe and shot number independently. Panel (d) shows the absolute errors whereas (e) shows the relation to the MSE of the optimal two-term recipe, for which we use the coefficient y_1° as described in the main text. The solid vertical lines indicate the shot numbers of the first three panels, the dashed vertical lines mark the shot budget s_u at which the central difference is the best two-term recipe and s_x for which $\Delta_{\text{PS}}^2 = \Delta_{\text{cent}}^2$.

Above we discussed the role of ρ for the analysis of the estimators. The variance and bias contribution to the average MSEs Δ^2 are linear in σ_\circ^2 and $E_\circ'^2$, respectively. By dividing the averages we computed above, we obtain an estimate for ρ that is compatible with the objective $\Delta_{[\cdot]}^2$ and can be used to obtain y° , as well as the constant that determines the MSE of the LCU-based estimator:

$$\rho_\circ = \frac{\sigma_\circ^2}{E_\circ'^2} \approx \begin{cases} 32 & N = 4 \\ 150 & N = 6 \\ 509 & N = 8 \\ 2514 & N = 10 \end{cases} \quad \frac{a_\circ^2}{\sigma_\circ^2} \approx \begin{cases} 5.3 \times 10^{-2} & N = 4 \\ 7.4 \times 10^{-3} & N = 6 \\ 2.9 \times 10^{-3} & N = 8 \\ 4.1 \times 10^{-4} & N = 10. \end{cases} \quad (1.139)$$

These values tell us that the bias plays a much smaller role than the variance for a significant range of shot budgets $s_\times < 6.211\rho_\circ$, and we conclude that estimators which minimize the variance perform better (in terms of the MSE) in this regime than those that are unbiased at the cost of a larger variance. This is illustrated by the numerical results for Δ^2 I show in the following.

In Fig. 1.4 I show the MSEs for selected s and all h (a-c), as well as for the full range of shots at the respective minimizing shifts h^* (d, e). The shows results are for $N = 6$ qubits, $K = 12$ terms in the Hamiltonian, and $L = 20$ parameter positions to average over, and the selected shot budgets for (a-c) are $s \in \{10, 10^3, 10^6\}$. For small $s = 10$ ($\rho/s \gg 1$, Fig. 1.4a), the central difference has a lower MSE than the parameter-shift rule for all shifts h , which matches our expectation based on the analysis above (in particular Eq. (1.129)) and the values we obtained for ρ . Similarly, we confirm that higher-order stencils have a strictly increased MSE over the central difference and that the optimal shift value for all finite differences lies outside of the considered range $(0, \pi)$, as indicated for the central difference by the shaded red lines. The optimal two-term recipe with $y_1 = y_1^\circ(h)$ performs orders of magnitude better than all other methods. As shown in Fig. 1.5, the numerically optimized coefficient y_1° attains a very small value in the few-shot regime, suggesting to estimate an almost vanishing derivative. This estimator may be impractical and might not be useful for our purposes, but this ultimately depends on whether Δ^2 is the only relevant criterion to rate the estimator. All finite difference recipes reproduce this solution for small s if we send $h \rightarrow \infty$, which we did not allow in our analysis. In particular one could consider the central difference with shifts $h_j = h_0 + j\pi$, which produces estimators with decreasing Δ^2 for increasing j until it reaches h_{cent}^* . In this case the implemented function evaluations on the QPU do not change due to the periodicity of E , but only the prefactor y_1 is modified, converging towards the optimal coefficient. This means that instead of looking at $h > \pi$ we may just reduce the coefficient in classical postprocessing.

For medium $s = 10^3$ ($\rho/s \approx 1$, Fig. 1.4b) we see that all tested derivative estimators achieve similar minimal errors, at different shifts h , but also that the parameter-shift rule becomes favourable over the central difference and any other tested finite difference sten-

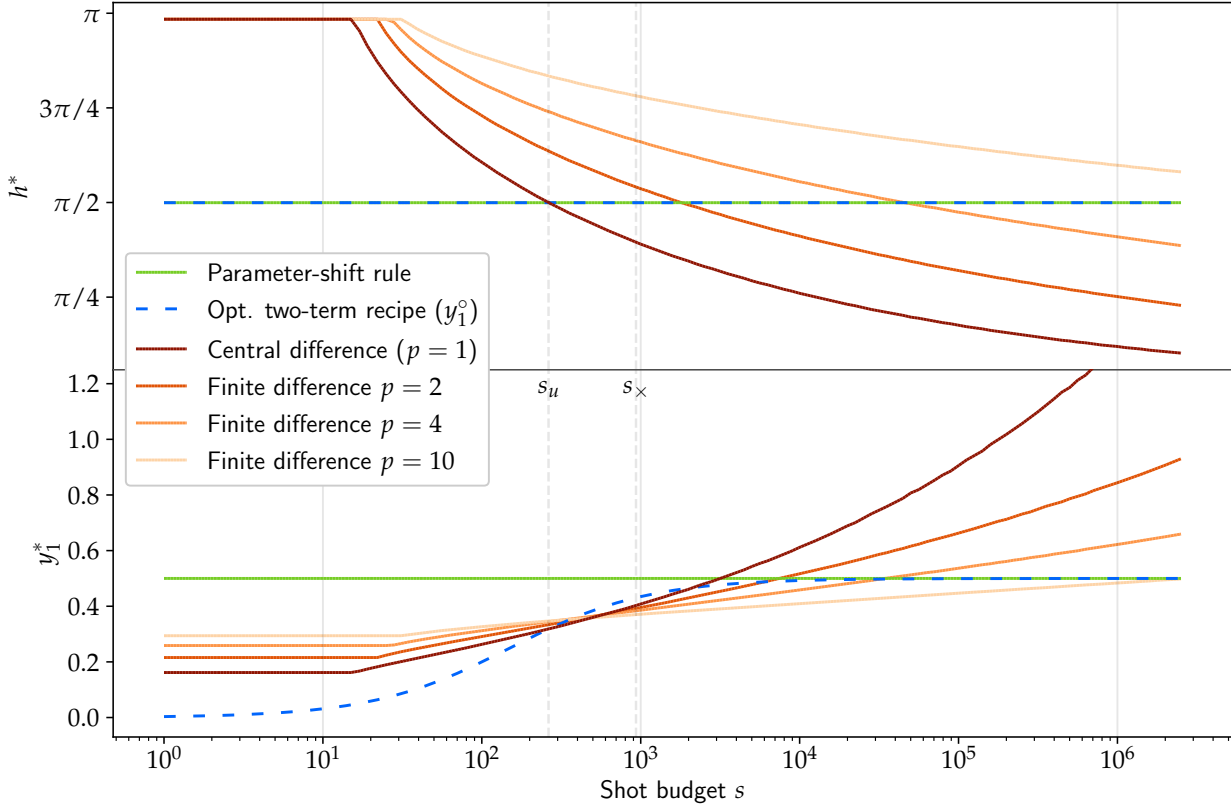


Figure 1.5: Optimal parameters for the derivative estimators MSEs presented in Fig. 1.4, which are used to obtain the best Δ^2 per recipe and per shot budget s shown in Fig. 1.4(d). The optimal shift h^* is $\pi/2$ for all s , as is also apparent from Eq. (1.124), whereas the optimal choice for finite difference methods varies with the shot budget due to the relationship between y_1 and h . The optimal recipe approaches the parameter-shift rule for large s but for small shot budgets it is better to rescale its coefficient. The finite difference methods are truncated artificially in h^* and y_1^* for small shots because we only considered $h \in (0, \pi)$, as discussed in the main text. Again the solid vertical lines indicate the s from Fig. 1.4(a-c) and the dashed vertical lines mark s_u , where the optimal two-term recipe and the central difference are the same, and s_x for which $\Delta_{\text{PS}}^2 = \Delta_{\text{cent}}^2$ but h and y_1 differ between the two methods. Furthermore, we remark that the higher-order stencils have more coefficients than just y_1 and that the LCU-based estimator is missing here as it does not have parameters.

cil. This means that this shot budget is very close to s_\times . Importantly, the parameter-shift rule (at its optimal shift size $h_{\text{PS}}^* = \pi/(2\Omega)$) already performs close to optimal out of all antisymmetric two-term recipes in this regime, i.e. $y_1^o(h_{\text{opt}}^*) \approx 1/2$.

For large $s = 10^6$ ($\rho/s \ll 1$, Fig. 1.4c), the bias dominates the MSE of all finite difference stencils, and the higher-order stencils gain performance by reducing it at the cost of an increased variance. Both the best achieved Δ^2 and the optimal h depend (for the shown data) monotonously on p . As expected, the parameter-shift rule becomes the optimal two-term differentiation strategy in this shot number regime, as it is unbiased and the variance is strongly suppressed by s .

Panel (d) shows the dependency of $\Delta_{[\cdot]}^2(h_{[\cdot]}^*)$ at its respective minimizing shift on the number of shots s for each method. This is complemented by panel (e), which displays the overhead factor $\Delta_{[\cdot]}^2(h_{[\cdot]}^*)/\Delta_{\text{opt}}^2(h_{\text{opt}}^*)$ with respect to the optimal two-term recipe. At low shot budgets the performance of all finite-difference methods is artificially reduced because we restricted the minimization to $h \in (0, \pi)$. The unique shot budget for which the central difference is the same as the optimal recipe is $s_u \approx 263$ for this circuit, which matches²³ the prediction $\frac{2\rho_o}{\pi-2} \approx 263$. Furthermore, we observe that the parameter-shift rule exhibits a better scaling with s than the central difference at h_{cent}^* and becomes favourable at $s_\times \approx 934 \approx 6.211\rho_o$, indeed close to the situation shown in panel (b). For $s \rightarrow \infty$ the parameter-shift rule approaches the optimal two-term recipe as expected. The scaling of $\Delta_{\text{FD},p}^2$ with s is improved by increasing the order $2p$, so that it approaches the asymptotic behaviour of the parameter-shift rule. For the given data this only pays off above s_\times so that there is no regime in which a generalized finite difference is favourable.

So far we have not looked at the LCU-based derivative estimator, which is shown as a constant line in Fig. 1.4(a-c) because it does not have a shift parameter. The MSE is the same as for the parameter-shift rule at h_{PS}^* , up to the factor $1 + a_0^2/\sigma^2$. As we computed above, a_0^2 turned out to be much smaller than σ_o^2 for the investigated circuit, so that the MSEs become approximately equal and all statements for the parameter-shift rule at h_{PS}^* can be made for this estimator as well, up to the prefactor slightly larger than 1.

Based on this numerical experiment, we conclude the following: first that higher order stencils are not favourable in any shot number regime, but considering antisymmetric two-term rules is sufficient within our scope²⁴. Second, for low shot numbers the central difference has significantly better properties than the parameter-shift rule, but for large shot numbers the latter becomes increasingly better and is asymptotically equal to the optimal two-term recipe. Third, in the regime of low shots the optimal two-term recipe becomes a somewhat pathological solution to the minimization problem and does not necessarily yield a useful estimator, depending on the use case. In any case, the optimal shift for this estimator is $h_{\text{opt}}^* = \pi/(2\Omega)$, so that the best estimator is given by executing

²³ Note that this match between prediction and data does not testify the validity of any assumptions we made, but rather of the performed calculations.

²⁴ This of course only covers the investigated derivative estimators and does not imply a statement about other possible estimators.

the parameter-shift rule and multiplying with the prefactor $(1 + \rho/s)^{-1}$ as also discussed in [56]. In this regime the central difference matches the optimal two-term recipe in performance for some rather large $h > \pi$. Finally, the additional MSE of the LCU_s-based estimator, as compared to the parameter-shift rule, is only marginal for the investigated circuit, but given that it requires additional qubits, connectivity and compiling it seems reasonable to not choose this estimator in common use cases.

In practice, these insights lead to the following guidelines for gradient estimation, assuming that no adaptive methods are employed and respecting the limitations discussed further below: If the average Fourier coefficients of E , over a relevant ensemble of parameter positions and indices, are not known, the central difference and the original parameter-shift rule should be used for small and large shot budgets, respectively. Determining the exact transition point again requires an estimate for the average Fourier coefficients. If those are known, the best gradient estimator is the optimal antisymmetric two-term recipe with

$$h_{\text{opt}} = \frac{\pi}{2\Omega}, \quad y_1^\circ = \frac{\Omega}{2 + 2\rho_0/s}. \quad (1.140)$$

The number of shots is not the only relevant quantity when assessing the cost of the derivative estimators. In practice a time cost model is needed that respects all levels of the used quantum hardware stack. In addition to the number of shots one can expect the number of unique circuits, i.e. unique parameter settings, to have a sizeable impact on most hardware architectures even when classical communication, queueing and compiling are reduced to an $\mathcal{O}(1)$ overhead when computing a gradient. As we mostly found gradient estimators based on two shifted evaluations to be of interest and the overhead due to circuit configurations will be equal for all these estimators, we will not pursue this discussion further, but note that it should be relevant when comparing derivative estimators in other scenarios e.g. the generalized parameter-shift rule for larger R and the central difference.

Regarding the scope of the presented analysis, there is a number of limitations: first, we selected a few commonly used gradient estimators, excluding other approaches that also are popular in practice, like SPSA (see Sec. 1.2.6). Second, we only considered the estimation of the gradient entries regarding the MSE. For applications it might be interesting to look at other quantities like the quantum natural gradient [90] or the (approximate) Hessian [91, 92, 93, 94], and to consider other error metrics, like the gradient *direction* rather than its elementwise precision. An investigation building on [57] that includes more general estimators, estimated quantities and error metrics could be performed to address these first two limitations. Third, the numerical data I used stems from a specific, somewhat arbitrary circuit structure leading to single-frequency objective functions. Moreover, I used parameter positions sampled uniformly at random, which likely is not a representative distribution of positions seen during an optimization workflow. Extending the analysis to

larger Fourier spectra requires some work but should be possible in a direct generalization of the analysis. Similarly, there should be no conceptual hurdle to consider the Fourier coefficients witnessed during an optimization of interest and comparing them to the data from positions sampled uniformly at random. Lastly, we did not consider device noise in the analysis, which e.g. may be addressed by combining the analysis with the ideas presented in [35].

The computer programs for the presented analysis and figures can be found in [95].

Chapter 2

Parameter-shift rules

This chapter contains a literature discussion regarding parameter-shift rules, an appendix I contributed to Ref. [13] and the first complete publication, which treats generalizations of the parameter-shift rule.

2.1 Literature discussion

The parameter-shift rule has been the topic of various research works, some of which appeared simultaneously (at the time scale of research projects) and others of which seem to have taken more time to become known in the community. This led to duplications and rediscoveries and I would like to give the corresponding references and a brief review of the literature.

The first mention of the original rule for Pauli rotation gates, i.e. gates generated by a Pauli word rescaled by $\frac{1}{2}$, to the best of my knowledge is in [62], in the context of quantum optimal control. Shortly after that it was transferred to expectation values of parametrized quantum circuits (PQCs) in [96], and [97] applied it to probability distributions created by PQCs. An extension to arbitrary operators with eigenvalues $\pm\frac{\Omega}{2}$ together with parameter-shift rules for continuous-variable architectures was presented in [98].

All of the above works used the operator identity

$$i[G, \rho] = U(\pi/4)\rho U^\dagger(\pi/4) - U(-\pi/4)\rho U^\dagger(-\pi/4) \quad (2.1)$$

for the derivation of the parameter-shift rule for the case $\Omega = 1$, as resulting from the standard Pauli rotation gates $\exp(ixP/2)$. The first work considering the structure of the PQC-based objective function itself was [25]. This made it possible to use the large established toolbox of Fourier transforms and signal processing, and in [25] the authors derived a parameter-shift rule for all gates whose generator has eigenvalues Ωk_i for natural numbers k_i , which is based on the discrete Fourier transform (DFT). For R frequencies in the cost function, this leads to $2R + 1$ evaluations, but for $R \in \{1, 2\}$ the authors improve

the rule to $2R$ evaluations. This reproduces the original shift rule, but using an arbitrary shift $\pm h$, and is the first derivation of the four-term parameter-shift rule for gates with 3 unique eigenvalues. This is complemented by the later suggestion of [99] that proposes a one-rule-fits-all approach that decomposes two-qubit gates and arrives at 30 shifted terms to differentiate such a gate. The approach by [25] requires at most 13 shifted evaluations¹, even if we do not assume any structure on the eigenvalues of G .

A new direction was explored by [36], which considers perturbed gates of the form $U(x) = \exp(i(xG + F))$ with $[G, F] \neq 0$ and derives a stochastic parameter-shift rule that produces the correct derivative on expectation over a (classical) random distribution of shift values, assuming access to $\exp(i\pi G/4)$. Without this access, the authors show that the derivative can be approximated using a parametrized version $\exp(it(xG + F))$ of the original gate. The generalization to use arbitrary shift values was rediscovered in [56], an article that also is discussed in more detail at the beginning of Sec. 1.2. Furthermore, the authors investigate the extension of parameter-shift rules to higher-order derivatives of PQC-based functions, including the Hessian and the metric tensor, which can be interpreted as the Hessian of an auxiliary function function. The Hessian also is considered (with fixed shift values) in [100].

A parameter-shift rule for gates used in unitary coupled cluster (UCC) PQCs for quantum chemistry was developed in [101]. It involves four unique circuit evaluations that do not only use shifted parameters but also insert modified gates, which can be constructed under reasonable assumptions in the application setting of quantum chemistry. If the quantum state prepared by the UCC circuit is real-valued², the rule can be reduced to two modified circuit executions instead. In App. F of [13], which also is included in Sec. 2.2 of this thesis, I derived a four-term parameter-shift rule for gates with three unique eigenvalues using a operator-based approach similar to Eq. (2.1). This reproduced the four-term rule from [25], which was unknown to me at the time. A further analysis of univariate shift rules, in particular with comparison to finite differences and for Hessians, was provided in [102] (also see Sec. 1.2).

In the publication included in this chapter [14] we considered a similar setting as in [25] but also looked at more general frequency spectra, higher-order single-parameter and multivariate derivatives and performed a thorough cost analysis. Our focus in the main text are cost functions with equidistant Fourier spectra and we show that the generalized shift rule can be combined straightforwardly with the stochastic shift rule from [36]. At the same time the article [103] appeared, also providing generalized parameter-shift rules but with a focus on the operator decomposition level and using the linearity of the derivative in the generator G^3 . The resulting differentiation technique seems to be complementary to our shift rules from [14] in the sense that it offers derivative estimation at reasonable cost

¹ At most 4 (unique) eigenvalues, at most 6 frequencies and hence at most 13 shifts.

² Meaning that a global phase can be chosen such that all entries of the state vector are real-valued.

³ Of course the generator does not only appear in the commutator in the derivative but also in the circuit itself. The latter occurrence is ignored here.

for quantum gates for which our technique is particularly costly. Furthermore, Ref. [26] appeared online at the same time as well, deriving essentially the same first-order shift rules, but focusing on the generality of the Fourier transform approach and on two-qubit and qutrit examples and not on shift rules for equidistant frequency spectra, for which the coefficients can be computed explicitly.

In [27] the author set up a rigorous mathematical framework for parameter-shift rules based on Borel measures, enabling an existence proof for arbitrary generator spectra, an optimality proof for the corresponding shift rules, including those for equidistant spectra from the attached publication, and leading to convex optimization approaches to find these optimal shift rules. Following up on this, [37] extends the framework to treat perturbed gates $\exp(i(xG + F))$ as in [36] but without requiring a changed circuit structure, including a truncation scheme that allows for practical implementation of approximate shift rules for perturbed gates. However, it is also proven that the largest required shift will scale polynomially in the desired accuracy, posing challenges on quantum hardware regarding control precision, gate durations and coherence times.

This discussion shows that parameter-shift rules have drawn a lot of attention in the last five years, and that significant progress has been made in setting up a general mathematical framework, deriving explicit shift rules and developing heuristics to implement derivative estimators in a hardware-friendly way. At the same time, the numerous publications and preprints on shift rules and the field of near-term quantum computing more generally has made it difficult for authors to keep an overview over recent works. A particular example is Ref. [25] which contains a very clean and simple derivation of rather general parameter-shift rules but unfortunately only became known in the community later on⁴. This work also suggests to use coordinate descent based on a reconstruction of the Fourier series E for optimization in VQAs, which then was rediscovered at least three more times as *sequential minimal optimization* [29], *Jacobi-1* [28] and *Rotosolve* [30], respectively, and turned out to be a good optimization method for various PQC-based functions [29, 30, 104, 105], though not for all [106].

⁴ This statement is based on citations and personal correspondence with researchers in the field.

2.2 Four-term shift rule for quantum chemistry gates

As mentioned above, I developed a four-term parameter-shift rule in the specific context of quantum chemistry gates before working on the more general rules presented in the publication attached to this chapter. This rule was published as Appendix F of [13], which is presented in the following section in a slightly modified version. Both content and presentation are my own work (up to comments on the text by my coauthors). I believe that it can be useful and encouraging to see this partial progress before discovering a broader framework. In addition it may show the limitations of the perspective taken in the following section.

2.2.1 Introduction to the four-term rule

In order to compute the derivative of expectation values with respect to quantum gate parameters, the so-called parameter-shift rule has been established as a tool to avoid finite difference derivatives, which become unstable under the influence of noise from both, measurements and circuit imperfections [62]. In addition to the original concept, multiple efforts have been made to analyse and generalize the parameter-shift rule [36, 56, 96, 98, 107]. In this section we introduce the concept of tuning the shift angle in parameter-shift rules for a minor algorithmic advantage (Sec. 2.2.2), a new four-term parameter-shift rule for gates with three distinct eigenvalues (Sec. 2.2.3) and exclude a further generalization of this type of parameter-shift rules based on the perspective taken here (Sec. 29). We also compare our new four-term rule to the one recently presented in [101] (Sec. 29) and extend the variance minimization strategy from [56] to both four-term rules. This four term shift rule is applicable to all of the quantum number-preserving gates introduced in [13], except for the spin adapted QNP_{OR} gate, which can be analytically differentiated by applying shift rules to the Givens rotations and using the chain rule.

2.2.2 Two-term rule and shift tuning

We briefly recap the derivation of the standard parameter-shift rule without fixing the shift angle, leading to a free parameter in the rule. Consider a parametrized gate of the form

$$U(x) = \exp\left(i\frac{x}{2}P\right), \quad (2.2)$$

where $P^2 = \mathbb{I}$, as is the case e.g. for Pauli rotation gates. In a circuit with an arbitrary number of parameters, let's single out the parameter of the gate U above and write our cost function of interest as

$$E(x) = \langle \psi(x) | H | \psi(x) \rangle =: \langle \phi | U(x)^\dagger B U(x) | \phi \rangle, \quad (2.3)$$

where the part of the PQC before the gate U has been absorbed into $|\phi\rangle$ and the part after U is absorbed in B . Then the derivative is, by the product rule, given by

$$\frac{\partial}{\partial x} E(x) = \langle \phi | U(x)^\dagger \left(\frac{i}{2} [B, P] \right) U(x) | \phi \rangle. \quad (2.4)$$

Now look at the conjugation of B by U at arbitrary shift angles $\pm x_1$:

$$\mathcal{U}(\pm x_1)(B) := U(\pm x_1)^\dagger B U(\pm x_1) \quad (2.5)$$

$$= U(\pm x_1)^\dagger B \left(\cos\left(\frac{x_1}{2}\right) \mathbb{I} \pm i \sin\left(\frac{x_1}{2}\right) P \right) \quad (2.6)$$

$$= \cos\left(\frac{x_1}{2}\right)^2 B + \sin\left(\frac{x_1}{2}\right)^2 P B P \pm \frac{i}{2} \sin(x_1) [B, P]. \quad (2.7)$$

Subtracting $\mathcal{U}(-x_1)(B)$ from $\mathcal{U}(x_1)(B)$ and excluding multiples of π as values for x_1 , we obtain the generalized two-term parameter-shift rule

$$\mathcal{U}(x_1)(B) - \mathcal{U}(-x_1)(B) = i \sin(x_1) [B, P] \quad (2.8)$$

$$\Rightarrow \frac{\partial E}{\partial x}(x) = \frac{1}{2 \sin(x_1)} (E(x + x_1) - E(x - x_1)), \quad (2.9)$$

where the original parameter-shift rule corresponds to choosing $x_1 = \pi/2$. We note that the concept of shift tuning was independently discovered in [56] and introduced in the quantum computing software package PennyLane [80].

Reducing the gate count In particular, the general form of Eq. (2.8) allows us – provided that x is not a multiple of π – to choose $x_1 = -x$, making the first of the cost function evaluations $E(0)$ and therefore reducing the gate count because $U(0) = \mathbb{I}$ can be skipped in the circuit. This may lead to an additional gate count reduction if the neighbouring gates on both sides of U can be merged, which is true e.g. in circuits for the quantum approximate optimization algorithm (QAOA).

2.2.3 Four-term parameter-shift rule

Here we derive a four-term parameter-shift rule for gates that do not fulfil the two-term rule, e.g. controlled Pauli rotation gates like $c\text{-}R_Z(x)$ or many of the quantum number-preserving gates with one parameter in [13]. Consider a gate

$$U(x) = \exp\left(i \frac{x}{2} Q\right) \quad (2.10)$$

with $Q^3 = Q$, as is true for any operator that has spectrum $\{-1, 0, 1\}$, but not necessarily with $Q^2 = \mathbb{I}$. Then the exponential series of the gate can be rewritten as

$$U(x) = \mathbb{I} + \left(\cos\left(\frac{x}{2}\right) - 1 \right) Q^2 + i \sin\left(\frac{x}{2}\right) Q, \quad (2.11)$$

and a computation similar to the one above leads to

$$\mathcal{U}(x_\mu)(B) - \mathcal{U}(-x_\mu)(B) = 2i \sin\left(\frac{x_\mu}{2}\right) \left[[B, Q] + \left(\cos\left(\frac{x_\mu}{2}\right) - 1\right) [Q, QBQ] \right]. \quad (2.12)$$

We can then obtain the commutator by linearly combining this difference with itself for two angles x_1 and x_2 , so that

$$\frac{i}{2}[B, Q] = (y_1 (\mathcal{U}(x_1) - \mathcal{U}(-x_1)) + y_2 (\mathcal{U}(x_2) - \mathcal{U}(-x_2)))(B), \quad (2.13)$$

which holds if the angles x_μ and the prefactors y_μ satisfy

$$\frac{1}{4} = y_1 \sin\left(\frac{x_1}{2}\right) + y_2 \sin\left(\frac{x_2}{2}\right) \quad (2.14)$$

$$\frac{1}{2} = y_1 \sin(x_1) + y_2 \sin(x_2). \quad (2.15)$$

Therefore, we get the four-term parameter-shift rule

$$\frac{\partial E}{\partial x}(x) = y_1 (E(x + x_1) - E(x - x_1)) + y_2 (E(x + x_2) - E(x - x_2)), \quad (2.16)$$

where we again can choose x_1 or x_2 such that one of the function evaluations skips the gate U . A particularly symmetric solution of Eq. (2.14) and (2.15) is

$$y_1 = \frac{1}{2}, \quad y_2 = \frac{1 - \sqrt{2}}{4}, \quad x_1 = \frac{\pi}{2}, \quad x_2 = \pi. \quad (2.17)$$

In general, any gate for which the spectrum of the generator is $\{-a + c, c, a + c\}$ obeys the four-term parameter-shift rule as the shift c can be absorbed into a global phase that does not contribute to the gradient and a can be absorbed into the variational parameter of the gate.

As an example, the four-term rule is applicable to (multi-)controlled Pauli rotations $c\text{-}R_P(x)$ for which Q is the zero matrix except for the Pauli operator P on the target qubit. For multiple control qubits and the quantum number-preserving gates, this may lead to fewer circuit evaluations than using the chain rule and applying the two-term rule to the gate decomposition⁵

In order to find out whether an m -qubit single-parameter gate U satisfies the four-term rule, one can compute

$$Q = \left. \frac{\partial U}{\partial x}(x) \right|_{x=0}, \quad \overline{Q} := Q - \frac{1}{2^n} \text{tr}(Q), \quad (2.18)$$

and test if there is an $a \in \mathbb{R}$ such that $\overline{Q}^3 = a^2 \overline{Q}$. This is a sufficient condition, as the only property we needed for the four term rule to apply was this one of the generator spectrum.

⁵ The original appendix of [13] stated that this is always the case, which is not true.

Relation to another four-term rule

Previous work showed the existence of a four-term parameter-shift rule [101] for gates of the form in Eq. (2.10), which is implemented with only one shift angle but requires the two additional gates

$$V_{\pm} = \exp\left(\mp \frac{ix_1\pi}{4} P_0\right) \quad \text{with } P_0 = \mathbb{I} - Q^2. \quad (2.19)$$

There are four relevant aspects when comparing this rule to the one in Eq. (2.16): First, our four-term rule does not require any additional gates like V_{\pm} , which add overhead to the gradient evaluation circuits. While the authors bound the additional cost by the cost of the differentiated gate itself, it might more crucially be non-trivial to construct V_{\pm} for gates that do not have an obvious fermionic representation like the gates considered in [101].

Second, the shift tuning technique for gate count reduction in Sec. 2.2.2 can easily be extended to both our four-term rule and the rule derived in [101], provided one has access to the parametrized versions of V_{\pm} . As the construction of V_{\pm} for fermion-based gates is based on rotations, this access can be assumed for these gates whenever V_{\pm} themselves can be implemented.

Third, it was shown in [101] that their four-term rule reduces to a standard *two*-term rule up to the insertions of the V_{\pm} operators whenever both the circuit of interest and the measured observables are purely real-valued. This is the case for virtually all molecular Hamiltonians and most of the circuits proposed for quantum chemistry problems – including the circuit structures in [13] – such that gradients of highly complex gates may be computed with just two circuit executions including the gates V_{\pm} using the rule in [101].

Fourth, the variances of the derivative estimators given by the two rules can be minimized to the same value by choosing the shift angles optimally, as shown in Sec. 2.2.4. This means that for a given budget of circuit executions, the precision of the estimated derivative is the same, even though the number of unique circuits differs.

In summary, the specialized two-term parameter-shift rule in [101] is preferable if the following three criteria hold: First, the circuit and observable need to be real-valued. Second, the auxiliary gates V_{\pm} have to be available. Third, the number of *unique* circuits instead of the measurement budget must be the relevant cost metric of the computation, so that the reduction from four to two shifts provides an advantage which is larger than the overhead of adding V_{\pm} . In all other scenarios the four-term rule Eq. (2.16) with the optimal parameters in Eq. (2.31) and (2.32) requires slightly fewer gates and the same number of shots.

Impossibility of some further shift rules

One may wonder whether a three shift rule is possible for gates whose generators have just three distinct eigenvalues and whether shift rules exist for gates with more distinct

eigenvalues. We present some insights on these questions in the following.

During the derivation of the four-term parameter-shift rule we chose to first linearly combine $\mathcal{U}(\pm x_1)(B)$ and $\mathcal{U}(\pm x_2)(B)$ with the same prefactors, respectively. Alternatively one may try to combine $\mathcal{U}(x_\mu)(B)$ at three shift angles $\{x_\mu\}_\mu$ linearly and demand the result to fulfil

$$\sum_{\mu=1}^3 y_\mu \mathcal{U}(x_\mu)(B) \stackrel{!}{=} \frac{i}{2} [B, Q]. \quad (2.20)$$

This leads to the system of equations

$$0 = y_1 [c_1 - c_3] + y_2 [c_2 - c_3], \quad (2.21)$$

$$0 = y_1 [s_1^2 - s_3^2] + y_2 [s_2^2 - s_3^2], \quad (2.22)$$

$$1 = 2y_1 [s_1 - s_3] + 2y_2 [s_2 - s_3], \quad (2.23)$$

$$1 = y_1 [\sin(x_1) - \sin(x_3)] + y_2 [\sin(x_2) - \sin(x_3)], \quad (2.24)$$

with $c_\mu = \cos(\frac{x_\mu}{2})$ and $s_\mu = \sin(\frac{x_\mu}{2})$, which we conjecture to not have a solution.

Considering the generalization of the (standard) two-term shift rule to the four-term rule in Eq. (2.16) and their requirement on the gate generator, i.e. $Q^2 = \mathbb{I}$ and $Q^3 = Q$, it seems a natural question whether further generalization is possible to gates that, e.g. fulfil $Q^5 = Q$. We show next that this is not the case.

Consider the generalized condition $Q^k = Q^\ell$, $k \neq \ell$ for the generator of a d -dimensional one-parameter gate. We recall that we may absorb shifts and scaling prefactors of the spectrum of Q into a global phase gate and the variational parameter, respectively, which may be used to obtain gates satisfying the generalized condition $Q^k = Q^\ell$. In the eigenbasis of the Hermitian matrix Q , this condition becomes $\lambda_i^k = \lambda_i^\ell \forall 1 \leq i \leq d$, which only ever is solved by $-1, 0$ and 1 over \mathbb{R} (in which the spectrum of Q must be contained) with the additional condition $k - \ell \pmod{2} = 0$ for $\lambda_i = -1$. This means that Q already satisfies $Q^3 = Q$, allowing for the four-term rule to be applied.

Consequently, a direct⁶ generalization of the four-term rule is not possible. Note that this does not exclude the existence of other schemes to compute the derivative of an expectation value w.r.t. parametrized states that are based on linear combinations of shifted expectation values.

2.2.4 Minimizing the variance

If we approximate the physical variance of the expectation value to be independent of x , i.e. $\sigma^2 [\hat{E}(x)] = \sigma^2$, the variance of measuring E at a given parameter for sufficiently many measurements s is σ^2/s . The resulting variance of the two-term shift rule derivative for a

⁶ In the sense of the above attempt.

budget of s measurements is

$$\sigma_{\text{PS2}}^2 = \frac{\sigma^2}{s \sin^2(x_1)}, \quad (2.25)$$

where we chose the optimal allocation of $s/2$ measurements to each of the two terms in the shift rule⁷. We may optimize the shift angle in the two-term rule w.r.t. this variance, which yields the standard choice $\pi/2$ for the shift, because

$$\operatorname{argmin}_{x_1 \in (0, \pi)} \frac{\sigma^2}{s \sin^2(x_1)} = \frac{\pi}{2}. \quad (2.26)$$

The variance can be reduced further by introducing a multiplicative bias to the estimator, as presented in [56]; the optimal choice of the prefactor depends on the value and the variance of the derivative and is given by

$$\lambda^* = \left(1 + \frac{\sigma^2}{s(E'(x))^2} \right)^{-1}. \quad (2.27)$$

Note that λ^* has to be approximated because σ^2 and $E'(x)$ are not known exactly. The optimal choice of the shift parameter remains $\frac{\pi}{2}$.

For the four-term rule in Eq. (2.16), the optimal shot allocation⁸ is proportional to $|y_\mu|$ and leads to the variance

$$\sigma_{\text{PS4}}^2 = 4(|y_1| + |y_2|)^2 \frac{\sigma^2}{s}. \quad (2.28)$$

As for the two-term parameter-shift rule, we may minimize this variance w.r.t. x_1 and x_2 via y_1 and y_2 , which can be found to be

$$y_2 = \frac{1}{4 \sin(x_2/2)} \frac{1 - \cos(x_1/2)}{\cos(x_2/2) - \cos(x_1/2)}, \quad (2.29)$$

$$y_1 = \frac{1}{\sin(x_1)} \left(\frac{1}{2} + y_2 \sin(x_2) \right), \quad (2.30)$$

using Eq. (2.14) and (2.15). This results in

$$y_1 = \frac{1 + \sqrt{2}}{4\sqrt{2}}, \quad x_1 = \frac{\pi}{2}, \quad (2.31)$$

$$y_2 = \frac{1 - \sqrt{2}}{4\sqrt{2}}, \quad x_2 = \frac{3\pi}{2} \quad (2.32)$$

and three equivalent solutions based on the symmetries of Eq. (2.14) and (2.15).

⁷ In Chap. 1, this allocation is called practical allocation, but on average it equals the optimal allocation because the shift rule is unbiased.

⁸ Again: on average.

The variance then is $\sigma_{\text{PS4}}^2 = \sigma^2/s$ like it is for the optimal two-term rule⁹ and again it may be further reduced by introducing a bias via a multiplicative prefactor λ , with the same optimal λ^* as before. For both the specialized four-term and two-term rules in [101], the minimal variance is $\sigma_{[101]}^2 = \sigma^2/s$, as the prefactors are equally large and sum to one.

In conclusion, under the constant variance assumption, the variance for all discussed two- and four-term parameter-shift rules is the same at a given measurement budget. This shows that they are equally expensive on a quantum device, for which the number of measurements instead of the number of distinct circuits is relevant.

2.3 Contributions to the first publication

Here I describe my contributions to the attached publication, which was published in the journal *Quantum* [14] and is freely available online, with functioning (hyper)links. Overall, the core idea of using trigonometric Lagrange interpolation and Fourier analysis on PQC-based objective functions was developed and discussed within the team of authors, starting at some (but unfortunately not all) of the literature mentioned above. This means that it is difficult to discern my contribution to the core idea from those of the other authors. The following parts, however, were mainly carried out by me: first, computing the shift rule cost in detail, including the analysis of shot cost and number of unique circuits, and comparing them to differentiation via decomposition and existing shift rules. Second, the full generalization to arbitrary frequency spectra and perturbed unitaries requiring the stochastic shift rule. Third, the application to the metric tensor and its comparison to other methods of computing the tensor. Fourth, the application to QAOA and the relation to Rotosolve and quantum analytic descent (QAD), including the generalized QAD code and a program to estimate the required resources for QAOA. Finally, I wrote the majority of the manuscript, but of course in collaboration with the other authors, and implemented the general shift rules and the generalized version of Rotosolve in PennyLane [80], except for the parts contributed by Robert A. Lang.

Compared to the main text, the changes and additions to the notation shown in Tab. 2.1 are used in the publication. Additionally there are some specific notations used locally in single sections as defined therein.

⁹ This is not a contradiction to Eq. (1.73) in Chap. 1 because the definitions here imply $\Omega = 1$ for two-term gates and $\Omega = 1/2$ for four-term gates, so that $R^2\Omega^2$ is the same.

Symbol	in main text	Meaning
a, b	—	Graph vertex
b_{jk}	—	Matrix entry of B in eigenbasis of a gate
c_ℓ / c_{jk}	$\frac{1}{2}(a_\ell - ib_\ell)$	(Complex) Fourier coefficient of E
d	2^N	Hilbert space dimension
\mathcal{F}	—	Metric tensor of a PQC
$G(\mathcal{V}, \mathcal{E})$	—	Graph with vertices \mathcal{V} and edges \mathcal{E}
$\mathcal{G}(x)[\cdot]$	$i[(xG + F, \cdot)]$	Generator channel
H	—	Hessian of E
H_P	H	Hamiltonian (of a MAXCUT instance)
K	—	Degree of a regular graph
k	—	Halved degree $\lfloor K/2 \rfloor$
M	—	Number of edges in a graph
N	s	Shot budget
$U_{M/P}$	—	Mixer/problem layer in QAOA
\mathcal{P}	—	Number of Pauli rotations in a decomposition
r	$ \Lambda(G) $	Number of unique generator eigenvalues
\mathbf{v}	\mathbf{e}	Canonical basis vector
\mathbf{x}	$\boldsymbol{\theta}$	Parameters of a PQC
$ \psi\rangle$	$ \phi\rangle$	Quantum state prepared by part of a PQC
Δ	—	Linear combination of estimators of E
ε^2	$\mathbb{V}[\cdot]$	Variance of an estimator
$[d]$	—	The set $\{1, \dots, d\}$ for an integer d
$[d]_0$	—	$\{0\} \cup [d]$

Table 2.1: Notation changes and additions of the following publication compared to the main text of this thesis.

General parameter-shift rules for quantum gradients

David Wierichs^{1,2}, Josh Izaac¹, Cody Wang³, and Cedric Yen-Yu Lin³

¹Xanadu, Toronto, ON, M5G 2C8, Canada

²Institute for Theoretical Physics, University of Cologne, Germany

³AWS Quantum Technologies, Seattle, Washington 98170, USA

Variational quantum algorithms are ubiquitous in applications of noisy intermediate-scale quantum computers. Due to the structure of conventional parametrized quantum gates, the evaluated functions typically are finite Fourier series of the input parameters. In this work, we use this fact to derive new, general parameter-shift rules for single-parameter gates, and provide closed-form expressions to apply them. These rules are then extended to multi-parameter quantum gates by combining them with the stochastic parameter-shift rule. We perform a systematic analysis of quantum resource requirements for each rule, and show that a reduction in resources is possible for higher-order derivatives. Using the example of the quantum approximate optimization algorithm, we show that the generalized parameter-shift rule can reduce the number of circuit evaluations significantly when computing derivatives with respect to parameters that feed into many gates. Our approach additionally reproduces reconstructions of the evaluated function up to a chosen order, leading to known generalizations of the Rotosolve optimizer and new extensions of the quantum analytic descent optimization algorithm.

1 Introduction

With the advent of accessible, near-term quantum hardware, the ability to rapidly test and prototype quantum algorithms has never been as approachable [1, 2, 3, 4]. However, many of the canonical quantum algorithms developed over the last three decades remain unreachable in practice — requiring a large number of error corrected qubits and significant circuit depth. As a result, a new class of quantum algorithms — variational quantum algorithms (VQAs) [5, 6] — have come to shape the noisy intermediate-scale quantum (NISQ) era. First rising to prominence with the introduction of the variational quantum eigensolver (VQE) [7], they have evolved to cover topics such as optimization [8], quantum chemistry [9, 10, 11, 12, 13], integer factorization [14], compilation [15], quantum control [16], matrix diagonaliza-

David Wierichs: wierichs@thp.uni-koeln.de

tion [17, 18], and variational quantum machine learning [19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31].

These algorithms have a common structure: a parametrized circuit is executed and a cost function is composed from expectation values measured in the resulting state. A classical optimization routine is then used to optimize the circuit parameters by minimizing said cost function. Initially, gradient-free optimization methods, such as Nelder-Mead and COBYLA, were common. However, gradient-based optimization provides significant advantages, from convergence guarantees [32] to the availability of workhorse algorithms (e.g., stochastic gradient descent) and software tooling developed for machine learning [33, 34, 35, 36, 37].

The so-called parameter-shift rule [16, 23, 38, 39] can be used to estimate the gradient for these optimization techniques, without additional hardware requirements and — in contrast to naïve numerical methods — without bias; the cost function is evaluated at two shifted parameter positions, and the rescaled difference of the results forms an unbiased estimate of the derivative. However, this two-term parameter-shift rule is restricted to gates with two distinct eigenvalues, potentially requiring expensive decompositions in order to compute hardware-compatible quantum gradients [40]. While various extensions to the shift rule have been discovered, they remain restricted to gates with a particular number of distinct eigenvalues [10, 41].

In this manuscript, we use the observation that the restriction of a variational cost function to a single parameter is a finite Fourier series [42, 43, 44, 45]; as a result, the restricted cost function can be *reconstructed* from circuit evaluations at shifted positions using a discrete Fourier transform (DFT). By analytically computing the derivatives of the Fourier series, we extract general parameter-shift rules for arbitrary quantum gates and provide closed-form expressions to apply them. In the specific case of unitaries with equidistant eigenvalues, the general parameter-shift rule recovers known parameter-shift rules from the literature, including the original two-term parameter-shift rule. We then generalize our approach in two steps: first from equidistant to arbitrary eigenvalues of the quantum gate, and from there — by making use of stochastic parameter shifts — to more complicated unitaries like multi-parameter gates. This enables us

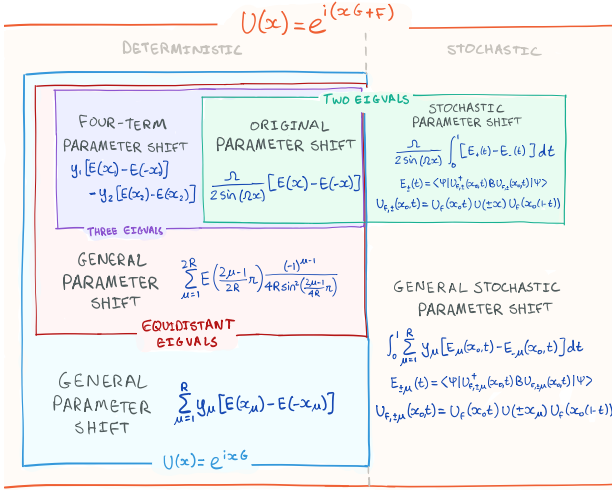


Figure 1: Overview of existing and new parameter-shift rules for first-order univariate derivatives as Venn diagram on the space of quantum gates. Each rule produces the analytic derivative for a set of gates, with more general rules reproducing the more specific ones. For gates of the form $U(x) = \exp(ixG)$ the rules are deterministic (*left*) whereas more involved gates of the form $U_F = \exp(ixG + F)$ require stochastic evaluations of shifted values (*right*). See Sec. 2.2 for a summary of previously known shift rules. The fermionic four-term shift rule in Ref. [41] covers the same gates as the shown four-term rule (*purple*).

to cover *all* practically relevant quantum gates. An overview of the existing parameter-shift rules and our new results is shown in Fig. 1.

Afterwards, we perform an extensive resource analysis to compare the computational expenses required by both the general shift rule presented here, and decomposition-based approaches. In particular, we note that evaluating the cost of gradient recipes by comparing the number of unique executed circuits leads to fundamentally different conclusions on the optimal differentiation technique than when comparing the total number of measurements.

Our analysis not only is fruitful for understanding the structure of variational cost functions, but also has several practical advantages. Firstly, second-order derivatives (such as the Hessian [46] and the Fubini-Study metric tensor [47, 48]) can be computed with fewer evaluations compared to naively iterating the two-term parameter-shift rule. We also show, using the example of the *quantum approximate optimization algorithm* (QAOA), that the generalized parameter-shift rule can reduce the number of quantum circuit evaluations required for ansätze with repeated parameters.

Finally, we generalize the *quantum analytic descent* (QAD) algorithm [49] using the reconstruction of variational cost functions discussed here. We also reproduce the known generalizations of *Rotosolve* [50, 51] from single Pauli rotations to groups of rotations controlled by the same parameter [42, 45]; reconstruct-

ing functions with *arbitrary* spectrum extends this algorithm even further. Furthermore, the cost reduction for the gradient we present in the context of QAOA applies to Rotosolve as well. Similarly, future improvements that reduce the cost for gradient computations might improve the efficiency of these model-based algorithms, based on the analysis presented here.

This manuscript is structured as follows. In Sec. 2, we lay out the setting for our results by deriving the general functional form for variational cost functions, followed by a survey of existing parameter-shift rules. In Sec. 3 we show how to fully reconstruct univariate variational cost functions from a finite number of evaluations assuming an equidistant frequency spectrum, and derive parameter-shift rules for arbitrary-order univariate derivatives, including a generalization of the stochastic parameter-shift rule. In Sec. 4 we demonstrate how to compute second-order derivatives, in particular the Hessian and the metric tensor, more cheaply compared to existing methods. In Sec. 5 we discuss applications, applying the new generalized parameter-shift rules to QAOA, and using the full univariate reconstruction to extend existing model-based optimization methods. We end the main text in Sec. 6 with a discussion of our work and potential future directions. Finally, in the appendix we summarize some technical derivations (App. A), and extend the results to more general frequency spectra (App. B). The general stochastic parameter-shift rule and details on quantum analytic descent can be found in Apps. C and D.

Related work: In Ref. [42], the functions of VQAs were considered as Fourier series and parameter-shift rules were derived. Regarding the shift rules, the authors of Ref. [42] consider integer eigenvalues and derive a rule with $2R + 1$ evaluations for equidistant eigenvalues. In particular, the two-term and four-term shift rules are reviewed and formulated as special cases with *fewer* evaluations than the general result presented there. In contrast, our work results in the exact generalization of those shift rules, which requires $2R$ evaluations. Remarkably, Refs. [42, 45] also propose a generalized Rotosolve algorithm prior to its eponymous paper.

In addition, during the final stages of preparation of this work, a related work considering algebraic extensions of the parameter-shift rule appeared online [52]. The general description of quantum expectation values in Sec. 2.1 of the present work, along with its initial consequences in Sec. 3.1, are shown in Sec. II A of this preprint. We present a simpler derivation and further explore the implications this description has. The generalization of the parameter-shift rule in Ref. [52] is obtained by decomposing the gate generator using Cartan subalgebras, which can yield fewer shifted evaluations than decompositions of the gate itself. In particular, decompositions into

non-commuting terms, which do not lead to a gate decomposition into native quantum gates directly, can be used in this approach.

At a similar time, yet another work appeared [53], presenting a derivation similar to Sec. 2.1 and parameter-shift rules for the first order derivative. These rules are based on the ideas discussed here in Secs. 3.1 and 3.2.

2 Background

We start by deriving the form of a VQA cost function of a single parameter for a general single-parameter quantum gate. Then we review known parameter-shift rules and briefly discuss resource measures to compare these gradient recipes.

2.1 Cost functions arising from quantum gates

Let us first consider the expectation value for a general gate $U(x) = \exp(ixG)$, defined by a Hermitian generator G and parametrized by a single parameter x . Let $|\psi\rangle$ denote the quantum state that U is applied to, and B the measured observable¹. The eigenvalues of $U(x)$ are given by $\{\exp(i\omega_j x)\}_{j \in [d]}$ with real-valued $\{\omega_j\}_{j \in [d]}$ where we denote $[d] := \{1, \dots, d\}$ and have sorted the ω_j to be non-decreasing. Thus, we have:

$$E(x) := \langle \psi | U^\dagger(x) B U(x) | \psi \rangle \quad (1)$$

$$= \sum_{j,k=1}^d \overline{\psi_j e^{i\omega_j x}} b_{jk} \psi_k e^{i\omega_k x} \quad (2)$$

$$= \sum_{\substack{j,k=1 \\ j < k}}^d \left[\overline{\psi_j} b_{jk} \psi_k e^{i(\omega_k - \omega_j)x} + \psi_j b_{jk} \overline{\psi_k} e^{i(\omega_k - \omega_j)x} \right] + \sum_{j=1}^d |\psi_j|^2 b_{jj}, \quad (3)$$

where we have expanded B and $|\psi\rangle$ in the eigenbasis of U , denoted by b_{jk} and ψ_j , respectively.

We can collect the x -independent part into coefficients $c_{jk} := \overline{\psi_j} b_{jk} \psi_k$ and introduce the R *unique positive* differences $\{\Omega_\ell\}_{\ell \in [R]} := \{\omega_k - \omega_j | j, k \in [d], \omega_k > \omega_j\}$. Note that the differences are not necessarily equidistant, and that for $r = |\{\omega_j\}_{j \in [d]}|$ *unique* eigenvalues of the gate generator, there are at most $R \leq \frac{r(r-1)}{2}$ unique differences. However, many quantum gates will yield $R \leq r$ *equidistant* differences in-

¹Here we consider any pure state in the Hilbert space; in the context of VQAs, $|\psi\rangle$ is the state prepared by the subcircuit prior to $U(x)$. Similarly, B includes the subcircuit following up on $U(x)$.

stead; a common example for this is

$$G = \sum_{k=1}^{\mathcal{P}} \pm P_k \quad (4)$$

for commuting Pauli words P_k ($P_k P_{k'} = P_{k'} P_k$), which yields the frequencies $[\mathcal{P}]$ and thus $R = \mathcal{P}$.

In the following, we implicitly assume a mapping between the two indices $j, k \in [d]$ and the frequency index $\ell \in [R]$ such that $c_\ell = c_{\ell(j,k)}$ is well-defined². We can then write the expectation value as a trigonometric polynomial (a finite-term Fourier series):

$$E(x) = a_0 + \sum_{\ell=1}^R c_\ell e^{i\Omega_\ell x} + \sum_{\ell=1}^R \overline{c_\ell} e^{-i\Omega_\ell x} \quad (5)$$

$$= a_0 + \sum_{\ell=1}^R a_\ell \cos(\Omega_\ell x) + b_\ell \sin(\Omega_\ell x), \quad (6)$$

with frequencies given by the differences $\{\Omega_\ell\}$, where we defined $c_\ell := \frac{1}{2}(a_\ell - ib_\ell) \forall \ell \in [R]$ with $a_\ell, b_\ell \in \mathbb{R}$, and $a_0 := \sum_j |\psi_j|^2 b_{jj} \in \mathbb{R}$.

Since $E(x)$ is a finite-term Fourier series, the coefficients $\{a_\ell\}$ and $\{b_\ell\}$ can be obtained from a finite number of evaluations of $E(x)$ through a *discrete Fourier transform*. This observation (and variations thereof in Sec. 3) forms the core of this work: we can obtain the full functional form of $E(x)$ from a finite number of evaluations of $E(x)$, from which we can compute arbitrary order derivatives.

2.2 Known parameter-shift rules

Parameter-shift rules relate derivatives of a quantum function to evaluations of the function itself at different points. In this subsection, we survey known parameter-shift rules in the literature.

For functions of the form (6) with a single frequency $\Omega_1 = \Omega$ (i.e., G has two eigenvalues), the derivative can be computed via the parameter-shift rule [16, 23, 38]

$$E'(0) = \frac{\Omega}{2 \sin(\Omega x_1)} [E(x_1) - E(-x_1)], \quad (7)$$

where x_1 is a freely chosen shift angle from $(0, \pi)$ ³.

This rule was generalized to gates with eigenvalues $\{-1, 0, 1\}$, which leads to $R = 2$ frequencies, in Refs. [41, 10] in two distinct ways. The rule in Ref. [10] is an immediate generalization of the one above:

$$E'(0) = y_1 [E(x_1) - E(-x_1)] - y_2 [E(x_2) - E(-x_2)], \quad (8)$$

²That is, $\ell(j, k) = \ell(j', k') \Leftrightarrow \omega_k - \omega_j = \omega_{k'} - \omega_{j'}$.

³The position 0 for the derivative is chosen for convenience but the rule can be applied at any position. To see this, note that shifting the argument of E does not change its functional form.

with freely chosen shift angles $x_{1,2}$ and corresponding coefficients $y_{1,2}$, requiring four evaluations to obtain $E'(0)$. A particularly symmetric choice of shift angles is $x_{1,2} = \pi/2 \mp \pi/4$ with coefficients $y_{1,2} = \frac{\sqrt{2 \pm 1}}{2\sqrt{2}}$. In contrast, the rule in Ref. [41] makes use of an auxiliary gate to implement slightly altered circuits, leading to a structurally different rule:

$$E'(0) = \frac{1}{4}[E_+^+ - E_-^+ + E_+^- - E_-^-], \quad (9)$$

where E_\pm^α is the measured energy when replacing the gate $U(x)$ in question by $U(x \pm \pi/2) \exp(\mp \alpha i \frac{\pi}{4} P_0)$ and P_0 is the projector onto the zero-eigenspace of the generator of U . Remarkably, this structure allows a reduction of the number of distinct circuit evaluations to two if the circuit and the Hamiltonian are real-valued, which is often the case for simulations of fermionic systems and forms a unique feature of this approach. This second rule is preferable whenever this condition is fulfilled, the auxiliary gates $\exp(\pm i \frac{\pi}{4} P_0)$ are available, and simultaneously the number of distinct circuits is the relevant resource measure.

Furthermore, the two-term parameter-shift rule Eq. (7) was generalized to gates with the more complicated gate structure $U_F(x) = \exp(i(xG + F))$ via the *stochastic parameter-shift rule* [39]

$$E'(x_0) = \frac{\Omega}{2 \sin(\Omega x_1)} \int_0^1 [E_+(t) - E_-(t)] dt. \quad (10)$$

Here, $E_\pm(t)$ is the energy measured in the state prepared by a modified circuit that splits $U_F(x_0)$ into $U_F(tx_0)$ and $U_F((1-t)x_0)$, and interleaves these two gates with $U_{F=0}(\pm x_1)$. See Sec. 3.6 and App. C for details. The first-order parameter-shift rules summarized here and their relationship to each other is also visualized in Fig. 1.

A parameter-shift rule for higher-order derivatives based on repeatedly applying the original rule has been proposed in Ref. [46]. The shift can be chosen smartly so that two function evaluations suffice to obtain the second-order derivative:

$$E''(0) = \frac{1}{2}[E(\pi) - E(0)], \quad (11)$$

which like Eq. (7) is valid for single-frequency gates. Various expressions to compute combinations of derivatives with few evaluations were explored in Ref. [54].

2.3 Resource measures for shift rules

While the original parameter-shift rule Eq. (7) provides a unique, unbiased method to estimate the derivative $E'(0)$ via evaluations of E if it contains a single frequency, we will need to compare different shift rules for the general case. To this end, we consider two resource measures. Firstly, the number of distinct circuits that need to be evaluated to obtain all

terms of a shift rule, N_{eval} . This is a meaningful quantity on both, simulators that readily produce many measurement samples after executing each unique circuit once, as well as quantum hardware devices that are available via cloud services. In the latter case, quantum hardware devices are typically billed and queued per unique circuit, and as a result N_{eval} often dictates both the financial and time cost. Note that overhead due to circuit compilation and optimization scale with this quantity as well.

Secondly, we consider the overall number N of measurements — or *shots* — irrespective of the number of unique circuits they are distributed across. To this end, we approximate the physical (one-shot) variance σ^2 of the cost function E to be constant across its domain⁴. For an arbitrary quantity Δ computed from \mathcal{M} values of E via a shift rule,

$$\Delta = \sum_{\mu}^{\mathcal{M}} y_{\mu} E(\mathbf{x}_{\mu}), \quad (12)$$

we obtain the variance for the estimate of Δ as

$$\varepsilon^2 = \sum_{\mu}^{\mathcal{M}} |y_{\mu}|^2 \frac{\sigma^2}{N_{\mu}}, \quad (13)$$

where N_{μ} expresses the number of shots used to measure $E(\mathbf{x}_{\mu})$. For a total budget of N shots, the optimal shot allocation is $N_{\mu} = N |y_{\mu}| / \|\mathbf{y}\|_1$ such that

$$N = \frac{\sigma^2 \|\mathbf{y}\|_1^2}{\varepsilon^2}. \quad (14)$$

This can be understood as the number of shots needed to compute Δ to a tolerable standard deviation ε .

The number of shots N is a meaningful quantity for simulators whose runtime scales primarily with the number of requested samples (e.g., Amazon Braket's TN1 tensor network simulator [1]), and for actual quantum devices when artificial resource measures like pricing per unique circuit and queuing time do not play a role.

In this work we will mostly use N_{eval} to compare the requirements of different parameter-shift rules as it is more accessible, does not rely on the assumption of constant physical variance like N does, and the coefficients \mathbf{y} to estimate N are simply not known analytically in most general cases. For the case of equidistant frequencies and shift angles as discussed in Sec. 3.4 we will additionally compare the number of shots N in Sec. 3.5.

3 Univariate cost functions

In this section we study how a quantum cost function, which in general depends on multiple parameters, varies if only one of these parameters is changed.

⁴As it is impossible in general to compute σ^2 analytically, we are forced to make this potentially very rough approximation.

The results of this section will be sufficient to evaluate the gradient as well as the diagonal of the Hessian of a quantum function. We restrict ourselves to functions that can be written as the expectation value of an observable with respect to a state that is prepared using a unitary $U(x) = \exp(ixG)$ — capturing the full dependence on x . That is, all parameters but x are fixed and the operations they control are considered as part of the prepared state and the observable. As shown in Sec. 2.1, this yields a trigonometric polynomial, i.e.,

$$E(x) = a_0 + \sum_{\ell=1}^R a_\ell \cos(\Omega_\ell x) + b_\ell \sin(\Omega_\ell x). \quad (15)$$

In the following, we will assume the frequencies to be equidistant, i.e., $\Omega_\ell = \ell\Omega$, and generalize to arbitrary frequencies in App. B. While it is easy to construct gate sequences that do not lead to equidistant frequencies, many conventional gates and layers of gates do yield such a regular spectrum. The equidistant frequency case has two major advantages over the general case: we can derive closed-form parameter-shift rules (Sec. 3.4); and the number of circuits required for the parameter-shift rule scales much better (Sec. 3.5).

Without loss of generality, we further restrict the frequencies to integer values, i.e., $\Omega_\ell = \ell$. For $\Omega \neq 1$, we may rescale the function argument to achieve $\Omega_\ell = \ell$ and once we reconstruct the rescaled function, the original function is available, too.

3.1 Determining the full dependence on x

As we have seen, the functional form of $E(x)$ is known exactly. We can thus determine the function by computing the $2R + 1$ coefficients $\{a_\ell\}$ and $\{b_\ell\}$. This is the well-studied problem of *trigonometric interpolation* (see e.g., [55, Chapter X]).

To determine $E(x)$ completely, we can simply evaluate it at $2R + 1$ distinct points $x_\mu \in [-\pi, \pi)$. We obtain a set of $2R + 1$ equations

$$E(x_\mu) = a_0 + \sum_{\ell=1}^R a_\ell \cos(\ell x_\mu) + b_\ell \sin(\ell x_\mu), \quad \mu \in [2R]_0$$

where we denote $[2R]_0 := \{0, 1, \dots, 2R\}$. We can then solve these linear equations for $\{a_\ell\}$ and $\{b_\ell\}$; this process is in fact a nonuniform *discrete Fourier transform* (DFT).

A reasonable choice is $x_\mu = \frac{2\pi\mu}{2R+1}, \mu = -R, \dots, R$, in which case the transform is the usual (uniform) DFT. For this choice, an explicit reconstruction for E follows directly from [55, Chapter X]; we reproduce it in App. A.1.1.

3.2 Determining the odd part of $E(x)$

It is often the case in applications that we only need to determine the odd part of E ,

$$E_{\text{odd}}(x) = \frac{1}{2}(E(x) - E(-x)) \quad (16)$$

$$= \sum_{\ell=1}^R b_\ell \sin(\ell x). \quad (17)$$

For example, calculating odd-order derivatives of $E(x)$ at $x = 0$ only requires knowledge of $E_{\text{odd}}(x)$, since those derivatives of the even part vanish. Note that the reference point with respect to which E_{odd} is odd may be chosen arbitrarily, and does not have to be 0.

The coefficients in E_{odd} can be determined by evaluating E_{odd} at R distinct points x_μ with $0 < x_\mu < \pi$. This gives us a system of R equations

$$E_{\text{odd}}(x_\mu) = \sum_{\ell=1}^R b_\ell \sin(\ell x_\mu), \quad \mu \in [R] \quad (18)$$

which we can use to solve for the R coefficients $\{b_\ell\}$.

Using Eq. (16) we see that each evaluation of E_{odd} can be done with two evaluations of $E(x)$. Thus, the odd part of E can be completely determined with $2R$ evaluations of E , saving one evaluation compared to the general case. Note however that the saved $E(0)$ evaluation is evaluated regardless in many applications, and may be used to recover the full reconstruction — so, in effect, this saving does not have a significant impact⁵.

3.3 Determining the even part of $E(x)$

We might similarly want to obtain the even part of E ,

$$E_{\text{even}}(x) = \frac{1}{2}(E(x) + E(-x)) \quad (19)$$

$$= a_0 + \sum_{\ell=1}^R a_\ell \cos(\ell x), \quad (20)$$

which can be used to compute even-order derivatives of E .

Determining $E_{\text{even}}(x)$ requires $R + 1$ evaluations of E_{even} , which leads to $2R + 1$ evaluations of E for arbitrary frequencies. However, in the case where Ω_ℓ are integers, $R + 1$ evaluations of E_{even} can be obtained

⁵If $E(0)$ is available, we can recover the full function, allowing us to, for example, evaluate its second derivative $E''(0)$ “for free”. However, in practice many more repetitions may be needed for reasonable accuracy. This fact was already noted in [46] for the $R = 1$ case.

with $2R$ evaluations of $E(x)$ by using periodicity:

$$E_{\text{even}}(0) = E(0) \quad (21)$$

$$E_{\text{even}}(x_\mu) = \frac{1}{2}(E(x_\mu) + E(-x_\mu)), \quad (22)$$

$$0 < x_\mu < \pi, \mu \in [R-1]$$

$$E_{\text{even}}(\pi) = E(\pi). \quad (23)$$

Thus, in this case $2R$ evaluations of $E(x)$ suffice to determine its even part, saving one evaluation over the general case. In contrast to the odd part, this saving genuinely reduces the required computations as $E(0)$ is also used in the cheaper computation of $\{a_\ell\}$; therefore, if $E(0)$ is already known, we only require $2R-1$ new evaluations.

We note that even though both the odd and the even part of $E(x)$ require $2R$ evaluations, the full function can be obtained at the price of $2R+1$ evaluations.

3.4 Explicit parameter-shift formulas

Consider again the task of determining E_{odd} (E_{even}) based on its value at the shifted points $\{x_\mu\}$ with $\mu \in [R]$ ($\mu \in [R]_0$). This can be done by linearly combining elementary functions that vanish on all but one of the $\{x_\mu\}$, i.e., kernel functions, using the evaluation $E(x_\mu)$ as coefficients. If we restrict ourselves to evenly spaced points $x_\mu = \frac{2\mu-1}{2R}\pi$ ($x_\mu = \frac{\mu}{R}\pi$), we can choose these functions to be Dirichlet kernels. In addition to a straightforward reconstruction of the odd (even) function this delivers the *general parameter-shift rules*, which we derive in App. A.1:

$$E'(0) = \sum_{\mu=1}^{2R} E\left(\frac{2\mu-1}{2R}\pi\right) \frac{(-1)^{\mu-1}}{4R \sin^2\left(\frac{2\mu-1}{4R}\pi\right)}, \quad (24)$$

$$E''(0) = -E(0) \frac{2R^2+1}{6} + \sum_{\mu=1}^{2R-1} E\left(\frac{\mu\pi}{R}\right) \frac{(-1)^{\mu-1}}{2 \sin^2\left(\frac{\mu\pi}{2R}\right)}. \quad (25)$$

We remark that derivatives of higher order can be obtained in an analogous manner, and with the same function evaluations for all odd (even) orders. Furthermore, this result reduces to the known two-term (Eq. (7)) and four-term (Eq. (8)) parameter-shift rules for $R=1$ and $R=2$, respectively, as well as the second-order derivative for $R=1$ (Eq. (11)).

We again note that the formulas above use different evaluation points for the first and second derivatives ($2R$ evaluations for each derivative). Closed-form parameter-shift rules that use $2R+1$ shared points can be obtained by differentiating the reconstruction formula Eq. (57).

3.5 Resource comparison

As any unitary may be compiled from (single-qubit) Pauli rotations, which satisfy the original parameter-

shift rule, and CNOT gates, an alternative approach to compute $E'(0)$ is to decompose $U(x)$ into such gates and combine the derivatives based on the elementary gates. As rotation gates about any multi-qubit Pauli word satisfy the original parameter-shift rule as well, a more coarse-grained decomposition might be possible and yield fewer evaluations for this approach.

For instance, for the MAXCUT QAOA ansatz⁶ on a graph $G=(\mathcal{V}, \mathcal{E})$ with vertices \mathcal{V} and edges \mathcal{E} , one of the operations is to evolve under the problem Hamiltonian:

$$U_P(x) \propto \exp\left(-i\frac{x}{2} \sum_{(a,b) \in \mathcal{E}} Z_a Z_b\right) \quad (26)$$

$$= \prod_{(a,b) \in \mathcal{E}} \exp\left(-i\frac{x}{2} Z_a Z_b\right). \quad (27)$$

Eq. (26) treats $U_P(x)$ as a single operation with at most $M=|\mathcal{E}|$ frequencies $1, \dots, R \leq M$, and we can apply the generalized parameter-shift rules of this section. Alternatively, we could decompose $U_P(x)$ with Eq. (27), apply the two-term parameter-shift rule to each R_{ZZ} rotation, and sum up the contributions using the chain rule.

3.5.1 Number of unique circuits

If there are \mathcal{P} gates that depend on x in the decomposition, this approach requires $2\mathcal{P}$ unique circuit evaluations; as a result, the general parameter-shift rule is cheaper if $R < \mathcal{P}$. The evaluations used in the decomposition-based approach cannot be expressed by E directly because the parameter is shifted only in one of the \mathcal{P} gates per evaluation, which makes the general parameter-shift rule more convenient and may reduce compilation overhead for quantum hardware, and the number of operations on simulators.

In order to compute $E''(0)$ via the decomposition, we need to obtain and sum the full Hessian of all elementary gates that depend on x (see App. A.4.2), which requires $2\mathcal{P}^2 - \mathcal{P} + 1$ evaluations, including $E(0)$, and thus is significantly more expensive than the $2R$ evaluations for the general parameter-shift rule.

While the derivatives can be calculated from the functional form of E_{odd} or E_{even} , the converse is not true for $R > 1$, i.e., the full functional dependence on x cannot be extracted from the first and second derivative alone. Therefore, the decomposition-based approach would demand a full multivariate reconstruction for all \mathcal{P} parametrized elementary gates to obtain this dependence, requiring $\mathcal{O}(2^{\mathcal{P}})$ evaluations. The approach shown here allows us to compute the dependence in $2R+1$ evaluations and thus is the only method for which the univariate reconstruction is viable.

⁶A more detailed description of the QAOA ansatz can be found in Sec. 5.1.

3.5.2 Number of shots

For equidistant evaluation points, we explicitly know the coefficients of the first and second-order shift rule given in Eqs. (24, 25), and thus can compare the variance of the derivatives in the context and under the assumptions of Sec. 2.3.

The coefficients satisfy (see App. A.4.1)

$$\sum_{\mu=1}^{2R} \left(4R \sin^2 \left(\frac{2\mu-1}{4R} \pi \right) \right)^{-1} = R$$

$$\frac{2R^2+1}{6} + \sum_{\mu=1}^{2R-1} \left(2 \sin^2 \left(\frac{\mu\pi}{2R} \right) \right)^{-1} = R^2.$$

This means that the variance-minimizing shot allocation requires a shot budget of

$$N_{\text{genPS}, 1} = \frac{\sigma^2 R^2}{\varepsilon^2} \quad (28)$$

$$N_{\text{genPS}, 2} = \frac{\sigma^2 R^4}{\varepsilon^2} \quad (29)$$

using the generalized parameter-shift rule for the first and second derivative, respectively.

Assuming integer-valued frequencies in the cost function typically means, in the decomposition-based approach, that x enters the elementary gates without any additional prefactors⁷. Thus, optimally all evaluations for the first-order derivative rule are performed with the same portion of shots; whereas the second-order derivative requires an adapted shot allocation which, in particular, measures $E(0)$ with high precision as it enters $E''(0)$ with the prefactor $\mathcal{P}/2$. This yields (see App. A.4.2)

$$N_{\text{decomp}, 1} = \frac{\sigma^2 \mathcal{P}^2}{\varepsilon^2} \quad (30)$$

$$N_{\text{decomp}, 2} = \frac{\sigma^2 \mathcal{P}^4}{\varepsilon^2}. \quad (31)$$

Comparing with $N_{\text{genPS}, 1}$ and $N_{\text{genPS}, 2}$ above, we see that the shot budgets are equal at $\mathcal{P} = R$. That is, for both the first and second derivative, the general parameter-shift rule does not show lower shot requirements in general, in contrast to the previous analysis that showed a significantly smaller number of unique circuits for the second derivative. This shows that the comparison of recipes for gradients and higher-order derivatives crucially depends on the chosen resource measure. In specific cases we may be able to give tighter upper bounds on R so that $R < \mathcal{P}$ (see Sec. 5.1) and the general shift rule becomes favourable regarding the shot count as well.

3.6 General stochastic parameter-shift rule

Next, we will apply the *stochastic parameter-shift rule* to our general shift rule. For this section we do *not*

⁷Of course, one can construct less efficient decompositions that do not satisfy this rule of thumb.

assume the frequencies to be equidistant but address arbitrary spectra directly. Additionally we make the reference point x_0 at which the derivative is computed explicit.

In Ref. [39], the authors derive the stochastic parameter-shift rule for gates of the form

$$U_F(x) = \exp(i(xG + F)) \quad (32)$$

where G is a Hermitian operator with eigenvalues ± 1 (so that $G^2 = \mathbb{1}$), e.g., a Pauli word. F is any other Hermitian operator, which may not necessarily commute with G^8 . Key to the derivation of the stochastic rule is an identity relating the derivative of the quantum channel $\mathcal{U}_F(x)[\rho] = U_F^\dagger(x)\rho U_F(x)$ to the derivative of the generator channel $\mathcal{G}(x)[\rho] = i[(xG + F), \rho]$. We may extend this directly to the general parameter-shift rule for the case when $G^2 = \mathbb{1}$ is no longer satisfied (see App. C for the derivation):

$$E'(x_0) = \int_0^1 \sum_{\mu=1}^R y_\mu [E_\mu(x_0, t) - E_{-\mu}(x_0, t)] dt \quad (33)$$

$$E_{\pm\mu}(x_0, t) := \langle B \rangle_{U_F(t x_0) U(\pm x_\mu) U_F((1-t)x_0) |\psi\rangle}.$$

The integration is implemented in practice by sampling values for t for each measurement of $E_\mu(x_0, t)$ and $E_{-\mu}(x_0, t)$.

The stochastic parameter-shift rule in combination with the generalized shift rule in Eq. (24) allows for the differentiation of any unitary with equidistant frequencies. As F in $U_F(x)$ above is allowed to contain terms that depend on other variational parameters, this includes multi-parameter gates in particular. Furthermore, combining Eq. (33) with the generalized shift rule for arbitrary frequencies in Eq. (90) allows us to compute the derivative of *any* quantum gate as long as the frequencies of $U_{F=0}(x)$ are known. We thus obtain an improved rule for $U_{F \neq 0}(x)$ over the original stochastic shift rule whenever the generalized shift rule is beneficial for $U(x) = U_{F=0}(x)$, compared to the decomposition-based approach.

4 Second-order derivatives

As noted in Sec. 3.3, higher-order derivatives of univariate functions are easily computed using the even or odd part of the function. In the following sections, we will extend our discussion to multivariate functions $E(\mathbf{x})$, where derivatives may be taken with respect to different variables. Each single parameter dependence is assumed to be of the form Eq. (5), with equidistant (and by rescaling integer-valued) frequencies $\{\Omega_\ell^{(k)}\}_{\ell \in [R_k]} = [R_k]$ for the k th parameter. We

⁸If $GF = FG$, the exponential may be split into $\exp(ixG)$ and $\exp(iF)$ and we are back at the situation $\exp(ixG)$.

may collect the numbers of frequencies in a vector $(\mathbf{R})_k = R_k$. It will again be useful in the following to make the reference point \mathbf{x}_0 , at which these derivatives are computed, explicit.

4.1 Diagonal shift rule for the Hessian

Here we show how to compute the Hessian H of a multivariate function $E(\mathbf{x})$ at some reference point \mathbf{x}_0 using the Fourier series representation of E . We allow for single-parameter gates $U(x) = \exp(ixG)$ with equidistant frequencies and will use fewer evaluations of E than known schemes. An indication that this may be possible for gates with two eigenvalues was made in [54, Eq. (37)].

First, for the k th diagonal entry $H_{kk} = \partial_k^2 E(\mathbf{x}_0)$ of the Hessian, we previously noted in Sec. 3.3 that $2R_k$ evaluations are sufficient as it is the second derivative of a univariate restriction of E . Recall that one of the $2R_k$ evaluations is $E(\mathbf{x}_0)$; we can reuse this evaluation for all diagonal entries of H , and thus require $1 + \sum_{k=1}^n (2R_k - 1) = 2\|\mathbf{R}\|_1 - n + 1$ evaluations for the full diagonal. Further, if we compute the Hessian diagonal $(\nabla^{\odot 2} E)_k := \partial_k^2 E$ in addition to the gradient, we may reuse the $2\|\mathbf{R}\|_1$ evaluations computed for the gradient, only requiring a single additional function value, namely $E(\mathbf{x}_0)$. In this case, we do not make use of the periodicity $E(\mathbf{x}_0 + \pi \mathbf{v}_k) = E(\mathbf{x}_0 - \pi \mathbf{v}_k)$, where \mathbf{v}_k is the k th canonical basis vector, because this shift is not used in the gradient evaluation (see Sec. 3.2).

Next, for an off-diagonal entry $H_{km} = \partial_k \partial_m E(\mathbf{x}_0)$, consider the *univariate* trigonometric function that shifts the two parameters x_k and x_m *simultaneously*:

$$E^{(km)}(x) := E(\mathbf{x}_0 + x\mathbf{v}_{k,m}), \quad (34)$$

where we abbreviated $\mathbf{v}_{k,m} := \mathbf{v}_k + \mathbf{v}_m$. We show in App. A.2 that $E^{(km)}$ again is a Fourier series of x with $R_{km} = R_k + R_m$ equidistant frequencies. This means that we can compute $E^{(km)''}(0)$ via Eq. (25) with $R = R_{km}$, using $2R_{km} - 1$ evaluations of E (as we may reuse $E(\mathbf{x}_0)$ from the diagonal computation). Note that

$$\left. \frac{d^2}{dx^2} E^{(km)}(x) \right|_{x=0} = H_{kk} + H_{mm} + 2H_{km}, \quad (35)$$

and that we have already computed the diagonal entries. We thus may obtain H_{km} via the *diagonal parameter-shift rule*

$$H_{km} = \frac{1}{2} \left(E^{(km)''}(0) - H_{kk} - H_{mm} \right). \quad (36)$$

In Fig. 2, we visually compare the computation of H_{km} via the diagonal shift rule to the chained application of univariate parameter-shift rules for x_k and x_m .

As an example, consider the case when $R_k = R_m = 1$ (e.g., where all parametrized gates are of the form

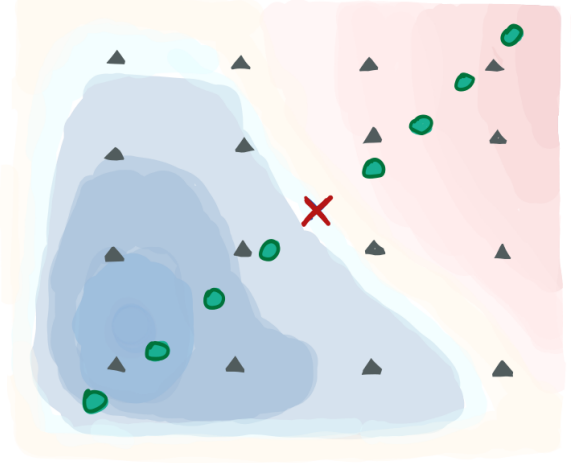


Figure 2: Visual representation of two approaches to compute a Hessian entry H_{km} at the position \mathbf{x}_0 (red cross). The parameters x_k and x_m lie on the coordinate axes and the heatmap displays the cost function $E(\mathbf{x})$. We may either combine the general shift rule for x_k and x_m (grey triangles) or compute the univariate derivative $E^{(km)''}(0)$ and extract H_{km} via Eq. (36) (green circles).

$\exp(ix_k G_k/2)$ with $G_k^2 = 1$). By setting $R = 2$ in Eq. (25), we obtain the explicit formula for $E^{(km)''}(0)$,

$$E^{(km)''}(0) = -\frac{3}{2}E(\mathbf{x}_0) - \frac{1}{2}E(\mathbf{x}_0 + \pi \mathbf{v}_{k,m}) \quad (37) \\ + E\left(\mathbf{x}_0 + \frac{\pi}{2} \mathbf{v}_{k,m}\right) + E\left(\mathbf{x}_0 - \frac{\pi}{2} \mathbf{v}_{k,m}\right)$$

which can be combined with Eq. (36) to give an explicit formula for the Hessian. This formula (for $R_k = R_m = 1$) was already discovered in [54, Eq. (37)].

The computation of H_{km} along the main diagonal in the x_k - x_m -plane can be modified by making use of the second diagonal as well: define $\bar{\mathbf{v}}_{k,m} := \mathbf{v}_k - \mathbf{v}_m$ and $\bar{E}^{(km)}(x) := E(\mathbf{x}_0 + x\bar{\mathbf{v}}_{k,m})$, and compute

$$\left. \frac{d^2}{dx^2} \bar{E}^{(km)}(x) \right|_{x=0} = H_{kk} + H_{mm} - 2H_{km}, \quad (38) \\ H_{km} = \frac{1}{4} \left(E^{(km)''}(0) - \bar{E}^{(km)''}(0) \right).$$

This means we can replace the dependence on the diagonal elements H_{kk} and H_{mm} by another univariate second-order derivative on the second diagonal. We will not analyze the resources required by this method in detail but note that for many applications it forms a compromise between the two approaches shown in Fig. 2.

We note that an idea similar to the ones presented here can be used for higher-order derivatives, but possibly requires more than one additional univariate reconstruction per derivative.

4.2 Resource comparison

For the Hessian computation, we will again look at the number of unique circuit evaluations N_{eval} and the number of shots N , as introduced in Sec. 2.3.

4.2.1 Number of unique circuits

In Tab. 1, we summarize the number of distinct circuit evaluations required to compute several combinations of derivatives of $E(\mathbf{x})$, either by decomposing the gate or by using the general parameter-shift rule together with the diagonal shift rule for the Hessian. We also include the generalized case of non-equidistant frequencies covered in App. B.2 for completeness. To obtain the cost for the repeated general shift rule, i.e., without the diagonal shift rule for the Hessian or decomposition, simply replace \mathcal{P} by \mathbf{R} in the left column.

For equidistant frequencies, the diagonal shift rule for H_{km} requires $2(R_k + R_m) - 1$ evaluations, assuming the diagonal and thus $E(\mathbf{x}_0)$ to be known already. Like the gradient, H_{km} may instead be computed by decomposing $U_k(x_k)$ and $U_m(x_m)$ into \mathcal{P}_k and \mathcal{P}_m elementary gates, respectively, and repeating the parameter-shift rule twice [46, 56]. All combinations of parameter shifts are required, leading to $4\mathcal{P}_k\mathcal{P}_m$ evaluations. Finally, as a third option, one may repeat the general parameter-shift rule in Eq. (24) twice, leading to $4R_kR_m$ evaluations⁹.

The repeated general shift rule requires strictly more circuit evaluations than the diagonal shift rule, since

$$2\|\mathbf{R}\|_1^2 - \|\mathbf{R}\|_1 + 1 > 2n\|\mathbf{R}\|_1 - \frac{1}{2}(n^2 + n - 2). \quad (39)$$

Similar to the discussion for the scaling of gradient computations, the optimal approach depends on $R_{k,m}$ and $\mathcal{P}_{k,m}$, but \mathcal{P} and R often have a linear relation so that the diagonal shift rule will be significantly cheaper for many cost functions than decomposing the unitaries.

4.2.2 Number of shots

Next we compare the numbers of measurements required to reach a precision ε . While the approach via repeated shift rules uses distinct circuit evaluations for each Hessian entry, the diagonal shift rule in Eq. (36) reuses entries of the Hessian and thus correlates the optimal shot allocations and the statistical errors of the Hessian entries. We therefore consider an error measure on the full Hessian matrix instead of a single entry, namely the root mean square of the Frobenius norm of the difference between the true and the estimated Hessian. This norm is computed in

⁹These $4R_kR_m$ shifted evaluations are *not* simultaneous shifts in both directions of the form Eq. (34).

App. A.5 for the three presented approaches, and we conclude the number of shots required to achieve a norm of ε to be

$$N_{\text{diag}} = \frac{\sigma^2}{2\varepsilon^2} \left[(\sqrt{n+1} + n - 2) \|\mathbf{R}\|_2^2 + \|\mathbf{R}\|_1^2 \right]^2 \quad (40)$$

$$N_{\text{genPS}} = \frac{\sigma^2}{2\varepsilon^2} \left[(\sqrt{2} - 1) \|\mathbf{R}\|_2^2 + \|\mathbf{R}\|_1^2 \right]^2 \quad (41)$$

$$N_{\text{decomp}} = \frac{\sigma^2}{2\varepsilon^2} \left[(\sqrt{2} - 1) \|\mathcal{P}\|_2^2 + \|\mathcal{P}\|_1^2 \right]^2 \quad (42)$$

In general, the diagonal shift rule for the Hessian is significantly less efficient than the repeated execution of the general parameter-shift rule if the shot count is the relevant resource measure. This is in sharp contrast to the number of unique circuits, which is strictly smaller for the diagonal shift rule. We note that the two resource measures yield *incompatible* recommendations for the computation of the Hessian. The overhead of the diagonal shift rule reduces to a (to leading order in n) constant prefactor if $R_k = R$ for all $k \in [n]$: in this case, we know $\|\mathbf{R}\|_1 = n = \|\mathbf{R}\|_2^2$ and therefore

$$\frac{N_{\text{diag}}}{N_{\text{genPS}}} = \frac{2n + \sqrt{n+1} - 2}{n + \sqrt{2} - 1} \xrightarrow{n \rightarrow \infty} 2. \quad (43)$$

4.3 Metric tensor

The Fubini-Study metric tensor \mathcal{F} is the natural metric on the manifold of (parametrized) quantum states, and the key ingredient in quantum natural gradient descent [48]. The component of the metric belonging to the parameters x_k and x_m can be written as

$$\mathcal{F}_{km}(\mathbf{x}_0) = \Re\left\{ \langle \partial_k \psi(\mathbf{x}) | \partial_m \psi(\mathbf{x}) \rangle \right\} \Big|_{\mathbf{x}=\mathbf{x}_0} \quad (44)$$

$$- \langle \partial_k \psi(\mathbf{x}) | \psi(\mathbf{x}) \rangle \langle \psi(\mathbf{x}) | \partial_m \psi(\mathbf{x}) \rangle \Big|_{\mathbf{x}=\mathbf{x}_0},$$

or, alternatively, as a Hessian [46]:

$$\mathcal{F}_{km}(\mathbf{x}_0) = -\frac{1}{2} \partial_k \partial_m |\langle \psi(\mathbf{x}) | \psi(\mathbf{x}_0) \rangle|^2 \Big|_{\mathbf{x}=\mathbf{x}_0}$$

$$=: \partial_k \partial_m f(\mathbf{x}_0). \quad (45)$$

It follows that we can compute the metric using the same method as for the Hessian, with $f(\mathbf{x})$ as the cost function. We know the value of f without shift as

$$f(\mathbf{x}_0) = -\frac{1}{2} |\langle \psi(\mathbf{x}_0) | \psi(\mathbf{x}_0) \rangle|^2 = -\frac{1}{2}. \quad (46)$$

The values with shifted argument can be calculated as the probability of the zero bitstring $\mathbf{0}$ when measuring the state $V^\dagger(\mathbf{x})V(\mathbf{x}_0)|\mathbf{0}\rangle$ in the computational basis, which requires circuits with up to doubled depth compared to the original circuit $V(\mathbf{x})$. Alternatively, we may use a Hadamard test to implement f , requiring an auxiliary qubit, two operations controlled by that qubit as well as a measurement on it, but only

Quantity	Decomposition	Gen. shift rule, equidistant	Gen. shift rule
$E(\mathbf{x}_0)$	1	1	1
$\partial_k E(\mathbf{x}_0)$	$2\mathcal{P}_k$	$2R_k$	$2R_k$
$\nabla E(\mathbf{x}_0)$	$2\ \mathcal{P}\ _1$	$2\ \mathbf{R}\ _1$	$2\ \mathbf{R}\ _1$
$\partial_k^2 E(\mathbf{x}_0)$	$2\mathcal{P}_k^2 - \mathcal{P}_k + 1$	$2R_k$	$2R_k + 1$
$\nabla^{\odot 2} E(\mathbf{x}_0)$	$2\ \mathcal{P}\ _2^2 - \ \mathcal{P}\ _1 + 1$	$2\ \mathbf{R}\ _1 - n + 1$	$2\ \mathbf{R}\ _1 + 1$
$\partial_k \partial_m E(\mathbf{x}_0)$	$4\mathcal{P}_k \mathcal{P}_m$	$2(R_k + R_m) - 1^{(*)}$	$4R_k R_m + 2R_k + 2R_m - 4^{(*)}$
$\nabla^{\otimes 2} E(\mathbf{x}_0)$	$2\ \mathcal{P}\ _1^2 - \ \mathcal{P}\ _1 + 1$	$2n\ \mathbf{R}\ _1 - \frac{1}{2}(n^2 + n - 2)$	$2(\ \mathbf{R}\ _1^2 - \ \mathbf{R}\ _2^2 + n\ \mathbf{R}\ _1) - 2n(n-1) + 1$
$\partial_k E(\mathbf{x}_0) \ \& \ \partial_k^2 E(\mathbf{x}_0)$	$2\mathcal{P}_k^2 + 1$	$2R_k + 1$	$2R_k + 1$
$\nabla E(\mathbf{x}_0) \ \& \ \nabla^{\odot 2} E(\mathbf{x}_0)$	$2\ \mathcal{P}\ _2^2 + 1$	$2\ \mathbf{R}\ _1 + 1$	$2\ \mathbf{R}\ _1 + 1$
$\nabla E(\mathbf{x}_0) \ \& \ \nabla^{\otimes 2} E(\mathbf{x}_0)$	$2\ \mathcal{P}\ _1^2 + 1$	$2n\ \mathbf{R}\ _1 - \frac{1}{2}(n^2 - n - 2)$	$2(\ \mathbf{R}\ _1^2 - \ \mathbf{R}\ _2^2 + n\ \mathbf{R}\ _1) - 2n(n-1) + 1$

Table 1: Number of distinct circuit evaluations N_{eval} for measuring combinations of derivatives of a parametrized expectation value function E at parameter position \mathbf{x}_0 . The compared approaches include decomposition of the unitaries together with the original parameter-shift rule (*left*), and the generalized parameter-shift rule Eq. (24) together with the diagonal shift rule for the Hessian in Eq. (36). The requirements for the latter differ significantly for equidistant (*center*) and arbitrary frequencies (*right*, see App. B.2). A third approach is to repeat the general parameter-shift rule, the cost of which can be read off by replacing \mathcal{P} by \mathbf{R} in the left column. Here, n is the number of parameters in the circuit, \mathcal{P}_k is the number of elementary gates with two eigenvalues in the decomposition of the k th parametrized unitary, and R_k denotes the number of frequencies for the k th parameter. The asterisk $(^*)$ indicates that the derivatives $\partial_k^2 E$ and $\partial_m^2 E$ need to be known in order to obtain the mixed derivative at the shown price (see main text). The evaluation numbers take savings into account that are based on using evaluated energies for multiple derivative quantities; hence, they are not additive in general.

halved depth on average (see App. A.3). With either of these methods, the terms for the shift rule in Eq. (36) and thus the metric tensor can be computed via the parameter-shift rule.

The metric can also be computed analytically without parameter shifts via a *linear combination of unitaries (LCU)* [57, 58], which also employs Hadamard tests. As it uses the generator as an operation in the circuit, any non-unitary generator needs to be decomposed into Pauli words for this method to be available on quantum hardware, similar to a gate decomposition. Afterwards, this method uses one circuit evaluation per pair of Pauli words from the k th and m th generator to compute the entry \mathcal{F}_{km} . A modification of all approaches that use a Hadamard test is possible by replacing it with projective measurements [56].

Metric entries that belong to operations that commute *within the circuit*¹⁰ can be computed block-wise without any auxiliary qubits, additional operations or deeper circuits [48]. For a given block, we execute the subcircuit V_1 prior to the group of mutually commuting gates and measure the covariance matrix of the generators $\{G_k\}$ of these gates:

$$\mathcal{F}_{km} = \langle \mathbf{0} | V_1^\dagger G_k G_m V_1 | \mathbf{0} \rangle - \langle \mathbf{0} | V_1^\dagger G_k V_1 | \mathbf{0} \rangle \langle \mathbf{0} | V_1^\dagger G_m V_1 | \mathbf{0} \rangle. \quad (47)$$

By grouping the measurement bases of all $\{G_k G_m\}$

¹⁰For example, operations on distinct wires commute in general but not necessarily within the circuit if entangling operations are carried out between them.

and $\{G_k\}$ of the block, the covariance matrix can typically be measured with only a few unique circuit evaluations¹¹, making this method the best choice for the block-diagonal. One may then either use the result as an approximation to the full metric tensor, or use one of the other methods to compute the off-block-diagonal entries; the approximation has been shown to work well for some circuit structures [48], but not for others [59]. The methods to obtain the metric tensor and their resource requirements are shown in Tab. 2.

Since we run a different circuit for the metric tensor than for the cost function itself, the $2R_k - 1$ evaluations at shifted positions needed for the k th diagonal entry cannot reuse any prior circuit evaluations, as is the case for the cost function Hessian. Consequentially, the natural gradient of a (single term) expectation value function E ,

$$\nabla_n E(\mathbf{x}) := \mathcal{F}^{-1}(\mathbf{x}) \nabla E(\mathbf{x}), \quad (48)$$

with ∇E referring to the Euclidean gradient, requires more circuit evaluations than its Hessian and gradient together.

However, the utility of the metric tensor becomes apparent upon observing that it depends solely on the *ansatz*, and not the observable being measured. This

¹¹For a layer of simultaneous single-qubit rotations on all N qubits, even a single measurement basis is sufficient for the corresponding $N \times N$ block.

	Parameter shift rule		LCU	Covariance
	Overlap	Hadamard		
Aux. qubits	0	1	1	0
off-block-diag.	✓	✓	✓	
Depth (avg)	$\sim \frac{4}{3}D_V$	$\sim \frac{2}{3}D_V$	$\sim \frac{2}{3}D_V$	$\frac{2}{3}D_V$
Depth (max)	$2D_V$	$\sim D_V$	$\sim D_V$	D_V
$N_{\text{eval}}(\mathcal{F}_{kk})$	$\begin{cases} 2R_k - 1 \\ 2R_k \end{cases}$		$\mathcal{Q}_k \leq \frac{1}{2}(\mathcal{P}_k^2 - \mathcal{P}_k)$	$\bar{\mathcal{P}}_k \leq \mathcal{P}_k$
$N_{\text{eval}}(\mathcal{F}_{km})$	$\begin{cases} 2(R_k + R_m) - 1 \\ 2(2R_k R_m + R_k + R_m - 2) \end{cases}$		$\mathcal{P}_k \mathcal{P}_m$	$\bar{\mathcal{P}}_{km} \leq \mathcal{P}_k \mathcal{P}_m$
$N_{\text{eval}}(\mathcal{F})$	$\begin{cases} 2n\ \mathbf{R}\ _1 - \frac{1}{2}(n^2 + n) \\ 2(\ \mathbf{R}\ _1^2 - \ \mathbf{R}\ _2^2 + n(\ \mathbf{R}\ _1 - n + 1)) \end{cases}$		$\frac{1}{2}(\ \mathcal{P}\ _1^2 - \ \mathcal{P}\ _2^2) + \ \mathcal{Q}\ _1$	—

Table 2: Quantum hardware-ready methods to compute the Fubini-Study metric tensor and their resource requirements. The cost function $f(\mathbf{x})$ (see Eq. (45)) for the parameter-shift rule can be implemented with increased depth by applying the adjoint of the original circuit to directly realize the overlap (*left*) or with an auxiliary qubit and Hadamard tests (*center left*, App. A.3). The LCU method (*center right*) is based on Hadamard tests as well and both these methods can spare the auxiliary qubit and instead employ projective measurements [56]. The cheapest method is via measurements of the covariance of generators (*right*) but it can only be used for the block-diagonal of the tensor, i.e., not for all \mathcal{F}_{km} . We denote the depth of the original circuit V by D_V and the number of Pauli words in the decomposition of G_k and its square with \mathcal{P}_k and \mathcal{Q}_k , respectively. The \mathcal{P}_k Pauli words of G_k can be grouped into $\bar{\mathcal{P}}_k$ groups of pairwise commuting words; the number of groups of pairwise commuting Pauli words in the product $G_k G_m$ similarly is $\bar{\mathcal{P}}_{km}$. For the covariance-based approach, we overestimate the number of required circuits, as typically many of the measurement bases of the entries in the same block will be compatible. The number of unique circuits to be evaluated for a diagonal element \mathcal{F}_{kk} , an off-diagonal element \mathcal{F}_{km} , and the full tensor \mathcal{F} is given in terms of the number of frequencies R_k and of \mathcal{Q}_k , \mathcal{P}_k , $\bar{\mathcal{P}}_k$ and $\bar{\mathcal{P}}_{km}$. The entries for N_{eval} in the first and second row of the braces refer to equidistant (main text) and arbitrary frequencies (see App. B.2), respectively.

means that if a cost function has multiple terms, like in VQEs, the metric only needs to be computed once per epoch, rather than once per term, as is the case of the cost function Hessian. Therefore, an epoch of quantum natural gradient descent can be cheaper for such cost functions than an epoch of optimizers using the Hessian of the cost function. In addition, the block-diagonal of the metric tensor can be obtained with few circuit evaluations per block for conventional gates without any further requirements and with reduced average circuit depth.

5 Applications

In this section, we will present QAOA as concrete application for our general parameter-shift rule, which reduces the required resources significantly when computing derivatives. Afterwards, we use the approach of trigonometric interpolation to generalize the Rotosolve algorithm. This makes it applicable to arbitrary quantum gates with equidistant frequencies, which reproduces the results in Refs. [42, 45], and extends them further to more general frequency spectra. In addition, we make quantum analytic descent (QAD) available for arbitrary quantum gates with equidistant frequencies, which previously required a higher-dimensional Fourier reconstruction and thus was infeasible.

5.1 QAOA and Hamiltonian time evolution

In Eq. (24) we presented a generalized parameter-shift rule that makes use of $2R$ function evaluations for R frequencies in E . A particular example for single-parameter unitaries with many frequencies are layers of single- or two-qubit rotation gates, as can be found e.g., in QAOA circuits or digitized Hamiltonian time evolution algorithms.

The quantum approximate optimization algorithm (QAOA) was first proposed in 2014 by Farhi, Goldstone and Gutmann to solve classical combinatorial optimization problems on near-term quantum devices [8]. Since then, it has been investigated analytically [60, 61, 62], numerically [63, 64], and on quantum computers [65, 66].

In general, given a problem Hamiltonian H_P that encodes the solution to the problem of interest onto N qubits, QAOA applies two types of layers alternately to an initial state $|+\rangle^{\otimes N}$:

$$V_{\text{QAOA}}(\mathbf{x}) = \prod_{j=p}^1 U_M(x_{2j}) U_P(x_{2j-1}), \quad (49)$$

where p is the number of blocks which determines the depth of the circuit, $U_M(x) = \exp(-ixH_M)$ with $H_M = \sum_{k=1}^N X_k$ is the so-called *mixing layer*, and $U_P(x) = \exp(-ixH_P)$ is the time evolution under H_P . The parameters \mathbf{x} can then be optimized to try to

minimize the objective function

$$E(\mathbf{x}) = \langle + |^{\otimes N} V_{\text{QAOA}}^\dagger(\mathbf{x}) H_P V_{\text{QAOA}}(\mathbf{x}) | + \rangle^{\otimes N}. \quad (50)$$

Here we focus on the layer U_P , and we look at the example of MAXCUT in particular. The corresponding problem Hamiltonian for an unweighted graph $G = (\mathcal{V}, \mathcal{E})$ with N vertices \mathcal{V} and M edges \mathcal{E} reads

$$H_P = \sum_{(a,b) \in \mathcal{E}} \frac{1}{2} (1 - Z_a Z_b), \quad (51)$$

and U_P correspondingly contains M two-qubit Pauli- Z rotations R_{ZZ} .

We note that H_M has eigenvalues $-N, -N + 2, \dots, N$, which means the corresponding frequencies (differences of eigenvalues) are $2, \dots, 2N$. Thus, treating $U_M(x_{2j})$ as a single operation, Eq. (6) implies that $E(\mathbf{x})$ can be considered an N -order trigonometric polynomial in x_{2j} , and the parameter-shift rules we derive in Sec. 3 will apply with $R = N$. Similarly, H_P has corresponding frequencies in the set $[M]$, and it will obey the parameter-shift rule for $R = M$, although we may be able to give better upper bounds λ for R . Thus the unique positive differences $\{\Omega_\ell\}$ for those layers, i.e., the frequencies of $E(\mathbf{x})$ with respect to the parameter $\{x_{2j-1}\}_{j \in [p]}$, take integer values within the interval $[0, \lambda]$ as well. We may therefore use Eq. (24), with the knowledge that $R \leq \lambda \leq M$.

Note that knowing *all* frequencies of $E(x)$ requires knowledge of the full spectrum of H_P — and in particular of λ — which in turn is the solution of MAXCUT itself. As a consequence, the motivation for performing QAOA becomes obsolete. Therefore, in general we cannot assume to know $\{\Omega_\ell\}$ (or even R), but instead require upper bounds $\varphi(G) \geq \text{MAXCUT}(G) = \lambda$ which can be used to bound the largest frequency, and thus the number of frequencies R and subsequently the number of terms in the parameter-shift rule. It is noteworthy that even if the *largest* frequency λ is known exactly via a tight bound — which restricts the Fourier spectrum to the integers $[\lambda]$ — not *all* integers smaller than λ need to be present in the set of frequencies $\{\Omega_\ell\}$, so that the estimate for R may be too large¹².

One way to obtain an upper bound uses analytic results based on the Laplacian of the graph of interest [67, 68], for which automatic bound generating programs exist [69]. An alternative approach uses semi-definite programs (SDPs) that solve relaxations of the MAXCUT problem, the most prominent being the *Goemans-Williamson (GW)* algorithm [70] and recent extensions thereof that provide tighter upper bounds [71, 72]. The largest eigenvalue is guaranteed to be within ~ 0.878 of these SDP upper bounds.

¹²A simple example for this is the case of $2k$ -regular graphs; here, H_P only has even eigenvalues, and therefore all frequencies are even as well. Given an upper bound φ , we thus know the number of frequencies to satisfy $R \leq \varphi/2$.

To demonstrate the above strategy, we summarize the number of evaluations required for the gradient and Hessian of an n -parameter QAOA circuit on N qubits for MAXCUT in Tab. 3, comparing the approach via decomposing the circuit, to the one detailed above based on φ and the improved Hessian measurement scheme in Sec. 4.1. Here, we take into account that half of the layers are of the form U_P , and the other half are mixing layers with $R = N$ frequencies. We systematically observe the number of evaluations for the gradient to be cut in half, and the those for the gradient and Hessian together to scale with halved order in N (and k , for regular graphs).

In addition, we display the numbers of circuit evaluations from Tab. 3 together with SDP-based bounds for λ and the true minimal number of evaluations required for the parameter-shift rule in Fig. 3. For this, we sampled random unweighted graphs of the corresponding type and size and ran the GW algorithm as well as an improvement thereof to obtain tighter bounds [71]. On one hand we observe the advantage of the generalized parameter-shift rule and the cheaper Hessian method that can be read off already from the scalings in Tab. 3. On the other hand, we find both SDP-based upper bounds to provide an exact estimate of the largest eigenvalue in the $N \leq 20$ regime, as can be seen from the cut values obtained from the GW algorithm that coincide with the upper bound. In cases in which the frequencies $\{\Omega_\ell\}$ occupy all integers in $[R]$, this leads to an exact estimate of R and the evaluations in the shift rule. For all graph types but complete graphs, the SDP-based upper bounds yield a better estimate for the number of terms than the respective analytic bound φ , which improves the generalized shift rule further.

In summary, we find the generalized parameter-shift rule to offer a constant prefactor improvement when computing the gradient and an improvement of at least $\mathcal{O}(N)$ when computing both the gradient and the Hessian. For certain graph types, knowledge about the structure of the spectrum and tight analytic bounds provide this advantage already, whereas for other graph types the SDP-based bounds reduce the evaluation numbers significantly.

5.2 Rotosolve

The *Rotosolve* algorithm is a coordinate descent algorithm for minimizing quantum cost functions. It has been independently discovered multiple times [42, 45, 51, 50], with [50] giving the algorithm its name but only (along with [51]) considering parametrized Pauli rotations, and [42, 45] covering other unitaries with integer-valued generator eigenvalues.

The Rotosolve algorithm optimizes the rotation angles sequentially: for one variational parameter x_k at a time, the cost function is reconstructed as a function of that parameter using $2R_k + 1$ evaluations, the mini-

Graph type	Decomposition-based		Gen. shift rule		
	∇E	$\nabla E \& \nabla^{\otimes 2} E$	Bound φ	∇E	$\nabla E \& \nabla^{\otimes 2} E$
General	$(M + N)n$	$\mathcal{O}(n^2(M + N)^2)$	φ	$n(\varphi + N)$	$\mathcal{O}(n^2(\varphi + N))$
Complete	$\frac{1}{2}n(N^2 + N)$	$\mathcal{O}(n^2N^4)$	$\lfloor \frac{N^2}{4} \rfloor$	$n\left(\lfloor \frac{N^2}{4} \rfloor + N\right)$	$\mathcal{O}(n^2N^2)$
$2k$ -regular	$(k + 1)nN$	$\mathcal{O}(k^2n^2N^2)$	kN	$\frac{k+2}{2}nN$	$\mathcal{O}(kn^2N)$
$(2k+1)$ -regular	$\frac{2k+3}{2}nN$	$\mathcal{O}(k^2n^2N^2)$	$\frac{2k+1}{2}N$	$\frac{2k+3}{2}nN$	$\mathcal{O}(kn^2N)$

Table 3: Evaluation numbers for the gradient, or both the gradient and the Hessian, for QAOA circuits for MAXCUT on several types of graphs. Each graph has N vertices and a graph type-specific number M of edges, and the (even) number of parameters is denoted as n . For K -regular graphs, we know $M = \min\{(N^2 - N)/2, KN/2\}$, and the latter value is used in the displayed evaluation costs; if the former value forms the minimum, the graph is in fact complete. The left column is based on decomposing the circuit, applying the conventional two-term parameter-shift rule per elementary gate and iterating it for the Hessian. The right column employs the generalized parameter-shift rule Eq. (24) combined with an upper bound φ for the largest eigenvalue λ of the problem Hamiltonian, as well as the reduced number of evaluations for Hessian off-diagonal terms from Sec. 4.1. The bound φ for complete graphs can be found in Ref. [67].

mum of the reconstruction is calculated, and then the parameter is updated to the minimizing angle. For the case of Pauli rotation gates this minimum can be found via a closed-form expression. Recent studies have shown such coordinate descent methods to work well on many tasks [73, 50, 45, 74], although there are limited cases where these methods fail [75].

While Rotosolve is not gradient-based, our cost reduction for the gradient presented in Sec. 5.1 stems from a cost reduction for function reconstruction, and hence is applicable to Rotosolve as well.

As shown in Sec. 3.1, the univariate objective function can also be fully reconstructed if the parametrized unitaries are more complicated than Pauli rotations, using the function value itself and the evaluations from the generalized parameter-shift rule. Since the generalized parameter-shift rule also applies for non-equidistant frequencies (see App. B), the reconstruction works in the same way for arbitrary single-parameter gates. This extends our generalization of Rotosolve beyond the previously known integer-frequency case [42, 45], although the number of frequencies—and thus the cost—for the reconstruction are typically significantly increased for non-integer frequencies. While the minimizing angle might not be straightforward to express in a closed form as it is the case for a single frequency, the one-dimensional minimization can efficiently be carried out numerically to high precision, via grid search or semi-definite programming [76, Chapter 4.2].

5.3 Quantum analytic descent

Quantum analytic descent (QAD) [49] also approaches the optimization problem in VQAs via trigonometric interpolation. In contrast to Rotosolve, it considers a model of all parameters *simultaneously* and includes second-order derivatives, but this model only is a *local approximation* of the full cost function. Additionally, QAD has been developed for circuits that exclusively contain Pauli rotations as

parametrized gates.

The algorithm evaluates the cost function E at $2n^2 + n + 1$ points around a reference point \mathbf{x}_0 , and then constructs a trigonometric model of the form¹³

$$\begin{aligned} \hat{E}(\mathbf{x}_0 + \mathbf{x}) = & A(\mathbf{x}) \left[E^{(A)} + 2\mathbf{E}^{(B)} \cdot \tan\left(\frac{\mathbf{x}}{2}\right) \right. \\ & + 2\mathbf{E}^{(C)} \cdot \tan\left(\frac{\mathbf{x}}{2}\right)^{\odot 2} \\ & \left. + 4 \tan\left(\frac{\mathbf{x}}{2}\right) \cdot E^{(D)} \cdot \tan\left(\frac{\mathbf{x}}{2}\right) \right], \end{aligned} \quad (52)$$

Here, we introduced $A(\mathbf{x}) := \prod_k \cos^2\left(\frac{x_k}{2}\right)$ and the element-wise square of a vector \mathbf{v} , $(\mathbf{v}^{\odot 2})_k := v_k^2$ as for the Hessian diagonal. The coefficients $E^{(A/B/C/D)}$ are derived from the circuit evaluations, taking the form of a scalar, two vectors and an upper triangular matrix. More precisely, the expansion basis is chosen such that $\mathbf{E}^{(B)} = \nabla E(\mathbf{x}_0)$, $\mathbf{E}^{(C)} = \nabla^{\odot 2} E(\mathbf{x}_0)$, and $E^{(D)}$ is the strictly upper triangular part of the Hessian. Note that for this model $2n^2 + n + 1$ evaluations are used to obtain $n^2/2 + 3n/2 + 1$ parameters. In the presence of statistical noise from these evaluations, it turns out that building the model to a desired precision and inferring modelled gradients close to the reference point \mathbf{x}_0 has resource requirements similar to measuring the gradient directly [49].

This model coincides with $E(\mathbf{x})$ at \mathbf{x}_0 up to second order, and in the vicinity its error scales with the third order of the largest parameter deviation [49]. After the construction phase, the model cost is minimized in an inner optimization loop, which only requires classical operations. For an implementation and demonstration of the optimization, we also refer the reader to [77] and [78].

In the light of the parameter-shift rules and reconstruction methods, we propose three (alternative) modifications of QAD. The first change is to reduce the required number of evaluations. As the coeffi-

¹³We slightly modify the trigonometric basis functions from Ref. [49] to have leading order coefficients 1.

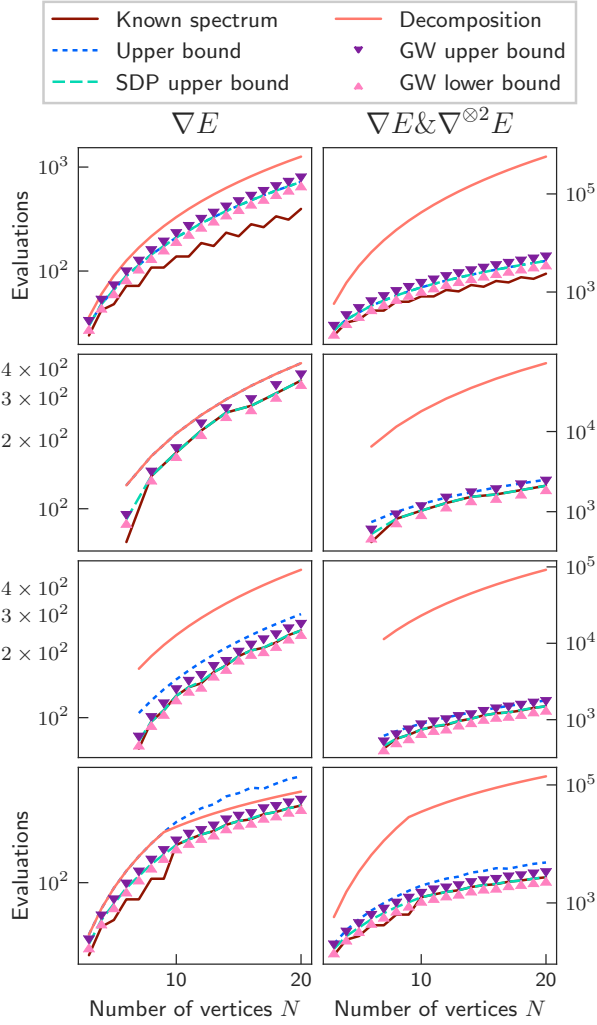


Figure 3: Evaluation numbers N_{eval} for the gradient (left) or both the gradient and the Hessian (right) for $n = 6$ parameter QAOA circuits for MAXCUT on graphs of several types and sizes. Using numerical upper bounds together with our new parameter-shift rule (GW – purple triangles and its generalization – dashed turquoise) reduces the resource requirements for both quantities significantly, compared to the previously available decomposition-based method (solid orange). The rows correspond to the various considered graph types (top to bottom): complete, 5-regular, 6-regular and (up to) $4N$ randomly sampled edges. The requirements for the decomposition-based approach and the analytic upper bound (dotted blue) correspond to the results in the left and right column of Tab. 3, respectively. The numerical upper bounds both use the minimized objective value of SDPs for relaxations of MAXCUT to obtain the bound φ , which depends on the graph instance. The GW-based lower bound (pink triangles) is obtained by randomly mapping the output state of the GW algorithm to 10 valid cuts and choosing the one with the largest cut value. Note that K -regular graphs are only defined for $N > K$ and $NK \bmod 2 = 0$ and that graphs with κN sampled edges are complete for $N \leq 2\kappa + 1$, leading to a change in the qualitative behaviour in the last row at $N = 2\kappa + 2 = 10$.

cients $E^{(A/B/C/D)}$ consist of the gradient and Hessian, they allow us to exploit the reduced resource requirements presented in Tab. 1¹⁴. In the case originally considered by the authors, i.e., for Pauli rotations only, this reduces the number of evaluations from $2n^2 + n + 1$ to $(3n^2 + n)/2 + 1$.

A second, alternative modification of QAD is to keep all evaluations as originally proposed to obtain the full second-order terms, i.e., we may combine the shift angles for each pair of parameters, and use them for coefficients of additional higher-order terms. This extended model (see App. D.1) has the form

$$\begin{aligned} \hat{E}(\mathbf{x}_0 + \mathbf{x}) &= \hat{E}(\mathbf{x}_0 + \mathbf{x}) + 4A(\mathbf{x}) \tan\left(\frac{\mathbf{x}}{2}\right)^{\odot 2} \\ &\cdot \left[E^{(F)} \cdot \tan\left(\frac{\mathbf{x}}{2}\right) + E^{(G)} \cdot \tan\left(\frac{\mathbf{x}}{2}\right)^{\odot 2} \right], \end{aligned} \quad (53)$$

where $E^{(F)}$ is symmetric with zeros on its diagonal and $E^{(G)}$ is a strictly upper triangular matrix. This extended model has $2n^2 + 1$ degrees of freedom, which matches the number of evaluations exactly.

While the QAD model reconstructs the univariate restrictions of E to the coordinate axes correctly, the extended model \hat{E} does so for the bivariate restrictions to the plane spanned by any pair of coordinate axes. It remains to investigate whether and for which applications the extension yields a better optimization behaviour; for functions in which pairs of parameters yield a good local approximation of the landscape, it might provide an improvement.

The third modification we consider is to generalize the previous, extended QAD model to *general* single-parameter quantum gates. This can be done via a full trigonometric interpolation to second order, which is detailed in App. D.2, exactly reconstructing the energy function when restricted to any coordinate plane at the price of $2(\|\mathbf{R}\|_1^2 - \|\mathbf{R}\|_2^2 + \|\mathbf{R}\|_1) + 1$ evaluations.

Using toy model circuits and Hamiltonians, we demonstrate the qualitative difference between the QAD model, its extension \hat{E} , and the generalization to multiple frequencies in Fig. 4.

6 Discussion

In this work, we derive interpolation rules to exactly express quantum functions $E(x)$ as a linear combination of evaluations $E(x_\mu)$, assuming $E(x)$ derives from parametrized gates of the form $U(x) = \exp(ixG)$. Our method relies on the observation that $E(x)$ can be expressed as trigonometric polynomial in x , characterized by a set of R frequencies that correspond to distinct differences in the eigenvalues of G . This observation allows us to derive our results using trigonometric interpolation methods.

¹⁴In addition, we may skip the n evaluations with shift angle π proposed in Ref. [49], and instead measure the Hessian diagonal as discussed in Sec. 4.1.

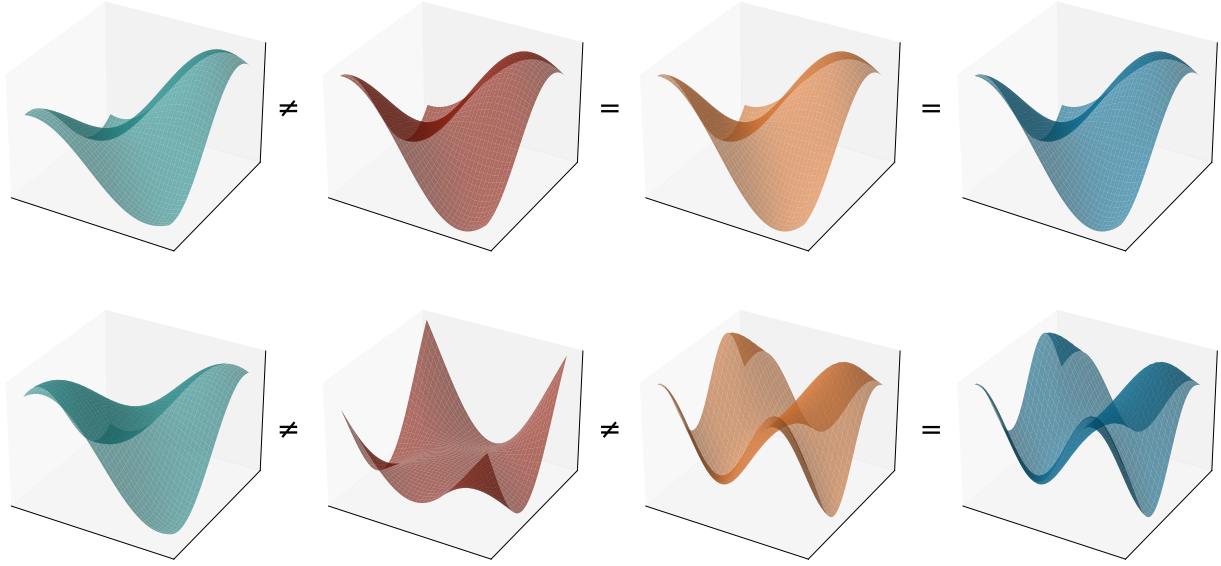


Figure 4: The QAD model (*left*), its extension \hat{E} , see Eq. (53), that includes full second-order terms (*center left*), and the second-order trigonometric interpolation model (*center right*), as well as the original expectation value E (*right*). The original function is generated from toy Hamiltonians in a two-parameter example circuit, with one frequency (*top*) and two frequencies (*bottom*) per parameter. The QAD model produces a local approximation to E that deviates away from x_0 at a slow rate for $R = 1$ but faster for $R = 2$. The extension \hat{E} reuses evaluations made for the Hessian to capture the full bivariate dependence for a single frequency but is not apt to model multiple frequencies either. Finally, the trigonometric interpolation generalizes \hat{E} . This means it coincides with \hat{E} for $R = 1$, but also reproduces the full bivariate function for $R > 1$.

In addition to a full reconstruction of $E(x)$, the presented approach offers parameter-shift rules for derivatives of arbitrary order and recipes to evaluate multivariate derivatives more cheaply. Using the concept of the stochastic parameter-shift rule, quantum gates of the form $U_F(x) = \exp(i(xG + F))$ can be differentiated as well.

Nevertheless, much remains unknown about the practicality of our new parameter-shift rules. For the common case that we have R equidistant frequencies, Sec. 3.5 shows that the scaling of the required resources is similar between naively applying our generalized parameter-shift rules, and applying parameter-shift rules to a decomposition of $U(x)$. This holds for the first derivative and also for the required shot budget when computing the second derivative, whereas the number of unique circuits is significantly smaller for the new, generalized shift rule.

Our observations lead to several open questions:

- In which situations can we obtain better bounds on the number of frequencies? We investigated an example for QAOA in Sec. 5.1, but are there other examples?
- For general G (e.g., $G = \sum_j c_j P_j$ with real c_j and Pauli words P_j), the frequencies will not be equidistant, and in fact R may scale quadratically in the size of U . Naively applied, our method would then scale poorly compared to decomposing G . Can we

apply an approximate or stochastic parameter-shift rule with a better scaling?

- Would it ever make sense to *truncate* these parameter-shift rules to keep only terms corresponding to smaller frequencies? This is inspired by the idea of using low-pass filters to smooth out rapid changes of a signal.
- Our work on function reconstruction extends QAD to all gates with equidistant frequencies. Similarly, it allows Rotosolve, which has been shown to work remarkably well on some applications, to be used on all quantum gates with arbitrary frequencies. Is there a classification of problems on which these model-based algorithms work well? And can we reduce the optimization cost based on the above ideas?
- More generally, can we apply the machinery of Fourier analysis more broadly, e.g., to improve optimization methods in the presence of noise?

We hope that this work serves as an impetus for future work that will further apply signal processing methods to the burgeoning field of variational quantum computing.

Acknowledgements

We would like to thank Nathan Killoran, Maria Schuld, Matthew Beach, and Eric Kessler for helpful comments on the manuscript, as well as Christian Gogolin and Gian-Luca Anselmetti for valuable discussions.

Code availability

The scripts used to create the data and plots for Figs. 3 and 4 can be found at [79].

References

- [1] Amazon Web Services. “Amazon Braket”. url: aws.amazon.com/braket/.
- [2] J.M. Arrazola, V. Bergholm, K. Brádler, T.R. Bromley, M.J. Collins, I. Dhand, A. Fumagalli, T. Gerrits, A. Goussev, L.G. Helt, J. Hundal, T. Isacsson, R.B. Israel, J. Izaac, S. Jangiri, R. Janik, N. Killoran, S.P. Kumar, J. Lavoie, A.E. Lita, D.H. Mahler, M. Menotti, B. Morrison, S.W. Nam, L. Neuhaus, H.Y. Qi, N. Quesada, A. Reppingon, K.K. Sabapathy, M. Schuld, D. Su, J. Swinerton, A. Száva, K. Tan, P. Tan, V.D. Vaidya, Z. Vernon, Z. Zabaneh, and Y. Zhang. “Quantum circuits with many photons on a programmable nanophotonic chip”. *Nature* **591**, 54–60 (2021).
- [3] IBM Corporation. “IBM Quantum”. url: quantum-computing.ibm.com/.
- [4] Microsoft. “Azure Quantum”. url: azure.microsoft.com/./quantum/.
- [5] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. “Parameterized quantum circuits as machine learning models”. *Quantum Science and Technology* **4**, 043001 (2019).
- [6] Marco Cerezo, Andrew Arrasmith, Ryan Babush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. “Variational quantum algorithms”. *Nature Reviews Physics* **3**, 625–644 (2021).
- [7] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O’Brien. “A variational eigenvalue solver on a photonic quantum processor”. *Nature Communications* **5**, 4213 (2014).
- [8] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. “A quantum approximate optimization algorithm” (2014). [arXiv:1411.4028](https://arxiv.org/abs/1411.4028).
- [9] Tyson Jones, Suguru Endo, Sam McArdle, Xiao Yuan, and Simon C. Benjamin. “Variational quantum algorithms for discovering Hamiltonian spectra”. *Phys. Rev. A* **99**, 062304 (2019).
- [10] Gian-Luca R. Anselmetti, David Wierichs, Christian Gogolin, and Robert M Parrish. “Local, expressive, quantum-number-preserving VQE ansätze for fermionic systems”. *New Journal of Physics* **23**, 113010 (2021).
- [11] Harper R. Grimsley, Sophia E. Economou, Edwin Barnes, and Nicholas J. Mayhall. “An adaptive variational algorithm for exact molecular simulations on a quantum computer”. *Nature communications* **10**, 1–9 (2019).
- [12] Ken M. Nakanishi, Kosuke Mitarai, and Keisuke Fujii. “Subspace-search variational quantum eigensolver for excited states”. *Phys. Rev. Research* **1**, 033062 (2019).
- [13] Alain Delgado, Juan Miguel Arrazola, Soran Jangiri, Zeyue Niu, Josh Izaac, Chase Roberts, and Nathan Killoran. “Variational quantum algorithm for molecular geometry optimization”. *Phys. Rev. A* **104**, 052402 (2021).
- [14] Eric Anschuetz, Jonathan Olson, Alán Aspuru-Guzik, and Yudong Cao. “Variational quantum factoring”. In *International Workshop on Quantum Technology and Optimization Problems*. Pages 74–85. Springer (2019).
- [15] Sumeet Khatri, Ryan LaRose, Alexander Poremba, Lukasz Cincio, Andrew T. Sornborger, and Patrick J. Coles. “Quantum-assisted quantum compiling”. *Quantum* **3**, 140 (2019).
- [16] Jun Li, Xiaodong Yang, Xinhua Peng, and Chang-Pu Sun. “Hybrid quantum-classical approach to quantum optimal control”. *Phys. Rev. Lett.* **118**, 150503 (2017).
- [17] Ryan LaRose, Arkin Tikku, Étude O’Neel-Judy, Lukasz Cincio, and Patrick J. Coles. “Variational quantum state diagonalization”. *npj Quantum Information* **5**, 1–10 (2019).
- [18] Benjamin Commeau, Marco Cerezo, Zoë Holmes, Lukasz Cincio, Patrick J. Coles, and Andrew Sornborger. “Variational Hamiltonian diagonalization for dynamical quantum simulation” (2020). [arXiv:2009.02559](https://arxiv.org/abs/2009.02559).
- [19] Jonathan Romero, Jonathan P. Olson, and Alan Aspuru-Guzik. “Quantum autoencoders for efficient compression of quantum data”. *Quantum Science and Technology* **2**, 045001 (2017).
- [20] Guillaume Verdon, Michael Broughton, and Jacob Biamonte. “A quantum algorithm to train neural networks using low-depth circuits” (2017). [arXiv:1712.05304](https://arxiv.org/abs/1712.05304).

- [21] Edward Farhi and Hartmut Neven. “Classification with quantum neural networks on near term processors” (2018). [arXiv:1802.06002](#).
- [22] Maria Schuld and Nathan Killoran. “Quantum machine learning in feature Hilbert spaces”. *Phys. Rev. Lett.* **122**, 040504 (2019).
- [23] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. “Quantum circuit learning”. *Phys. Rev. A* **98**, 032309 (2018).
- [24] Maria Schuld, Alex Bocharov, Krysta M. Svore, and Nathan Wiebe. “Circuit-centric quantum classifiers”. *Phys. Rev. A* **101**, 032308 (2020).
- [25] Edward Grant, Marcello Benedetti, Shuxiang Cao, Andrew Hallam, Joshua Lockhart, Vid Stojevic, Andrew G. Green, and Simone Severini. “Hierarchical quantum classifiers”. *npj Quantum Information* **4**, 1–8 (2018).
- [26] Jin-Guo Liu and Lei Wang. “Differentiable learning of quantum circuit Born machines”. *Phys. Rev. A* **98**, 062324 (2018).
- [27] Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. “Supervised learning with quantum-enhanced feature spaces”. *Nature* **567**, 209–212 (2019).
- [28] Hongxiang Chen, Leonard Wossnig, Simone Severini, Hartmut Neven, and Masoud Mohseni. “Universal discriminative quantum neural networks”. *Quantum Machine Intelligence* **3**, 1–11 (2021).
- [29] Nathan Killoran, Thomas R. Bromley, Juan Miguel Arrazola, Maria Schuld, Nicolás Quesada, and Seth Lloyd. “Continuous-variable quantum neural networks”. *Phys. Rev. Research* **1**, 033063 (2019).
- [30] Gregory R. Steinbrecher, Jonathan P. Olson, Dirk Englund, and Jacques Carolan. “Quantum optical neural networks”. *npj Quantum Information* **5**, 1–9 (2019).
- [31] Andrea Mari, Thomas R. Bromley, Josh Izaac, Maria Schuld, and Nathan Killoran. “Transfer learning in hybrid classical-quantum neural networks”. *Quantum* **4**, 340 (2020).
- [32] Ryan Sweke, Frederik Wilde, Johannes Meyer, Maria Schuld, Paul K. Faehrmann, Barthélémy Meynard-Piganeau, and Jens Eisert. “Stochastic gradient descent for hybrid quantum-classical optimization”. *Quantum* **4**, 314 (2020).
- [33] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. “TensorFlow: a system for large-scale machine learning”. In OSDI. Volume 16, pages 265–283. Berkeley, CA, USA (2016). USENIX Association. url: [dl.acm.org/..3026877.3026899](#).
- [34] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. “Automatic differentiation in PyTorch”. NIPS 2017 Workshop Autodiff (2017). url: [openreview.net/forum?id=BJJsrnfCZ](#).
- [35] Dougal Maclaurin, David Duvenaud, and Ryan P. Adams. “Autograd: Effortless gradients in NumPy”. In ICML 2015 AutoML Workshop. (2015). url: [indico.ijclab.in2p3.fr/..](#)
- [36] Atılım Güneş Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. “Automatic differentiation in machine learning: a survey”. *Journal of Machine Learning Research* **18**, 1–153 (2018). url: [http://jmlr.org/papers/v18/17-468.html](#).
- [37] Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, M. Sohaib Alam, Shahnawaz Ahmed, Juan Miguel Arrazola, Carsten Blank, Alain Delgado, Soran Jahangiri, Keri McKiernan, Johannes Jakob Meyer, Zeyue Niu, Antal Száva, and Nathan Killoran. “PennyLane: Automatic differentiation of hybrid quantum-classical computations” (2020). [arXiv:1811.04968](#).
- [38] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. “Evaluating analytic gradients on quantum hardware”. *Phys. Rev. A* **99**, 032331 (2019).
- [39] Leonardo Banchi and Gavin E. Crooks. “Measuring analytic gradients of general quantum evolution with the stochastic parameter shift rule”. *Quantum* **5**, 386 (2021).
- [40] Gavin E. Crooks. “Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition” (2019). [arXiv:1905.13311](#).
- [41] Jakob S. Kottmann, Abhinav Anand, and Alán Aspuru-Guzik. “A feasible approach for automatically differentiable unitary coupled-cluster on quantum computers”. *Chemical Science* **12**, 3497–3508 (2021).
- [42] Javier Gil Vidal and Dirk Oliver Theis. “Calculus on parameterized quantum circuits” (2018). [arXiv:1812.06323](#).
- [43] Francisco Javier Gil Vidal and Dirk Oliver Theis. “Input redundancy for parameterized quantum circuits”. *Frontiers in Physics* **8**, 297 (2020).
- [44] Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. “Effect of data encoding on the expressive

- power of variational quantum-machine-learning models”. *Phys. Rev. A* **103**, 032430 (2021).
- [45] Ken M. Nakanishi, Keisuke Fujii, and Syngae Todo. “Sequential minimal optimization for quantum-classical hybrid algorithms”. *Phys. Rev. Research* **2**, 043158 (2020).
- [46] Andrea Mari, Thomas R. Bromley, and Nathan Killoran. “Estimating the gradient and higher-order derivatives on quantum hardware”. *Phys. Rev. A* **103**, 012405 (2021).
- [47] Johannes Jakob Meyer. “Fisher information in noisy intermediate-scale quantum applications”. *Quantum* **5**, 539 (2021).
- [48] James Stokes, Josh Izaac, Nathan Killoran, and Giuseppe Carleo. “Quantum natural gradient”. *Quantum* **4**, 269 (2020).
- [49] Bálint Koczor and Simon C. Benjamin. “Quantum analytic descent” (2020). [arXiv:2008.13774](https://arxiv.org/abs/2008.13774).
- [50] Mateusz Ostaszewski, Edward Grant, and Marcello Benedetti. “Structure optimization for parameterized quantum circuits”. *Quantum* **5**, 391 (2021).
- [51] Robert M. Parrish, Joseph T. Iosue, Asier Ozaeta, and Peter L. McMahon. “A Jacobi diagonalization and Anderson acceleration algorithm for variational quantum algorithm parameter optimization” (2019). [arXiv:1904.03206](https://arxiv.org/abs/1904.03206).
- [52] Artur F. Izmaylov, Robert A. Lang, and Tzu-Ching Yen. “Analytic gradients in variational quantum algorithms: Algebraic extensions of the parameter-shift rule to general unitary transformations”. *Phys. Rev. A* **104**, 062443 (2021).
- [53] Oleksandr Kyriienko and Vincent E. Elfving. “Generalized quantum circuit differentiation rules”. *Phys. Rev. A* **104**, 052417 (2021).
- [54] Thomas Hubregtzen, Frederik Wilde, Shozab Qasim, and Jens Eisert. “Single-component gradient rules for variational quantum algorithms” (2021). [arXiv:2106.01388v1](https://arxiv.org/abs/2106.01388v1).
- [55] Antoni Zygmund. “Trigonometric series, Volume II”. *Cambridge University Press* (1988).
- [56] Kosuke Mitarai and Keisuke Fujii. “Methodology for replacing indirect measurements with direct measurements”. *Phys. Rev. Research* **1**, 013006 (2019).
- [57] Sam McArdle, Tyson Jones, Suguru Endo, Ying Li, Simon C. Benjamin, and Xiao Yuan. “Variational ansatz-based quantum simulation of imaginary time evolution”. *npj Quantum Information* **5** (2019).
- [58] Ying Li and Simon C. Benjamin. “Efficient variational quantum simulator incorporating active error minimization”. *Phys. Rev. X* **7**, 021050 (2017).
- [59] David Wierichs, Christian Gogolin, and Michael Kastoryano. “Avoiding local minima in variational quantum eigensolvers with the natural gradient optimizer”. *Phys. Rev. Research* **2**, 043246 (2020).
- [60] Mauro E. S. Morales, Jacob D. Biamonte, and Zoltán Zimborás. “On the universality of the quantum approximate optimization algorithm”. *Quantum Information Processing* **19**, 1–26 (2020).
- [61] Seth Lloyd. “Quantum approximate optimization is computationally universal” (2018). [arXiv:1812.11075](https://arxiv.org/abs/1812.11075).
- [62] Matthew B. Hastings. “Classical and quantum bounded depth approximation algorithms” (2019). [arXiv:1905.07047](https://arxiv.org/abs/1905.07047).
- [63] Zhihui Wang, Stuart Hadfield, Zhang Jiang, and Eleanor G. Rieffel. “Quantum approximate optimization algorithm for MaxCut: A fermionic view”. *Phys. Rev. A* **97**, 022304 (2018).
- [64] Wen Wei Ho and Timothy H. Hsieh. “Efficient variational simulation of non-trivial quantum states”. *SciPost Phys* **6**, 29 (2019).
- [65] Leo Zhou, Sheng-Tao Wang, Soonwon Choi, Hannes Pichler, and Mikhail D. Lukin. “Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices”. *Phys. Rev. X* **10**, 021067 (2020).
- [66] Matthew P. Harrigan, Kevin J. Sung, Matthew Neeley, Kevin J. Satzinger, Frank Arute, Kunal Arya, Juan Atalaya, Joseph C. Bardin, Rami Barends, Sergio Boixo, et al. “Quantum approximate optimization of non-planar graph problems on a planar superconducting processor”. *Nature Physics* **17**, 332–336 (2021).
- [67] Charles Delorme and Svatopluk Poljak. “The performance of an eigenvalue bound on the Max-Cut problem in some classes of graphs”. *Discrete Mathematics* **111**, 145–156 (1993).
- [68] William N. Anderson Jr. and Thomas D. Morley. “Eigenvalues of the Laplacian of a graph”. *Linear and Multilinear Algebra* **18**, 141–145 (1985).
- [69] Vladimir Brankov, Pierre Hansen, and Dragan Stevanović. “Automated conjectures on upper bounds for the largest Laplacian eigenvalue of graphs”. *Linear Algebra and its Applications* **414**, 407–424 (2006).
- [70] Michel X. Goemans and David P. Williamson. “Improved approximation algorithms for Maximum Cut and satisfiability problems using semidefinite programming”. *J. ACM* **42**, 1115–1145 (1995).

- [71] Miguel F. Anjos and Henry Wolkowicz. “Geometry of semidefinite MaxCut relaxations via matrix ranks”. *Journal of Combinatorial Optimization* **6**, 237–270 (2002).
- [72] Liu Hongwei, Sanyang Liu, and Fengmin Xu. “A tight semidefinite relaxation of the MaxCut problem”. *J. Comb. Optim.* **7**, 237–245 (2003).
- [73] Andrea Skolik, Jarrod R. McClean, Masoud Mohseni, Patrick van der Smagt, and Martin Leib. “Layerwise learning for quantum neural networks”. *Quantum Machine Intelligence* **3**, 1–11 (2021).
- [74] Marcello Benedetti, Mattia Fiorentini, and Michael Lubasch. “Hardware-efficient variational quantum algorithms for time evolution”. *Phys. Rev. Research* **3**, 033083 (2021).
- [75] Ernesto Campos, Aly Nasrallah, and Jacob Biamonte. “Abrupt transitions in variational quantum circuit training”. *Phys. Rev. A* **103**, 032607 (2021).
- [76] Aharon Ben-Tal and Arkadi Nemirovski. “Lectures on modern convex optimization: Analysis, algorithms, and engineering applications”. *SIAM* (2001).
- [77] Elies Gil-Fuster and David Wierichs. “Quantum analytic descent (demo)”. url: [penny-lane.ai/qml/demos/..](http://penny-lane.ai/qml/demos/) (accessed: 2022-01-23).
- [78] Bálint Koczor (2021). code: [balintkoczor/quantum-analytic-descent](https://github.com/balintkoczor/quantum-analytic-descent).
- [79] David Wierichs, Josh Izaac, Cody Wang, and Cedric Yen-Yu Lin (2022). code: [dwierichs/General-Parameter-Shift-Rules](https://github.com/dwierichs/General-Parameter-Shift-Rules).
- [80] Leonard Benjamin William Jolley. “Summation of series”. *Dover Publications* (1961).
- [81] falagar. “Prove that $\sum_{k=1}^{n-1} \tan^2 \frac{k\pi}{2n} = \frac{(n-1)(2n-1)}{3}$ ”. url: math.stackexchange.com/q/2343. (accessed: 2022-01-23).

A Technical derivations

A.1 Derivation of explicit parameter-shift rules

Here we derive the trigonometric interpolation via Dirichlet kernels.

A.1.1 Full reconstruction

We start out by exactly determining $E(x)$ given its value at points $\{x_\mu = \frac{2\mu}{2R+1}\pi, \mu \in \{-R, \dots, R\}$. This is a well-known problem [55, Chapter X]; we reproduce the result below for completeness.

Consider the *Dirichlet kernel*

$$D(x) = \frac{1}{2R+1} + \frac{2}{2R+1} \sum_{\ell=1}^R \cos(\ell x) \quad (54)$$

$$= \frac{\sin\left(\frac{2R+1}{2}x\right)}{(2R+1)\sin\left(\frac{1}{2}x\right)} \quad (55)$$

where the limit $x \rightarrow 0$ is taken when evaluating $D(0)$. The functions $D(x-x_\mu)$ are linear combinations of the basis functions $\{\sin(\ell x)\}_{\ell \in [R]}$, $\{\cos(\ell x)\}_{\ell \in [R]_0}$, and they satisfy $D(x_{\mu'} - x_\mu) = \delta_{\mu\mu'}$. Therefore it is evident that

$$E(x) = \sum_{\mu=-R}^R E(x_\mu) D(x-x_\mu) \quad (56)$$

$$= \frac{\sin\left(\frac{2R+1}{2}x\right)}{2R+1} \sum_{\mu=-R}^R E(x_\mu) \frac{(-1)^\mu}{\sin\left(\frac{x-x_\mu}{2}\right)}. \quad (57)$$

As an example, for $R=1$ (e.g., when the generator G satisfies $G^2 = \mathbb{1}$) we have the formula

$$E(x) = \frac{\sin\left(\frac{3}{2}x\right)}{3} \left[-\frac{E\left(-\frac{2}{3}\pi\right)}{\sin\left(\frac{x}{2} + \frac{\pi}{3}\right)} + \frac{E(0)}{\sin\left(\frac{x}{2}\right)} - \frac{E\left(\frac{2}{3}\pi\right)}{\sin\left(\frac{x}{2} - \frac{\pi}{3}\right)} \right]. \quad (58)$$

Derivatives of $E(x)$ can be straightforwardly extracted from this full reconstruction.

A.1.2 Odd kernels

We now consider the case of determining E_{odd} given its value at evenly spaced points $\{x_\mu = \frac{2\mu-1}{2R}\pi\}_{\mu \in [R]}$ ¹⁵. Consider the *modified Dirichlet kernel*:

$$D^*(x) = \frac{1}{2R} + \frac{1}{2R} \cos(Rx) + \frac{1}{R} \sum_{\ell=1}^{R-1} \cos(\ell x) \quad (59)$$

$$= \frac{\sin(Rx)}{2R \tan\left(\frac{1}{2}x\right)} \quad (60)$$

where we again assume the limit $x \rightarrow 0$ is taken when evaluating $D^*(0)$. This kernel satisfies the relations

$$D^*(x_{\mu'} - x_\mu) = \delta_{\mu\mu'}, \quad D^*(x_{\mu'} + x_\mu) = 0, \quad (61)$$

but unfortunately, $D^*(x)$ is a linear combination of cosines, not sines; it’s an even function, not an odd function. We therefore instead consider the linear combinations

$$\begin{aligned} \tilde{D}_\mu(x) &:= D^*(x-x_\mu) - D^*(x+x_\mu) \quad (62) \\ &= \frac{\sin(R(x-x_\mu))}{2R \tan\left(\frac{1}{2}(x-x_\mu)\right)} - \frac{\sin(R(x+x_\mu))}{2R \tan\left(\frac{1}{2}(x+x_\mu)\right)} \\ &= \frac{1}{R} \cos(x_\mu) \left[\frac{1}{2} \sin(Rx) + \sum_{\ell=1}^{R-1} \sin(\ell x) \right]. \end{aligned}$$

¹⁵Unlike Sec. A.1.1, we are not aware of a prior reference for the derivations for this subsection (reconstructing the odd part) and the next (reconstructing the even part).

Similarly to D^* , this kernel satisfies $\tilde{D}_\mu(x_{\mu'}) = \delta_{\mu\mu'}$ but it's a linear combination of the odd basis functions $\sin(\ell x)$, $\ell \in [R]$. Following from these two properties, we know that

$$\begin{aligned} E_{\text{odd}}(x) &= \sum_{\mu=1}^R E_{\text{odd}}(x_\mu) \tilde{D}_\mu(x) \\ &= \sum_{\mu=1}^R \frac{E_{\text{odd}}(x_\mu)}{2R} \\ &\quad \times \left[\frac{\sin(R(x-x_\mu))}{\tan(\frac{1}{2}(x-x_\mu))} - \frac{\sin(R(x+x_\mu))}{\tan(\frac{1}{2}(x+x_\mu))} \right] \end{aligned} \quad (63)$$

and we thus can reconstruct E_{odd} with the R evaluations $E_{\text{odd}}(x_\mu)$.

We also can extract from here a closed-form formula for the derivative at $x=0$, as it only depends on the odd part of E . We arrive at the *general parameter-shift rule*:

$$E'(0) = \sum_{\mu=1}^R E_{\text{odd}}(x_\mu) \tilde{D}'_\mu(0) \quad (64)$$

$$\begin{aligned} &= \sum_{\mu=1}^R E_{\text{odd}}(x_\mu) \frac{\sin(Rx_\mu)}{2R \sin^2(\frac{1}{2}x_\mu)} \\ &= \sum_{\mu=1}^R E_{\text{odd}} \left(\frac{2\mu-1}{2R} \pi \right) \frac{(-1)^{\mu-1}}{2R \sin^2(\frac{2\mu-1}{4R} \pi)}. \end{aligned} \quad (65)$$

Similarly, as the higher-order derivatives of \tilde{D}_μ can be computed analytically, we may obtain derivatives of E of higher odd orders.

A.1.3 Even kernels

Next we reconstruct the even part E_{even} again using the kernel $D^*(x)$ from above but choosing the $R+1$ points $x_\mu = \mu\pi/R$ for $\mu \in [R]_0$. As the spacing between these points is the same as between the previous $\{x_\mu\}$, we again have $D^*(x_{\mu'} - x_\mu) = \delta_{\mu\mu'}$; but note we cannot directly use $D^*(x - x_\mu)$ as our kernel because $D^*(x - x_\mu)$ is an even function in $x - x_\mu$ but not in x . Instead we take the even linear combination

$$\hat{D}_\mu(x) := \begin{cases} D^*(x) & \text{if } \mu = 0 \\ D^*(x - x_\mu) + D^*(x + x_\mu) & \text{if } 0 < \mu < R \\ D^*(x - \pi) & \text{if } \mu = R. \end{cases}$$

Then the \hat{D}_μ are even functions and satisfy $\hat{D}_\mu(x_{\mu'}) = \delta_{\mu\mu'}$, leading to

$$E_{\text{even}}(x) = \sum_{\mu=0}^R E_{\text{even}}(x_\mu) \hat{D}_\mu(x). \quad (66)$$

The second derivative of D^* is

$$D^{*''}(x) = \frac{\sin(Rx) \left[1 - 2R^2 \sin^2(\frac{1}{2}x) \right]}{4R \tan(\frac{1}{2}x) \sin^2(\frac{1}{2}x)} - \frac{\cos(Rx)}{2 \sin^2(\frac{1}{2}x)}$$

and if we take the limit $x \rightarrow 0$:

$$D^{*''}(0) = -\frac{2R^2 + 1}{6}. \quad (67)$$

This yields the explicit parameter-shift rule for the second derivative:

$$\begin{aligned} E''(0) &= -E_{\text{even}}(0) \frac{2R^2 + 1}{6} + E_{\text{even}}(\pi) \frac{(-1)^{R-1}}{2} \\ &\quad + \sum_{\mu=1}^{R-1} E_{\text{even}} \left(\frac{\mu\pi}{R} \right) \frac{(-1)^{\mu-1}}{\sin^2(\frac{\mu\pi}{2R})}. \end{aligned} \quad (68)$$

Again, derivatives of E of higher even order can be computed in a similar manner, using the same evaluations $E_{\text{even}}(\frac{\mu\pi}{R})$.

A.2 Hessian parameter-shift rule

Here we consider the spectrum of the function

$$E^{(km)}(x) := E(\mathbf{x}_0 + x\mathbf{v}_{k,m}), \quad (69)$$

with $\mathbf{v}_{k,m} = \mathbf{v}_k + \mathbf{v}_m$. Without loss of generality, we assume U_k to act first within the circuit and set $\mathbf{x}_0 = \mathbf{0}$. As for the univariate case in Sec. 2.1, we may explicitly write the cost function as

$$\begin{aligned} E^{(km)}(x) &= \langle \psi | U_k^\dagger(x) V^\dagger U_m^\dagger(x) B U_m(x) V U_k(x) | \psi \rangle \\ &= \sum_{j_1, \dots, j_4=1}^d \overline{\psi_{j_1} v_{j_2 j_1} b_{j_2 j_3} v_{j_3 j_4} \psi_{j_4}} \\ &\quad \times \exp \left(i \left(\omega_{j_4}^{(k)} - \omega_{j_1}^{(k)} + \omega_{j_3}^{(m)} - \omega_{j_2}^{(m)} \right) x \right), \end{aligned} \quad (70)$$

where $\omega^{(k,m)}$ are the eigenvalues of the generators of U_k and U_m , respectively, and we denoted the entries of matrices by lowercase letters as before. We may read off the occurring frequencies in this Fourier series in terms of the unique positive differences $\Omega^{(k,m)}$, leading to $\delta\Omega_{l_1 l_2} = \pm\Omega_{l_1}^{(k)} \pm \Omega_{l_2}^{(m)}$. We again only collect the positive values as they come in pairs¹⁶.

In case of integer-valued frequencies, there are $R_{km} = R_k + R_m$ such positive frequencies, namely all integers in $[R_k + R_m]$. For arbitrary frequencies, all $\{\delta\Omega\}$ might be unique and we obtain up to $R_{km} = 2R_k R_m + R_k + R_m$ frequencies. Rescaling the smallest frequency enforces a small degree of redundancy so that $R_{km} = 2R_k R_m + R_k + R_m - 2$ is always achievable; for some scenarios specific rescaling factors might drastically reduce R_{km} ¹⁷.

¹⁶That is, for any $\delta\Omega$, we also have $-\delta\Omega$ in the Fourier series, and the representation as real-valued function subsums the two frequencies.

¹⁷Recall that we used rescaling for the equidistant frequency case to arrive at integer-valued $\{\Omega\}$, which in turn made the significant reduction above possible.

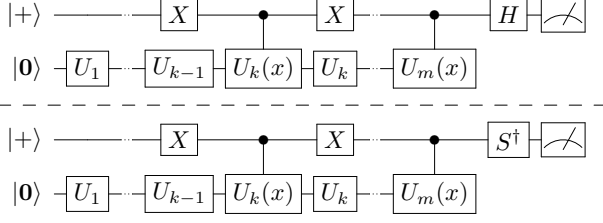


Figure 5: Circuits for the Hadamard tests to measure the overlap in Eq. (71), adapted from [57, Fig. 5]. The basis rotation in the last operation on the auxiliary qubit determines whether the real (*top*) or the imaginary (*bottom*) part of $\langle \psi(\mathbf{x}_0 + x\mathbf{v}_{k,m}) | \psi(\mathbf{x}_0) \rangle$ is calculated. All unitaries without argument are understood as $U_j = U_j((\mathbf{x}_0)_j)$.

A.3 Hadamard tests for the metric tensor

In order to compute the metric tensor as the Hessian of the overlap $f(\mathbf{x}) = -\frac{1}{2}|\langle \psi(\mathbf{x}) | \psi(\mathbf{x}_0) \rangle|^2$, we need to evaluate it at shifted positions $\mathbf{x} = \mathbf{x}_0 + x\mathbf{v}_{k,m}$. This can be done by executing the circuit $V(\mathbf{x}_0)$ and the adjoint circuit $V^\dagger(\mathbf{x})$ at the shifted position, and returning the probability to measure the $\mathbf{0}$ bitstring in the computational basis. As all operations after the latter of the two parametrized gates of interest cancel between the two circuits, those operations can be spared, but the maximal depth is (almost) the doubled depth of V .

Alternatively, we may use a Hadamard test as derived in the appendix of Ref. [57]. There, it was designed to realize the derivative overlaps $\Re\{\langle \partial_k \psi(\mathbf{x}) | \partial_m \psi(\mathbf{x}) \rangle\}$ for the metric tensor directly, assuming the generator to be a Pauli word and therefore unitary. However, it can also be used to calculate the real or imaginary part of

$$\begin{aligned} \langle \psi(\mathbf{x}) | \psi(\mathbf{x}_0) \rangle &= \langle \mathbf{0} | U_1^\dagger((\mathbf{x}_0)_1) \cdots U_k^\dagger((\mathbf{x}_0)_k + x) \\ &\quad \cdots U_{m-1}^\dagger((\mathbf{x}_0)_{m-1}) U_m^\dagger(x) U_{m-1}((\mathbf{x}_0)_{m-1}) \\ &\quad \cdots U_1((\mathbf{x}_0)_1) | \mathbf{0} \rangle. \end{aligned} \quad (71)$$

by measuring the auxiliary qubit in the Z or Y basis. The corresponding circuit is shown in Fig. 5.

While the original proposal has to split up the generators into Pauli words and implement one circuit per combination of Pauli words from x_k and x_m , the number of circuits here is dictated by the number of evaluations in the parameter-shift rule. In order to measure $f(\mathbf{x})$, the real and the imaginary part both have to be measured, doubling the number of circuits.

A.4 Coefficient norms for univariate derivatives via equidistant shifts

The ℓ_1 -norm of the coefficients in parameter-shift rules dictates the number of shots required to reach certain precision (see Sec. 2.3). Here, we explicitly compute this norm for both the general and decomposition-based parameter-shift rule for the first- and second-order univariate derivative. For the entire

analysis, we approximate the single-shot variance σ^2 to be constant as detailed in the main text.

A.4.1 Norm for general parameter-shift rule

For the case of equidistant shift angles, we can compute the norm of the coefficient vector $\mathbf{y}^{(1,2)}$ in the parameter-shift rules in Eqs. (24,25) explicitly, in order to estimate the required shot budget for the obtained derivative. For the first order, we note that the evaluations of E come in pairs, with the same coefficient up to a relative sign. This yields (recalling that $x_\mu = \frac{2\mu-1}{4R}\pi$):

$$\|\mathbf{y}^{(1)}\|_1 = \frac{1}{2R} \sum_{\mu=1}^R \frac{1}{\sin^2(x_\mu)} = R, \quad (72)$$

which follows from $\sin^{-2}(x_\mu) = \cot^2(x_\mu) + 1$ and [80, Formula (445)]:

$$\sum_{\mu=1}^R \cot^2(x_\mu) = 2R^2 - R. \quad (73)$$

A derivation for Eq. (73) can be adapted from Ref. [81], which we present below for completeness:

$$\begin{aligned} -i(-1)^\mu &= \exp(i2Rx_\mu) \\ &= \left(\cos(x_\mu) + i \sin(x_\mu) \right)^{2R} \\ &= \sum_{r=0}^{2R} \binom{2R}{r} (\cos(x_\mu))^{2R-r} (i \sin(x_\mu))^r \\ \Rightarrow 0 &= \sum_{r=0}^R \binom{2R}{2r} (\cos(x_\mu))^{2R-2r} (i \sin(x_\mu))^{2r} \\ &= \sum_{r=0}^R \binom{2R}{2r} \left(-\cot^2(x_\mu) \right)^{R-r} \end{aligned}$$

Here we have applied the binomial theorem, extracted the real part, and divided by $(i \sin(x_\mu))^{2R}$ (note that $0 < x_\mu < \pi/2$). From the last equation above, we see that $\cot^2(x_\mu)$ is a root of the function $g(\chi) = \sum_{r=0}^R \binom{2R}{2r} (-\chi)^{R-r}$ for all $\mu \in [R]$. As g is a polynomial of degree R , we thus know *all* its roots and may use the simplest of Vieta's formulas:

$$\sum_{\mu=1}^R \tau_\mu = -\frac{g_{R-1}}{g_R} \quad (74)$$

with roots $\{\tau_\mu\}_\mu$ of g , and g_j the j th order Taylor coefficient of g . Plugging in the known roots and coefficients we get

$$\sum_{\mu=1}^R \cot^2(x_\mu) = -\frac{(-1)^{R-1} \binom{2R}{2}}{(-1)^R \binom{2R}{0}} \quad (75)$$

$$= 2R^2 - R. \quad (76)$$

For the second order we may repeat the above computation with small modifications¹⁸, arriving at $g(\chi) = \sum_{r=0}^{R-1} \binom{2R}{2r+1} (-\chi)^{R-r}$ and therefore at

$$\begin{aligned} \|\mathbf{y}_1^{(2)}\| &= \frac{2R^2 + 1}{6} + \frac{1}{2} + (R-1) - \frac{(-1)^{R-1} \binom{2R}{3}}{(-1)^R \binom{2R}{1}} \\ &= R^2. \end{aligned} \quad (77)$$

A.4.2 Norm for decomposition

If we compute the first- and second-order derivatives via a decomposition that contains \mathcal{P} parametrized elementary gates, we need to apply the original two-term parameter-shift rule to each of these gates separately. For the first-order derivative, we simply sum all elementary derivatives. For integer-valued frequencies, x typically feeds without prefactor into the gates in the decomposition, so that the decomposition-based shift rule reads

$$E'(0) = \frac{1}{2 \sin(x_1)} \sum_{k=1}^{\mathcal{P}} [E^{(k)}(x_1) - E^{(k)}(-x_1)], \quad (78)$$

where $E^{(k)}$ denotes the cost function based on the decomposition, in which only the parameter of the k th elementary gate is set to the shifted angle x_1 and to 0 in all other gates. To maximize $\sin(x_1)$, we choose $x_1 = \pi/2$, and as a result all $2\mathcal{P}$ coefficients have magnitude $1/2$, and therefore

$$\|\mathbf{y}_{\text{decomp}}^{(1)}\|_1 = \mathcal{P}. \quad (79)$$

Due to all coefficients being equal, the optimal shot allocation is $N/(2\mathcal{P})$ for all terms.

For the second-order derivative, the full Hessian has to be computed from the decomposition as described in Ref. [46] and all elements have to be summed¹⁹:

$$\begin{aligned} E''(0) &= \frac{1}{2 \sin^2(x_1)} \sum_{\substack{k,m=1 \\ k < m}}^{\mathcal{P}} \quad (80) \\ &\left[E^{(km)}(x_1, x_1) - E^{(km)}(-x_1, x_1) \right. \\ &\quad \left. - E^{(km)}(x_1, -x_1) + E^{(km)}(-x_1, -x_1) \right] \\ &+ \frac{1}{2} \sum_{k=1}^{\mathcal{P}} [E^{(k)}(\pi) - E(0)] \end{aligned}$$

where $E^{(km)}(x_1, x_2)$ is defined analogously to $E^{(k)}$ but the shift angles put into the k th and m th elementary gate may differ. Fixing the shift angle to $\pi/2$ again, we have $2\mathcal{P}(\mathcal{P}-1)$ coefficients of magnitude

¹⁸Recall that the angles differ between the two derivatives.

¹⁹Here we do not anticipate the cheaper Hessian evaluation from Sec. 4.1.

$1/2$ for the off-diagonal terms, \mathcal{P} coefficients of magnitude $1/2$ for the $E^{(k)}(\pi)$ and one coefficient with magnitude $\mathcal{P}/2$ for $E(0)$, summing to

$$\|\mathbf{y}_{\text{decomp}}^{(1)}\|_1 = 2\mathcal{P}(\mathcal{P}-1)\frac{1}{2} + \mathcal{P}\frac{1}{2} + \frac{\mathcal{P}}{2} = \mathcal{P}^2. \quad (81)$$

Here the optimal shot allocation is to measure all shifted terms with $N/(2\mathcal{P}^2)$ shots, and $E(0)$ with $N/(2\mathcal{P})$ shots.

A.5 Coefficient norms for the Hessian

Similar to the previous section, we compute the coefficient norms for three methods to compute the Hessian for equidistant frequencies and shifts: We may use the diagonal shift rule in Eq. (36), repeat the general parameter-shift rule, or decompose the circuit and repeat the original parameter-shift rule. For the first approach, the diagonal entries of the Hessian—and thus the shifted evaluations for those entries—are reused to compute the off-diagonal ones, whereas the shifted evaluations for the repeated shift rule are distinct for all Hessian entries. This difference makes the cost comparison for a single Hessian entry difficult. We therefore consider the root mean square of the Frobenius norm of the difference between the true and the estimated Hessian as quality measure. The matrix of expected deviations is given by the standard deviations σ_{km} so that we need to compute

$$\varepsilon = \sqrt{\sum_{k,m=1}^n \sigma_{km}^2} = \sqrt{\sum_{k=1}^n \sigma_k^2 + \sum_{k < m} 2\sigma_{km}^2}. \quad (82)$$

A.5.1 Hessian shift rule

The variance for a Hessian diagonal entry H_{kk} is $\sigma^2 R_k^4 / N_{kk}$ if we use N_{kk} shots to estimate it (see Eq. (29))²⁰. For an off-diagonal element H_{km} computed via the diagonal shift rule in Eq. (36), the variance is

$$\sigma_{km}^2 = \frac{1}{4} \left(\frac{\sigma^2 (R_k + R_m)^4}{N_{km}} + \frac{\sigma^2 R_k^4}{N_{kk}} + \frac{\sigma^2 R_m^4}{N_{mm}} \right), \quad (83)$$

where we used that $R_{km} = R_k + R_m$ for equidistant frequencies. Overall, this yields

$$\varepsilon^2 = \sum_{k=1}^n \frac{\sigma^2 R_k^4}{N_{kk}} \frac{n+1}{2} + \sum_{k < m} \frac{\sigma^2 (R_k + R_m)^4}{2N_{km}} \quad (84)$$

If we allocate N_{diag} shots optimally, that is N_{km} is proportional to the square root of the coefficient of N_{km}^{-1} , we require

$$\begin{aligned} N_{\text{diag}} &= \frac{\sigma^2}{\varepsilon^2} \left[\sum_{k=1}^n R_k^2 \sqrt{\frac{n+1}{2}} + \sum_{k < m} \frac{1}{\sqrt{2}} (R_k + R_m)^2 \right]^2 \\ &= \frac{\sigma^2}{2\varepsilon^2} \left[(\sqrt{n+1} + n - 2) \|\mathbf{R}\|_2^2 + \|\mathbf{R}\|_1^2 \right]^2 \end{aligned} \quad (85)$$

shots to estimate H to a precision of ε .

²⁰Recall that σ^2 is the single-shot variance.

A.5.2 Repeated general parameter-shift rule

Without the diagonal shift rule, we compute H_{km} by executing the univariate general parameter-shift rule in Eq. (24) for x_k and x_m successively, i.e., we apply the rule for x_m to all terms from the rule for x_k . This leads to $4R_k R_m$ terms with their coefficients arising from the first-order shift rule coefficients by multiplying them together:

$$\begin{aligned} \|\mathbf{y}^{(km)}\|_1 &= \frac{1}{4R_k R_m} \sum_{\mu=1}^{R_k} \frac{1}{\sin^2(x_\mu)} \sum_{\mu'=1}^{R_m} \frac{1}{\sin^2(x_{\mu'})} \\ &= R_k R_m, \end{aligned} \quad (86)$$

where we used Eq. (72). Correspondingly, the variance for H_{km} computed by this methods with an optimal shot allocation of N_{km} shots is $\sigma_{km}^2 = \sigma^2 R_k^2 R_m^2 / N_{km}$. The mean square of the Frobenius norm then is

$$\varepsilon^2 = \sum_{k=1}^n \frac{\sigma^2 R_k^4}{N_{kk}} + \sum_{k < m} \frac{2\sigma^2 R_k^2 R_m^2}{N_{km}} \quad (87)$$

and an optimal shot allocation across the entries of the Hessian to achieve a precision of ε will require

$$\begin{aligned} N_{\text{genPS}} &= \frac{\sigma^2}{\varepsilon^2} \left[\sum_{k=1}^n R_k^2 + \sum_{k < m} \sqrt{2} R_k R_m \right]^2 \\ &= \frac{\sigma^2}{2\varepsilon^2} \left[(\sqrt{2} - 1) \|\mathbf{R}\|_2^2 + \|\mathbf{R}\|_1^2 \right]^2 \end{aligned} \quad (88)$$

shots in total.

A.5.3 Decomposition and repeated original shift rule

For the third approach, we only require the observation that again all (unique) Hessian entries are estimated independently and that the coefficients arise from all products of two coefficients from the separate shift rules for x_k and x_m . This yields $4\mathcal{P}_k \mathcal{P}_m$ coefficients with magnitude $1/4$, so that the calculation of ε is the same as for the previous approach, replacing \mathbf{R} by \mathcal{P} . The required shot budget for a precision of ε is thus

$$N_{\text{decomp}} = \frac{\sigma^2}{2\varepsilon^2} \left[(\sqrt{2} - 1) \|\mathcal{P}\|_2^2 + \|\mathcal{P}\|_1^2 \right]^2 \quad (89)$$

B Generalization to arbitrary spectra

Throughout this work, we mostly focused on cost functions E with equidistant — and thus, by rescaling, integer-valued — frequencies $\{\Omega_\ell\}$. Here we will discuss the generalization to arbitrary frequencies, mostly considering the changed cost.

B.1 Univariate functions

The nonuniform DFT used to reconstruct the full function E in Sec. 3.1, and its modifications for the

odd and even part in Secs. 3.2 and 3.3, can be used straightforwardly for arbitrary frequencies. However, choosing equidistant shift angles $\{x_\mu\}$ will no longer make the DFT uniform, as was the case for equidistant frequencies. Correspondingly, the explicit parameter-shift rules for $E'(0)$ and $E''(0)$ in Eqs. (24, 25) do not apply and in general we do not know a closed-form expression for the DFT or the parameter-shift rules. Symbolically, the parameter-shift rule takes the form

$$E'(0) = \sum_{\mu=1}^R y_\mu^{(1)} [E(x_\mu) - E(-x_\mu)] \quad (90)$$

$$E''(0) = y_0^{(2)} E(0) + \sum_{\mu=1}^R y_\mu^{(2)} [E(x_\mu) + E(-x_\mu)]. \quad (91)$$

Regarding the evaluation cost, the odd part and thus odd-order derivatives can be obtained at the same price of $2R$ evaluations of E as before, but the even part might no longer be periodic in general; as a consequence,

$$E_{\text{even}}(\pi) = \frac{1}{2}(E(\pi) + E(-\pi)) \neq E(\pi) \quad (92)$$

actually may require two evaluations of E , leading to $2R + 1$ evaluations overall. If the even part is periodic, which is equivalent to all involved frequencies being commensurable, with some period T , evaluating $E_{\text{even}}(T/2)$ allows to skip the additional evaluation.

When comparing to the first derivative based on a decomposition into \mathcal{P} parametrized elementary gates, the break-even point for the number of unique circuits remains at $R = \mathcal{P}$ as for equidistant frequencies, but we note that e.g., a decomposition of the form

$$U(x) = \prod_{k=1}^{\mathcal{P}} U_k(\beta_k x), \quad (93)$$

namely where x is rescaled individually in each elementary gate by some $\beta_k \in \mathbb{R}$, in general will result in $R = \mathcal{P}^2$ frequencies of E , making the decomposition-based parameter-shift rule beneficial. For the second-order derivative, the number of evaluations $2R + 1$ might be quadratic in \mathcal{P} in the same way, but the decomposition requires $2\mathcal{P}^2 - \mathcal{P} + 1$ as well, so that the requirements are similar if $R = \mathcal{P}$.

Regarding the required number of shots, we cannot make concrete statements for the general case as we don't have a closed-form expression for the coefficients \mathbf{y} , but note that for the decomposition approach, rescaling factors like the $\{\beta_k\}$ in Eq. (93) above have to be factored in via the chain rule, leading to a modified shot requirement.

An example for unitaries with non-equidistant frequencies would be the QAOA layer that implements the time evolution under the problem Hamiltonian (see Eq. (26)) for MAXCUT on *weighted* graphs with non-integer weights.

For the stochastic parameter-shift rule in Sec. 3.6 we did not restrict ourselves to equidistant frequencies and derive it in App. C for general unitaries of the form $U_F = \exp(i(xG + F))$ directly.

B.2 Multivariate functions

While the univariate functions do not differ strongly for equidistant and arbitrary frequencies in E and mostly the expected relation between R and \mathcal{P} changes, the shift rule for the Hessian and the metric tensor are affected heavily by generalizing the spectrum. First, the univariate restriction $E^{(km)}(x)$ in Eq. (34) still can be used to compute the off-diagonal entry H_{km} of the Hessian but this may require up to $2R_{km} + 1 = 4R_k R_m + 2R_k + 2R_m - 3$ evaluations (see App. A.2), in contrast to $2R_{km} = 2(R_k + R_m)$ in the equidistant case. Compared to the resource requirements of the decomposition-based approach, $4\mathcal{P}_k \mathcal{P}_m$, this makes our general parameter-shift rule more expensive if $R_k \gtrsim \mathcal{P}_k$.

As we use the same method to obtain the metric tensor \mathcal{F} , the number of evaluations grows in the same manner, making the decomposition-based shift rule more feasible for unitaries with non-equidistant frequencies. As $f(\mathbf{x}_0)$ does not have to be evaluated, an off-diagonal element \mathcal{F}_{km} requires one evaluation fewer than H_{km} , namely $4R_k R_m + 2R_k + 2R_m - 4$.

C General stochastic shift rule

In this section we describe a stochastic variant of the general parameter-shift rule which follows immediately from combining the rule for single-parameter gates in Eq. (90) with the result from Ref. [39].

First, note that any shift rule

$$E'(x_0) = \sum_{\mu} y_{\mu} E(x_0 + x_{\mu}), \quad (94)$$

with coefficients $\{y_{\mu}\}$ and shift angles $\{x_{\mu}\}$ for a unitary $U(x) = \exp(ixG)$, implies that we can implement the commutator with G :

$$i[G, \rho] = \sum_{\mu} y_{\mu} U(x_{\mu}) \rho U^{\dagger}(x_{\mu}), \quad (95)$$

since the commutator between G and the Hamiltonian directly expresses the derivative of the expectation value $E'(0)$ on the operator level, and shift rules hold for arbitrary states.

Now consider the extension $U_F(x) = \exp(i(xG + F))$ of the above unitary. In the original stochastic

parameter-shift rule, the authors show²¹

$$E'(x_0) = \int_0^1 dt \operatorname{tr} \left\{ U_F^{\dagger}(tx_0) B U_F(tx_0) \right. \\ \left. \times i \left[G, U_F((1-t)x_0) |\psi\rangle\langle\psi| U_F^{\dagger}((1-t)x_0) \right] \right\} \quad (96)$$

where we again denoted the state prepared by the circuit before U_F by $|\psi\rangle$ and the observable transformed by the circuit following U_F by B . By using Eq. (95) to express the commutator, we obtain

$$E'(x_0) = \int_0^1 dt \sum_{\mu} y_{\mu} \operatorname{tr} \left\{ U_F^{\dagger}(tx_0) B U_F(tx_0) \right. \\ \left. \times U(x_{\mu}) U_F((1-t)x_0) |\psi\rangle\langle\psi| U_F^{\dagger}((1-t)x_0) U^{\dagger}(x_{\mu}) \right\}. \quad (97)$$

We abbreviate the interleaved unitaries

$$U_{F,\mu}(x_0, t) := U_F(tx_0) U(x_{\mu}) U_F((1-t)x_0) \quad (98)$$

and denote the cost function that uses $U_{F,\mu}(x_0, t)$ instead of $U_F(x_0)$ as

$$E_{\mu}(x_0, t) := \operatorname{tr} \left\{ B U_{F,\mu}^{\dagger}(x_0, t) |\psi\rangle\langle\psi| U_{F,\mu}(x_0, t) \right\}.$$

Rewriting Eq. (97) then yields the *generalized stochastic parameter-shift rule*

$$E'(x_0) = \int_0^1 dt \sum_{\mu} y_{\mu} E_{\mu}(x_0, t). \quad (99)$$

It can be implemented by sampling values for the splitting time t , combining the shifted energies $E_{\mu}(x_0, t)$ for each sampled t with the coefficients y_{μ} , and averaging over the results.

D Details on QAD

In this section we provide details on the latter two of the three modifications of the QAD algorithm discussed in Sec. 5.3.

D.1 Extended QAD model for Pauli rotations

The QAD model introduced in Ref. [49] contains trigonometric functions up to second (leading) order. The free parameters of the model cannot be extracted with one function evaluation per degree of freedom, because unlike standard monomials in a Taylor expansion, the trigonometric basis functions mix the orders in the input parameters. This leads to the mismatch of $2n^2 + n + 1$ (original QAD) or $3n^2/2 + n/2 + 1$ (see above) evaluations to obtain $n^2/2 + 3n/2 + 1$ model parameters. We note that the QAD model contains full univariate reconstructions at optimal cost, extracting

²¹To be precise, we here combine Eqs. (11-13) in Ref. [39] into a general expression for E' .

the $2n + 1$ model parameters $E^{(A)}$, $E^{(B)}$ and $E^{(C)}$ from $2n + 1$ function evaluations. The doubly shifted evaluations, however, are used for the Hessian entry only:

$$E_{km}^{(D)} = \frac{1}{4} [E_{km}^{++} - E_{km}^{+-} - E_{km}^{-+} + E_{km}^{--}], \quad (100)$$

where $E_{km}^{\pm\pm} = E(\mathbf{x}_0 \pm \frac{\pi}{2}\mathbf{v}_k \pm \frac{\pi}{2}\mathbf{v}_m)$ and we recall that this QAD model is restricted to Pauli rotations only.

Let us now consider a slightly larger truncation of the cost function than the one presented in App. A 2 in [49]:

$$\begin{aligned} \hat{E}(\mathbf{x}_0 + \mathbf{x}) = & A(\mathbf{x}) \left[E^{(A)} \right. \\ & + 2\mathbf{E}^{(B)} \cdot \tan\left(\frac{\mathbf{x}}{2}\right) + 2\mathbf{E}^{(C)} \cdot \tan\left(\frac{\mathbf{x}}{2}\right)^{\odot 2} \\ & + 4 \tan\left(\frac{\mathbf{x}}{2}\right) E^{(D)} \tan\left(\frac{\mathbf{x}}{2}\right) \\ & + 4 \tan\left(\frac{\mathbf{x}}{2}\right) E^{(F)} \tan^2\left(\frac{\mathbf{x}}{2}\right) \\ & \left. + 4 \tan^2\left(\frac{\mathbf{x}}{2}\right) E^{(G)} \tan^2\left(\frac{\mathbf{x}}{2}\right) \right] \end{aligned} \quad (101)$$

with $A(\mathbf{x}) = \prod_k \cos^2(x_k/2)$. $E^{(F)}$ and $E^{(G)}$ have zeros on their diagonals because there are no terms of the form $\sin^3(x_k/2)$ or $\sin^4(x_k/2)$ in the cost function, and for $E^{(G)}$ we only require the strictly upper triangular entries due to symmetry. The higher-order terms contain at least three distinct variables x_k , x_l and x_m because all bivariate terms are captured in the above truncation. Using

$$A\left(\pm\frac{\pi}{4}\mathbf{v}_k \pm \frac{\pi}{4}\mathbf{v}_m\right) = \frac{1}{4} \quad \text{and} \quad \tan\left(\pm\frac{\pi}{4}\right) = \pm 1,$$

we now can compute:

$$\begin{aligned} E_{km}^{++} - E_{km}^{-+} + E_{km}^{+-} - E_{km}^{--} &= E_k^{(B)} + E_{km}^{(F)} \\ E_{km}^{++} + E_{km}^{-+} + E_{km}^{+-} + E_{km}^{--} &= E^{(A)} + 2E_k^{(C)} \\ &\quad + 2E_m^{(C)} + 4E_{km}^{(G)}. \end{aligned}$$

This means that the 4 function evaluations $E_{km}^{\pm\pm}$ that are used for $E_{km}^{(D)}$ in the original QAD can be recycled to obtain the 3 parameters $E_k^{(B)}$, $E_m^{(C)}$ and $E_{km}^{(G)}$. The corresponding model is of the form Eq. (101) and therefore includes *all* terms that depend on two parameters only. Consequently, the constructed model exactly reproduces the cost function not only on the coordinate axes but also on all coordinate planes spanned by any two of the axes. The number of model parameters is $2n^2 + 1$, which matches the total number of function evaluations.

D.2 Trigonometric interpolation for QAD

Both the original QAD algorithm, and the extension introduced above, assume the parametrized quantum

circuit to consist of Pauli rotation gates exclusively. In the spirit of the generalized function reconstruction and parameter-shift rule, we would like to relax this assumption and generalize the QAD model. However, there is no obvious unique way to do this, because the correspondence between the gradient and $E^{(B)}$ and between the Hessian and $E^{(C,D)}$ is not preserved for multiple frequencies. Instead, the uni- and bivariate Fourier coefficients of E form the model parameters and the derivative quantities are contractions with the frequencies thereof. There are multiple ways in which we could generalize QAD to multiple frequencies.

The first way to generalize QAD is to compute the gradient and Hessian with the generalized parameter-shift rule Eq. (24) and the shift rule for Hessian entries Eq. (36) and to construct a single-frequency model as in original QAD. Even though we know the original energy function to contain multiple frequencies, this would yield a local model with the correct second-order expansion at \mathbf{x}_0 that exploits the evaluations savings shown in this work. As QAD is supposed to use the model only in the neighbourhood of \mathbf{x}_0 , this might be sufficient for the optimization.

As a second generalization we propose a full trigonometric interpolation of E up to second order, similar to the univariate reconstruction in Sec. 3.1. First we consider the univariate part of the model: Start by evaluating E at positions shifted in the k th coordinate by equidistant points and subtract $E(\mathbf{x}_0)$,

$$E_\mu^{(k)} := E(\mathbf{x}_0 + x_\mu \mathbf{v}_k) - E(\mathbf{x}_0) \quad (102)$$

$$x_\mu := \frac{2\mu\pi}{2R_k + 1}, \quad \mu \in [2R_k]. \quad (103)$$

Then consider the (shifted) Dirichlet kernels

$$D_\mu^{(k)}(x) = \frac{1}{2R_k + 1} \left(1 + 2 \sum_{\ell=1}^{R_k} \cos(\ell(x - x_\mu)) \right) \quad (104)$$

$$= \frac{\sin\left(\frac{1}{2}(2R_k + 1)(x - x_\mu)\right)}{(2R_k + 1) \sin\left(\frac{1}{2}(x - x_\mu)\right)} \quad (105)$$

which satisfy $D_\mu^{(k)}(x_{\mu'}) = \delta_{\mu\mu'}$ and are Fourier series with integer frequencies up to R_k . Therefore, the function²²

$$\hat{E}^{(k)}(x) = \sum_{\mu=1}^{2R_k} E_\mu^{(k)} D_\mu^{(k)}(x) \quad (106)$$

coincides with $E(\mathbf{x}_0 + x\mathbf{v}_k) - E(\mathbf{x}_0)$ at $2R_k + 1$ points and is a trigonometric polynomial with the same R_k frequencies.

²²One might be wondering why to subtract $E(\mathbf{x}_0)$ just to add it manually back into the reconstruction now. This is because we need to avoid duplicating this term when adding up the univariate and bivariate terms of all parameters later on.

Similarly, the product kernels $D_{\mu\mu'}^{(km)}(x_k, x_m) = D_{\mu}^{(k)}(x_k)D_{\mu'}^{(m)}(x_m)$ can be used to reconstruct the bivariate restriction of E to the $x_k - x_m$ plane. For this, evaluate the function at doubly shifted positions and subtract both, $E(\mathbf{x}_0)$ and the univariate parts:

$$E_{\mu\mu'}^{(km)} := E(\mathbf{x}_0 + x_{\mu}\mathbf{v}_k + x_{\mu'}\mathbf{v}_m) \quad (107)$$

$$- \hat{E}^{(k)}(x_{\mu}) - \hat{E}^{(m)}(x_{\mu'}) - E(\mathbf{x}_0) \quad (108)$$

Then, the bivariate Fourier series

$$\hat{E}^{(km)}(x_k, x_m) = \sum_{\mu, \mu'=1}^{2R_k, 2R_m} E_{\mu\mu'}^{(km)} D_{\mu\mu'}^{(km)}(x_k, x_m) \quad (109)$$

coincides with $E(\mathbf{x}_0 + x_k\mathbf{v}_k + x_m\mathbf{v}_m) - E(\mathbf{x}_0) - \hat{E}^{(k)}(x_k) - \hat{E}^{(m)}(x_m)$ on the entire coordinate plane spanned by \mathbf{v}_k and \mathbf{v}_m .

As we constructed the terms such that they do not contain the respective lower order terms, we finally can combine them to the full trigonometric interpolation:

$$\begin{aligned} \hat{E}_{\text{interp}}(\mathbf{x}) &= E(\mathbf{x}_0) + \sum_{k=1}^n \hat{E}^{(k)}(x_k) \quad (110) \\ &+ \sum_{k < m} \hat{E}^{(km)}(x_k, x_m). \end{aligned}$$

This model has as many parameters as function evaluations, namely $2(\|\mathbf{R}\|_1^2 - \|\mathbf{R}\|_2^2 + \|\mathbf{R}\|_1) + 1$, and therefore, the trigonometric interpolation is the generalization of the extended QAD model in App. D.1. Indeed, for $R_k = 1$ for all k we get back $2(n^2 - n + n) + 1 = 2n^2 + 1$ evaluations and model parameters.

We note that the trigonometric interpolation can be implemented for non-equidistant evaluation points in a similar manner and with the same number of evaluations, although the elementary functions are no longer Dirichlet kernels but take the form

$$\hat{D}_{\mu}^{(k)}(x) = \frac{\sin(\frac{1}{2}x)}{\sin(\frac{1}{2}x_{\mu})} \prod_{\mu'=1}^{2R_k} \frac{\sin(\frac{1}{2}(x - x_{\mu'}))}{\sin(\frac{1}{2}(x_{\mu} - x_{\mu'}))}. \quad (111)$$

Chapter 3

Variational Quantum Algorithms

This chapter contains the second publication and a brief introduction to provide some context for it. I will not review variational quantum algorithms (VQAs) in depth but give additional references that provide detailed information and reviews. A review on VQAs and on the variational quantum eigensolver (VQE) for quantum chemistry can be found in [11] and [12], respectively. The problem Hamiltonians, optimization algorithms and circuit ansätze that are relevant for [15] in particular are introduced and discussed within the publication itself.

3.1 Ansätze, initializations and optimizers

Variational quantum algorithms aim at solving problems on NISQ computers that are hard to solve using classical computers. An archetypical structure for these algorithms is the following: starting from a problem to be solved, which is given in the form of a Hamiltonian H , choose a parametrized quantum circuit (PQC) C , a measurement scheme to evaluate the objective function $E(\boldsymbol{\theta}) = \langle 0 | C(\boldsymbol{\theta})^\dagger H C(\boldsymbol{\theta}) | 0 \rangle$, and initial input parameters $\boldsymbol{\theta}_0$ to the circuit. Afterwards, modify the parameters to find a minimum of E , by running an optimization strategy on a classical computer that is given access to evaluations of E and optionally other related quantities like the gradient ∇E , (co)variances e.g. between H and auxiliary observables, or geometric information about the prepared quantum state like the metric tensor F . Often all these quantities are supposed to be computed on the quantum processing unit (QPU), but there also are proposals to evaluate some auxiliary quantities approximately, e.g. to support the measurement strategy for the objective function [42].

Ansätze The choice of circuit, or *ansatz*, can but does not have to depend on the Hamiltonian. For example, in the quantum approximate optimization algorithm (QAOA) [16] and the related Hamiltonian variational ansatz [108], the Hamiltonian itself is used in the PQC, similar to adiabatic time evolution protocols. Other ansätze do not implement H itself but are heavily influenced by it, e.g. because they respect its symmetries [13, 15, 17, 109, 110].

Alternatively there are ansätze that are particularly convenient to implement on hardware [72], that show favourable training [89, 111] or generalization [24, 112] properties, or that are part of the problem formulation itself, like in variational compilation into a desired gate set or structure [113, 114].

Initialization The initialization of the variational parameters θ may seem like a rather insignificant detail in the larger scheme of VQAs, but turns out to influence the performance of the algorithms heavily [15, 89, 115]. For some applications the initial parameters can be chosen to resemble computational schemes from non-universal quantum processors like adiabatic protocols [116, 117] or we may inherit solutions of other problem instances of the same class, or of smaller size [118, 119, 120]. In other cases designated starting positions already correspond to approximate solutions to the problem in a bigger scheme, like in VQEs that start with the Hartree-Fock state, so that a generic initialization point is known. Initializing a sufficiently deep and unstructured PQC¹ uniformly at random across the parameter domain will lead to poor performance in the subsequent updating procedure due to vanishing gradients – or barren plateaus [89, 122] – and when in doubt, initializing a PQC close to the origin often is favourable over an initialization uniformly at random.

Optimizers The updating step(s) of the PQC parameters can be performed in many different ways and a lot of research has been put into coming up with new optimizers and evaluating them alongside with established algorithms e.g. from machine learning. This is also the subject of the publication attached in this chapter. I will not attempt summarize let alone review all relevant optimization strategies but give a few examples to reflect the spectrum of options. For more detailed reviews refer to e.g. [11, 73] and [12, Chap. 7].

While there are a lot of strategies to choose from, one class of commonly used optimizers are those that employ the gradient of the objective function and iteratively modify the parameters with an update rule of the form

$$\theta_{t+1} = \theta_t - f(\nabla E(\theta_t)), \quad (3.1)$$

where f is some function that processes the gradient before applying the update, which may or may not involve additional quantities computed on the quantum processor and also can depend on so-called hyperparameters, the parameter position θ itself or past update steps. The basic variant of this is *gradient descent*, for which $f_{\text{GD}}(\mathbf{x}, \eta) = \eta \mathbf{x}$ for some fixed *learning rate* $\eta > 0$. More complex variants include *Adam* [123] and the *Broyden-Fletcher-Goldfarb-Shanno (BFGS)* algorithm [91, 92, 93, 94]. The former uses memory across past updates to compute what can be understood as momentum in parameter space to-

¹Or a shallow PQC for a nonlocal cost function [121].

gether with an adaptive learning rate:

$$f_{\text{Adam}}(\mathbf{x}) = \eta \left(\sqrt[\odot]{v_t(\mathbf{x})} + \epsilon \right)^{-1} \odot \mathbf{m}_t(\mathbf{x}). \quad (3.2)$$

Here m_t and v_t are the first and second moment (power of the gradient) at the current step with exponentially decaying memory from past updates and ϵ is a regularization parameter. The latter algorithm also makes use of information from past update steps, but in contrast to the elementwise update rule of Adam it constructs an approximation to the Hessian of E from past evaluations of ∇E . This is paired with a line search to achieve improved, dynamic step sizes. The corresponding update rule reads

$$f_{\text{BFGS}}(\mathbf{x}) = \eta^*(\mathbf{x}) \tilde{H}_t^{-1} \mathbf{x}, \quad (3.3)$$

where $\eta^*(\mathbf{x})$ is the learning rate that is determined to be optimal and \tilde{H}_t is the approximate Hessian of E . An optimizer that uses geometric information about the quantum state is the *quantum natural gradient optimizer* (QNG) with the update rule

$$f_{\text{QNG}}(\mathbf{x}) = \eta (F_t + \epsilon)^{-1} \mathbf{x}, \quad (3.4)$$

where F_t is the *Fubini-Study metric tensor*, or *quantum Fisher information*, of $|\psi(\boldsymbol{\theta}_t)\rangle$ and ϵ again regularizes the update.

Popular optimizers that do *not* use the gradient include the *Nelder-Mead* (simplex) method [124], which is also widely used for other tasks, and *Rotosolve* [25, 28, 29, 30], which is a variant of coordinate descent which exploits that E is a (typically finite) Fourier series. Other optimizers aim at constructing (classical) local models of the objective function landscape, e.g. by fitting a quadratic function to evaluations of E as in *Modelgrad* [125] (quadratic in polynomial basis) or *quantum analytic descent* (QAD) [31] (quadratic in trigonometric polynomial basis).

Another type of optimization scheme breaks with the archetype of a VQA outlined above: in adaptive optimizers like *adaptive derivative-assembled pseudo-Trotter ansatz VQE* (ADAPT-VQE) [126], *Rotoselect* [30] or the recently proposed *generalized sequential quantum optimizer* [32] but also in the methods proposed in [127, 128] the structure of the PQC itself is modified during the optimization. This is done e.g. by selecting good candidates for extending the circuit from a pool of gates, by increasing the depth of a layered ansatz in a systematic manner, or by extending a circuit based on the Riemannian gradient flow of the special unitary group to which quantum circuits belong.

3.2 Contributions to the second publication


Here I describe my contributions to the enclosed publication, which was published in the journal *Physical Review Research* [15] and is freely available online, with functioning (hyper)links.

The entire project was predominantly carried out, and the manuscript written, by me, with the following exceptions: Christian Gogolin helped me with porting the created computer programs to a high-performance cluster by Covestro Deutschland AG and with executing them. Michael Kastoryano and Christian Gogolin wrote substantial parts of the abstract, introduction and conclusion of the manuscript. Of course the publication required many discussions and I received a lot of support and advice from both coauthors.

Compared to the main text, the following changes and additions to the notation are used in the publication. Additionally there are some specific notations used locally in single sections as defined therein.

Symbol	in main text	Meaning
$F^{(t)}$	—	Metric tensor at t th optimization step
$H^{(t)}$	\tilde{H}	Approximation of Hessian in BFGS algorithm
L_{\star}	$\prod_{k=1}^N R_{\star}^{(k)}$	Layer of Pauli rotations with Pauli word \star
t_x	—	Time scale in optimization cost
$\beta_{1,2}$	—	Decay rates in Adam optimizer
δ	—	Relative error of the variational energy
ε	ϵ	Regularization parameter in an optimizer
φ, ϑ	θ_k	Specific parameters in PQC

Avoiding local minima in variational quantum eigensolvers with the natural gradient optimizer

David Wierichs^{1,*}, Christian Gogolin^{1,2} and Michael Kastoryano^{1,3,4}¹*Institute for Theoretical Physics, University of Cologne, Germany*²*Covestro Deutschland AG, Kaiser Wilhelm Allee 60, 51373 Leverkusen, Germany*³*Amazon Quantum Solutions Lab, Seattle, Washington 98170, USA*⁴*AWS Center for Quantum Computing, Pasadena, California 91125, USA* (Received 14 May 2020; accepted 23 October 2020; published 17 November 2020; corrected 9 December 2020)

We compare the BFGS optimizer, ADAM and NatGrad in the context of VQES. We systematically analyze their performance on the QAOA ansatz for the transverse field Ising and the XXZ model as well as on overparametrized circuits with the ability to break the symmetry of the Hamiltonian. The BFGS algorithm is frequently unable to find a global minimum for systems beyond about 20 spins and ADAM easily gets trapped in local minima or exhibits infeasible optimization durations. NatGrad on the other hand shows stable performance on all considered system sizes, rewarding its higher cost per epoch with reliability and competitive total run times. In sharp contrast to most classical gradient-based learning, the performance of all optimizers decreases upon seemingly benign overparametrization of the ansatz class, with BFGS and ADAM failing more often and more severely than NatGrad. This does not only stress the necessity for good ansatz circuits but also means that overparametrization, an established remedy for avoiding local minima in machine learning, does not seem to be a viable option in the context of VQES. The behavior in both investigated spin chains is similar, in particular the problems of BFGS and ADAM surface in both systems, even though their effective Hilbert space dimensions differ significantly. Overall our observations stress the importance of avoiding redundant degrees of freedom in ansatz circuits and to put established optimization algorithms and attached heuristics to test on larger system sizes. Natural gradient descent emerges as a promising choice to optimize large VQES.

DOI: [10.1103/PhysRevResearch.2.043246](https://doi.org/10.1103/PhysRevResearch.2.043246)

I. INTRODUCTION

Variational quantum algorithms such as the variational quantum eigensolver (VQE) or the quantum approximate optimization algorithm (QAOA) [1] have received a lot of attention of late. They are promising candidates for gaining a quantum advantage already with noisy intermediate-scale quantum (NISQ) computers in areas such as quantum chemistry [2], condensed matter simulations [3], and discrete optimization tasks [4]. A major open problem is that of finding good classical optimizers which are able to guide such hybrid quantum-classical algorithms to desirable minima and to do this with the smallest possible number of calls to a quantum computer backend. In classical machine learning, the adaptive moment estimation (ADAM) optimizer [5] is among the most widely used and recommended algorithms [6,7], and has been one of the most important enablers of progress in deep learning in recent years. Such an accurate and versatile optimizer for quantum variational algorithms is yet to be found.

We are here mostly interested in variational algorithms for quantum many-body problems. To make progress towards finding an efficient and reliable optimizer for this domain, we concentrate on cost functions derived from typical quantum many-body Hamiltonians such as the transverse field Ising (TFIM) and the XXZ model (XXZM) for two reasons. First, their system size can be varied allowing us to systematically study scaling effects. Second, for *integrable* systems, the exact ground states are known and for the TFIM it is possible to construct ansatz classes for VQE circuits that provably contain the global minimum and can be simulated efficiently. Such systems thus allow us to distinguish between the performance of the optimizers and the expressiveness of the ansatz.

As a first result we show that the commonly used optimization strategies ADAM [8] and Broyden-Fletcher-Goldfarb-Shanno (BFGS) [9–18] both run into convergence problems when the system size of a VQE is increased. This happens already for system sizes within the reach of current and near future NISQ devices, which underlines the importance to a systematic search for suitable optimization strategies. The performance of ADAM is shown to depend strongly on the learning rate (the scaling prefactor determining the size of parameter update steps) via multiple effects and the number of epochs required for convergence increases fast with the problem size. Convergence can be improved but only with an expensive fine-tuning of the hyperparameters.

We then study the performance of an optimization strategy known as the quantum natural gradient or NatGrad [19–21]

*wierichs@thp.uni-koeln.de

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/). Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

and introduce Tikhonov regularization to the classical processing step in the VQE [22]. The key characteristic of NatGrad is that it uses the canonical metric on Hilbert space, the *Fubini-Study metric*, to determine improved updates to the variational parameters. While the proposal of NatGrad for VQES includes numerical experiments comparing it to several established optimizers as well as an ADAM variant that uses the natural gradient [19], the presented results extend this comparison in multiple directions: First, we include the BFGS optimizer which is widely used throughout the VQE literature. Second, different models are considered here, each of which is more complex than the example in Ref. [19] as they contain more than one two-qubit term. This extension is essential as is visible by the fact that no qualitative difference between the diagonal approximation of the Fubini-Study metric and the full metric was seen in Ref. [19] but the optimization problems presented here are not solvable with the diagonal approximation at all. Third, we extend the considered problem size from maximally 11 qubits to 40 qubits for the TFIM and 14 qubits for the XXZM. Fourth, our analysis includes the robustness of the investigated optimization algorithms regarding overparametrization, which can be expected to be of relevance in applications. We find that NatGrad does consistently find a global optimum for the largest system sizes we test (40 qubits) and requires significantly fewer epochs to do so than ADAM (in the cases where ADAM converges at all).

Our second set of results concerns the effect of overparametrization in VQES. We study the impact of adding redundant layers to the ideal circuit ansatz. This overparametrization not only increases the optimization cost, it actually appears to make finding the optimum significantly harder. The BFGS algorithm but also the ADAM optimizer, designed to thrive on additional degrees of freedom, fail frequently in this setting. This cannot easily be mitigated by increasing the epoch budget and reducing the learning rate of the ADAM optimizer. While also affected, NatGrad shows much higher resilience against this effect, compensating its higher cost per epoch with a higher chance to succeed. In applications on a relevant scale the circuit ansatz cannot be expected to be minimal making this resilience essential for the success of an optimizer for VQES. This also demonstrates the importance of understanding the role of redundant degrees of freedom in the variational class. When restricting the additional degrees of freedom to the symmetry sector of the model, ADAM does not profit from overparametrization and the BFGS optimizer performs worse whereas NatGrad reliably converges globally.

Our results are in sharp contrast to the usually very good performance of the ADAM optimizer and related (stochastic) gradient descend based techniques in the optimization of classical neural networks. A possible explanation for this good performance in usually overparametrized settings is the following: For common activation functions and random initialization, increasing overparametrization tends to transform local minima into saddle points [23,24]. The optimizer then mainly needs to follow a deep and narrow valley with comparably flat bottom to find a global minimum. The ADAM optimizer is perfectly suitable to pursue this path as it has individual learning rates per parameter that also take into account the average of recent updates (see Sec. II B for details). In this

way it avoids side-to-side oscillations in the valley and can build up momentum to slide down the relatively flat bottom of the valley.

The energy landscapes of typical variational quantum algorithms however look very different. First, having deep and wide circuits with many parametrized gates is prohibitive on NISQ computers, which excludes overparametrization as a tool to make the variational space more accessible. Second, the variational parameters usually feed into gates as prefactors of exponentials of Pauli words and thus the cost function is ultimately a combination of trigonometric functions of the parameters. It appears that NatGrad is able to effectively use the information about the ansatz class to navigate the resulting energy landscape with many local minima. Third, it is known that large parts of the parameter space form so-called barren plateaus with very small gradients [25]. A random initialization of the parameters in reasonably deep VQES is thus almost certainly going to leave one stuck in such a plateau. Of course this also implies that one must prevent the optimizer from jumping to a random location in parameter space during optimization. This can be achieved in NatGrad by inhibiting unsuitably large steps by means of Tikhonov regularization. Finally, due to the small number of variational parameters in VQE, the added (classical) computational cost of inverting the Fubini-Study metric, which is used to determine the parameter updates (see Sec. II B), is negligible as compared to the cost of sampling from the quantum backend. This fact, combined with the highly correlated nature of the learning landscape in quantum many-body problems [26], might render second-order methods such as NatGrad more amenable to quantum than to classical settings, where samples are cheap, but there are many variational parameters.

In order to generalize our results, we consider the XXZM together with the Trotterized time evolution operator as circuit ansatz. Indeed we find BFGS to experience the same difficulties in high-dimensional parameter spaces and ADAM to exhibit a similar behavior of the required number of epochs as for the TFIM. The performance of NatGrad mostly is as reliable for this model as for the TFIM.

A. Informal summary of the results

Our main results are the following. First, NatGrad is the most reliable optimization method. This is due to the capability to maneuver high-dimensional search spaces driven by the Natural gradient and its relatively high resilience to overparametrization, both within and outside of the symmetry sector of the solution. The BFGS optimizer fails to navigate towards global minima in large spaces and in the presence of redundant degrees of freedom even in small systems. ADAM suffers significantly from symmetry-breaking overparametrization and is not able to use additional degrees of freedom *within* the symmetry sector for improved performance.

Second, NatGrad has larger quantum computation cost per epoch than the other algorithms by design but the improved learning strategy remedies this via small epoch counts to convergence. Meanwhile, BFGS takes few epochs to convergence at low cost per epoch but produces low-quality results, including local minima and positions in very shallow plateaus

in the cost function. ADAM also has low cost per epoch but for large and complicated problems it takes many epochs to converge and this duration is hard to predict.

Third, the above properties generalize to a certain level. That is, the failure of BFGS and the rapid increase in cost of ADAM appeared at similar parameter counts for different models and ansatz circuits and NatGrad tackled both spin chain systems successfully.

The practical conclusions from the presented work are twofold: On one hand, when solving the ground state energy problem with a VQE on an application-relevant scale, NatGrad appears to be the optimizer of choice for the classical processing step. This holds for both investigated spin chain models and, given the asymptotically vanishing cost overhead of NatGrad for Hamiltonians with many noncommuting terms, probably even more so for quantum chemical systems.

Finally, we observed decisive differences in the cost function landscape and optimizer performance from classical machine learning beyond obvious deviations like the dimension of the parameter space. This implies that heuristics and established methods from machine learning require new evaluation and additional research in order to optimally utilize them for VQES.

II. METHODS

A. Variational quantum eigensolver

The framework of our work is the VQE, a proposal to use parametrized circuits on a quantum computer in combination with classical optimization routines to prepare the ground state of a target Hamiltonian H . In the first part of a VQE, one constructs a quantum circuit that contains parametrized gates. Given input parameters θ for the circuit, a quantum computer can then prepare the corresponding ansatz state and measure an objective function, chosen to be the energy of the Hamiltonian

$$E(\theta) := \langle \psi(\theta) | H | \psi(\theta) \rangle \quad (1)$$

and for benchmark problems with known ground state energy E_0 , the relative error δ can be calculated as

$$\delta(\theta) := \frac{E(\theta) - E_0}{|E_0|}. \quad (2)$$

Additionally one can prepare modified versions of the circuit to determine auxiliary quantities like the energy gradient in the parameter space [27]. The second part of the VQE scheme is an optimization strategy on a classical computer which is granted access to the quantum black box just constructed. In the most straightforward scenario this is a black box minimization scheme, but using auxiliary quantities, more sophisticated optimization methods can be realized as well.

There are two main theoretical challenges for successfully applying VQE. First, the construction of a sufficiently complex, but not overly expensive, circuit that gives rise to an ansatz class containing the ground state-*expressivity*. Second, the choice of a suitable optimizer that is able to search for the ground state within the created parameter space *efficiency*. The two challenges are often seen as independent, but explicit algorithms using information gathered about the variational space during optimization phases for adjusting the ansatz have

been proposed as well, some of which are inspired by concrete applications in quantum chemistry or by evolutionary strategies [8,16,28,29].

We now establish some notation for the general VQE setting where we assume the most common objective: Finding the ground state energy of a Hamiltonian H . Starting from an initial product state $|\bar{\psi}\rangle$, we apply parametrized unitaries $\{U_j(\theta_j)\}_{1 \leq j \leq n}$ to construct the ansatz state

$$|\psi(\theta)\rangle := \prod_{j=n}^1 U_j(\theta_j) |\bar{\psi}\rangle. \quad (3)$$

The parameters are typically initialized randomly close to zero to avoid the barren plateau problem [25]. For this work, the unitaries are going to be translationally invariant layers of one- or two-qubit rotations; consider, for instance,

$$L_{zz}(\theta_j) := \prod_{k=1}^N \exp \left[-\frac{i\theta_j}{2} Z^{(k)} Z^{(k+1)} \right] \quad (4)$$

$$= \exp \left[-\frac{i\theta_j}{2} \sum_{k=1}^N Z^{(k)} Z^{(k+1)} \right], \quad (5)$$

where we identified the qubits with index 1 and $N+1$, i.e., we adopt periodic boundary conditions. The ordering of the gates within a layer is not relevant because they commute but for convenience we write them such that terms acting on the first qubits are applied first. $Z^{(k)}$ is the Pauli Z operator acting on the k th qubit and we tacitly assume the tensor product between operators that act on distinct qubits as well as the missing tensor factors of identities. Compared to proposed ansatz circuits that employ full Hamiltonian time evolution $\exp(-i\theta H)$ (see Sec. II A 1 a), such a layer is rather easily implemented on present quantum machines because it only requires linear connectivity and one type of two-qubit rotation. There have been many proposed circuits to generate ansatz classes for a variety of problems, all of which can be boiled down to combining rotational gates and possibly other fixed gates such as the CNOT or SWAP gate (see Sec. II A 1). For the presented optimization methods the derivatives with respect to the variational parameters $\{\theta_j\}_j$ are important and for the above example we observe the special structure of translationally symmetric layers of Pauli rotation gates:

$$\frac{\partial}{\partial \theta_j} L_{zz}(\theta_j) = \left(-\frac{i}{2} \sum_{k=1}^N Z^{(k)} Z^{(k+1)} \right) L_{zz}(\theta_j). \quad (6)$$

The derivative only produces an operator-valued prefactor, and all prefactors can be summarized because the single gates commute. While the basic gates composing a unitary $U_j(\theta_j)$ typically take the form of (local) Pauli rotations, the full unitary often is more complex than the above layer and in particular the terms in U_j do not need to commute. However, the structure of rotations enables us in general to evaluate required expressions involving derivatives on a quantum computer, either via measurements of rotation generators or via ancilla qubit schemes.

1. A selection of ansatz classes

Among the ansatz families proposed in the literature we present the following which are used frequently and are directly connected to this work:

(a) *QAOA*. The quantum approximate optimization algorithm was first proposed by Farhi, Goldstone and Gutmann [1] in 2014 for approximate solutions to (classical) optimization problems by mapping them to a spin chain Hamiltonian. The algorithm looks similar to adiabatic time evolution methods with an inhomogeneous time resolution, which is rather coarse for typical circuit depths. A lot of work has been put into proving properties of the QAOA both in general and for certain problem types, including extensions to quantum cost Hamiltonians [30–33]. At the same time the algorithm has been refined, extended, and characterized on the basis of heuristics and numerical experiments, gaining insight into its properties beyond rigorous statements [14,34–37].

The QAOA circuit is constructed as follows. For a *cost Hamiltonian* H_S and a so-called *mixing Hamiltonian* H_B one alternately applies the unitaries $\exp(-i\vartheta_j H_S)$ and $\exp(-i\varphi_j H_B)$ p times, giving rise to a VQE ansatz class with “time” parameters $\{\vartheta_j, \varphi_j\}_{1 \leq j \leq p}$. Originally, the system Hamiltonian would encode a classical optimization problem and thus be diagonal while the mixing Hamiltonian was chosen to be off-diagonal and specifically has been kept fixed to the original $H_B = \sum_{k=1}^N X^{(k)}$ for many investigations. However, new choices of mixers have been proposed and investigated as well, giving rise to the more general quantum alternating operator ansatz (QAOa) [15,37,38].

Note that for quantum systems, the terms comprising the Hamiltonian H_S do not commute in general such that very large gate sequences would be necessary to realize the exact QAOA approach including $\exp(-i\vartheta H_S)$. In practice these blocks commonly are broken up in a Trotter-like fashion instead, yielding circuits that are implemented more readily but deviating from the original ansatz. For the TFIM, such a modified QAOA ansatz has been studied intensively [14,34,35] and we are going to use it as a starting point for our investigations.

(b) *Adaptive Ansätze*. Most prominently for this type of *Ansätze*, ADAPT-VQE tackles both the construction of a suitable ansatz class and the optimization within the constructed parameter space [16].

Instead of a fixed ansatz circuit layout, ADAPT-VQE takes a pool of gates as input and iterates the two steps of the VQE scheme: After rating all gates the most promising one is appended to the circuit (construction) and afterwards all the circuit parameters are optimized (minimization). The optimized parameters from the previous step are then used for both the rating of the gates for the next construction step and the initialization for the following optimization, where newly added gates are initialized close to the identity. For both the concept of allowed gates and the gate rating criteria, there are multiple options and we refer the reader to [16,28] for more detailed descriptions.

Besides ADAPT-VQE, multiple other methods which grow the ansatz circuit in interplay with the optimization have been proposed and demonstrated, including ROTOSELECT [8] and EVQE [29]. These demonstrations include the solution of five-qubit spin chains and small molecules (lithium hydride,

beryllium dihydride, and a hydrogen chain) to chemical precision using simulations with and without sampling noise or quantum hardware.

We will not be using any adaptive scheme in our work, but our results on stability and overparametrization raise serious doubts as to the reliability of any adaptive ansatz method (see Sec. III B).

B. Optimizers

A variety of optimizers have been used in the context of variational quantum algorithms. These optimizers are inspired by classical machine learning and can be sorted according to the order of information required about the cost function. Zeroth-order or direct optimization methods only evaluate the function itself, first-order methods need access to the gradient, and second-order optimization need access to the Hessian of the cost function, or some other metric reflecting the local curvature of the learning landscape. As direct optimization is not scalable to problem sizes of relevance we do not include it in our studies. A parameter update step of the optimizer—corresponding to one iteration at the algorithmic level—is called an *epoch* and corresponds to one execution of the update rules described in the following [see Eqs. (7), (10), (11), and (13)]. Most optimization algorithms have one or more hyperparameters, the most common being the learning rate η , which is a scalar prefactor rescaling the parameter update at each epoch.

1. First-order gradient descent

Optimization techniques using the gradient of the cost function are at this point the most widely used in machine learning. Starting from the simple gradient descent method that updates the parameters according to the gradient and a fixed learning rate, a whole family of minimization strategies has been developed. The improved routines are inspired by physical processes like momentum, based on heuristics like adaptive learning rate schedules, or a smart processing of the gradient information as in the Nesterov accelerated gradient. A review of this development can be found e.g., in Ref. [7], here we just present the first-order method we are going to use, the ADAM optimizer.

ADAM, which was proposed in 2014 [5], is probably the most prevalent optimization strategy for deep feed-forward neural networks [6] and has been used in VQE settings as well [8]. For completeness, we briefly outline the ADAM optimizer: Given the cost function $E(\theta)$, where θ recollects all variational parameters, a starting point $\theta^{(0)}$ and a learning rate η , Gradient Descent computes the gradient $\nabla E(\theta^{(t)})$ at the current position and accordingly updates the parameters rescaled by η :

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla E(\theta^{(t)}). \quad (7)$$

As the gradient points in the direction of steepest ascend, the parameter update is directed towards the steepest descend of the cost function and for η small enough, the convergence towards a minimum can be understood intuitively. Small learning rates yield slow convergence which increases the cost of the optimization whereas choosing η too large leads to overshooting and oscillations which might prevent con-

vergence. Furthermore, although the optimizer will diagnose convergence to a minimum due to a vanishing gradient, it cannot distinguish between local and global minima.

In order to fix both issues, i.e., the need for an optimally scheduled learning rate and the liability of getting stuck in local minima, various improvements have been proposed and ADAM uses several of these upgrades. The first feature is an *adaptive, componentwise learning rate*, which was introduced in ADAGRAD [39] and improved in RMSPROP [40] to avoid suppressed learning. The second feature ADAM uses is *momentum*, which is inspired by the physical momentum of a ball in a landscape with friction. This is realized by reusing past parameter upgrades weighted with an exponential decay towards the past and enables ADAM to overcome some local minima. The final form of the ADAM algorithm is as follows: Initialize with hyperparameters $\{\eta, \beta_1, \beta_2, \varepsilon\}$, momentum $m^{(0)} = 0$, average squared gradient $v^{(0)} = 0$ and initial position $\theta^{(0)}$. At the t th step, compute the gradient and update the momentum and the cumulated squared gradient as

$$m^{(t)} = \frac{\beta_1 - \beta_1^t}{1 - \beta_1^t} m^{(t-1)} + \frac{1 - \beta_1}{1 - \beta_1^t} \nabla E(\theta^{(t)}), \quad (8)$$

$$v^{(t)} = \frac{\beta_2 - \beta_2^t}{1 - \beta_2^t} v^{(t-1)} + \frac{1 - \beta_2}{1 - \beta_2^t} (\nabla E(\theta^{(t)}))^{\odot 2}, \quad (9)$$

where $x^{\odot 2}$ denotes the elementwise square of a vector x . The parameter update then is computed from these updated quantities via

$$\theta^{(t+1)} = \theta^{(t)} - \frac{\eta}{\sqrt{v^{(t)} + \varepsilon}} m^{(t)} \quad (10)$$

with the square root of $v^{(t)}$ taken elementwise. Besides the learning rate η , we identify the hyperparameters β_1 and β_2 as exponential memory decay factors of m and v respectively and the small constant ε as regularizer, which avoids unreasonably large updates in flat regions and division by zero at initialization or for irrelevant parameters.

Because of the advanced features that ADAM uses, it has been very successful at many tasks and even though there are applications for which more basic gradient-based optimizers can be advantageous, we choose ADAM to represent the family of local first-order optimizers.

2. BFGS optimizer

The second optimizer we look at is the BFGS algorithm, which was proposed by its four authors independently in 1970 [9–12]. Using first-order resources only it approximates the Hessian of the cost function and performs global line searches in the direction of the gradient transformed by the Hessian inverse. Therefore it is a global quasi second-order method using local first-order information and its categorization is not obvious. The algorithm is initialized with the starting point $\theta^{(0)}$ and a first guess for the approximate Hessian $H^{(0)}$ of the cost function E , which usually is set to the identity. At each step of the optimization one determines the gradient, computes the direction

$$n^{(t)} = H^{(t)-1} \nabla E(\theta^{(t)}) \quad (11)$$

and performs a line search on $\{\theta^{(t)} + \eta n^{(t)} | \eta \in \mathbb{R}\}$ which yields the optimal update in that direction and can optionally

be restricted to a bounded parameter subspace. Given the new point in parameter space, $\theta^{(t+1)}$, the change in the gradient $D^{(t)} = \nabla E(\theta^{(t+1)}) - \nabla E(\theta^{(t)})$ is calculated and used to update the approximate Hessian via

$$H^{(t+1)} = H^{(t)} + \frac{D^{(t)} D^{(t)T}}{\eta^{(t)} D^{(t)T} n^{(t)}} - \frac{H^{(t)} n^{(t)} n^{(t)T} H^{(t)}}{n^{(t)T} H^{(t)} n^{(t)}}.$$

As the parameter updates are found via line searches, the BFGS algorithm is not strictly local but due to its use of local higher-order information, the global search is much more efficient than direct optimization. The method has been successful in many applications and currently is of widespread use for VQES [13–18].

3. Natural gradient descent

The third optimization strategy we use is the NatGrad [19–21], which due to its increased cost per epoch is not adopted very often in machine learning settings itself but is connected to some successful methods. As an example, stochastic reconfiguration which is closely related to NatGrad [41] recently has been shown to work well for training restricted Boltzmann machines (RBMs) to describe ground states of spin models [42]. Despite this success, the insights into why and under which conditions the method works remain limited and recent work has been put into understanding the learning process for the mentioned application of RBMs and the natural gradient descent [26]. Before discussing NatGrad and its role in the VQE setting, we outline its update rule. Given a starting point $\theta^{(0)}$ and a learning rate η , a step is performed by first constructing the Fubini-Study metric of the ansatz class

$$(F^{(t)})_{ij} := \Re \epsilon \{ \langle \partial_i \psi^{(t)} | \partial_j \psi^{(t)} \rangle \} - \langle \partial_i \psi^{(t)} | \psi^{(t)} \rangle \langle \psi^{(t)} | \partial_j \psi^{(t)} \rangle \quad (12)$$

at the current position and then updating the parameters via

$$\theta^{(t+1)} = \theta^{(t)} - \eta F^{(t)-1} \nabla E(\theta^{(t)}), \quad (13)$$

where we abbreviated $|\psi^{(t)}\rangle := |\psi(\theta^{(t)})\rangle$ and $|\partial_i \psi^{(t)}\rangle := \frac{\partial}{\partial \theta_i} |\psi(\theta^{(t)})\rangle$.

The Fubini-Study metric is the quantum analog of the Fisher information matrix in the classical natural gradient [20]. It describes the curvature of the ansatz class rather than the learning landscape, but often performs just as well as Hessian based methods. In order to avoid unreasonably large updates caused by very small eigenvalues of F in standard natural gradient descent η has to be chosen very small for an unpredictable number of initial learning steps. Alternatively one can use *Tikhonov* regularization which amounts to adding a small constant to the diagonal of F before inverting it, also see Sec. II E.

Even though NatGrad is simple from an operational viewpoint, it is epochwise the most expensive optimizer of the three presented here (also see Sec. II C). This is due to the fact that it not only uses the gradient but, in order to construct the (Hermitian) matrix F for n parameters, one also needs to evaluate $\frac{1}{2}(n^2 + 3n)$ pairwise overlaps of the set $\{|\psi\rangle, |\partial_1 \psi\rangle, \dots, |\partial_n \psi\rangle\}$ (all but $\langle \psi | \psi \rangle = 1$). Depending on the gates in the ansatz circuit, each of these overlaps requires

at least one and possibly many individual circuit executions. For circuits containing \tilde{n} simple one- or two-qubit Pauli rotation gates, the number of circuits required is $\frac{1}{2}(\tilde{n}^2 + 3\tilde{n})$, independent of the number of shared parameters. Symmetries of the circuit may reduce the number of distinct terms in which case fewer quantum machine runs suffice.

Taking the j th parametrized unitary to have K_j Hermitian generators P_{j,k_j} , e.g., Pauli words up to prefactors $\{c_{j,k_j}\}$, the factors in the second expression of F take the shape of an expectation value [see also Eq. (6)]

$$\langle \psi | \partial_j \psi \rangle = \langle \bar{\psi} | \prod_{l=1}^{j-1} U_l^\dagger \left[\sum_{k_j=1}^{K_j} c_{j,k_j} P_{j,k_j} \right] \prod_{l=j-1}^1 U_l | \bar{\psi} \rangle. \quad (14)$$

The first term in Eq. (12) requires slightly more complex circuits using one ancilla qubit and a depth which depends on the indices of the matrix entry [13,17,43,44]. Both for simulation work and for applications on real quantum machines, the construction of the Fubini matrix is expected to take much more time than inverting it—in sharp contrast to typical classical machine learning problems. Given the scaling of the number of required circuits above and the fact that for a fixed number of qubits the depth has to grow at least linearly with the number of parameters, an asymptotic scaling of $\mathcal{O}(\tilde{n}^3)$ is a lower bound for the construction of the full matrix. Standard matrix inversion algorithms do not only show smaller or equal scaling but also exhibit as prefactor the time cost of a FLOP whereas computing the matrix elements scales with prefactors based on sampling for expectation values.

As the number of parameters in a typical VQE circuit is considerably smaller than in neural networks and the circuit chosen in this work exhibits beneficial symmetries, the high cost of the method are expected to be less problematic for our setting and bearable for VQE applications. Indeed, there have been some demonstrations of the natural gradient descent and the imaginary time evolution for small VQE instances [19,45,46] as well as comparisons to standard gradient descent methods and imaginary time evolution for one- and two-qubit systems [47]. Inspired by the classical machine learning context and aiming for reduced cost, modifications of natural gradient descent have been proposed such as a (block) diagonal approximation to the Fubini-Study matrix [19]. We will later show that such simplifications have to be performed with caution and can disturb the optimization.

Finally we want to mention optimizers that treat the variational parameters sequentially, updating only one parameter at each epoch. While such algorithms can be designed to use information about the ansatz class and use the parametrization directly (see, e.g., Ref. [48]), we expect them to behave differently than optimizers updating all parameters simultaneously on which we focus our studies.

C. Optimization cost

To make a fair comparison between the optimization schemes, we briefly lay out the scaling of the required operations and the resulting cost per epoch.

We will use the following notation during the comparison. There are n variational parameters in the circuit, K_H terms in the Hamiltonian and on average $K = \sum_{j=1}^n K_j/n$ Pauli gener-

ators per variational parameter, with an average of N_M samples required for each expectation value. In practice, one of course would measure whole sets of operators both from the Hamiltonian and from the Pauli generator set simultaneously, such that K and K_H essentially are numbers of bases in which measurements are required. For entries of the Fubini matrix, we assume N_a samples for sufficiently precise measurements, which has been shown to be smaller than N_M numerically;¹ for further discussion see Ref. [45]. Finally, we introduce the timescales

$$t_x := \frac{d}{x} t_{\text{gate}} + t_{\text{wrap}} \quad (15)$$

for integers x that capture effects of averaging the depths of used auxiliary circuits. t_{gate} is the time required by each layer of parallelized gates and t_{wrap} includes the needed time for initializing and measuring the quantum register. Evaluating the gradient of the energy function can be done with different methods yielding a trade-off between precision and cost. On one hand, the analytic gradient can be evaluated up to measurement precision at the expense of an ancilla qubit and a scaling prefactor Kn . On the other hand there is the standard finite difference method, which can be performed symmetrically, asymmetrically or via simultaneous perturbation stochastic approximation (SPSA) [52], with cost prefactors $2n$, $n+1$ and 2 , respectively. This means that robustness to imprecise gradients in general is a relevant property of any optimization scheme used for VQEs because these gradients are much cheaper to evaluate. Computing the Fubini-Study metric requires two terms and although the measurement cost scales with $\mathcal{O}((Kn)^2)$ for the first and with $\mathcal{O}(Kn)$ for the second, we keep both terms in the overall cost scaling because the VQE regime implies moderate values of Kn .

For the scalings presented in Table I, we assume a homogeneous distribution of the variational gates in the circuit and that similar numbers of samples N_M are required to measure expectation values of the Hamiltonian terms within one basis and each derivative for all gradient methods.

For the full optimization algorithms, the cost are given per epoch as we do not have access to generic scaling of epochs to convergence. Using the cost per epoch one can rescale the optimization cost from epochs to estimated run time on a quantum computer beyond estimates that are based on the classical simulation run times. For the BFGS algorithm, we can not predict the number γ of energy evaluations that are required for the line searches but our numeric experiments and the linear scaling of the cost for nonSPSA gradients suggest that it can be neglected as compared to the gradient computation.

For the quantum run time scalings shown in Figs. 3, 5 and 8, we give the time in units of $t_{\text{eval}} = N_M K_H t_1$, assumed $N_M/N_a \approx 10$ [45] and approximated $t_1 \approx t_2 \approx t_3$.

¹After submission of this manuscript, analytic bounds on the relative measurement cost of the gradient and the Fubini matrix have been presented in Ref. [60], underlining the numerical results in Ref. [45].

TABLE I. Cost on a quantum computer for selected VQE optimization methods and their subroutines. The optimizer cost are given per epoch, enabling us to compare the techniques beyond their simulation times with different scaling. We neglected terms which are small for $d, n \gg 1$ and used the timescales t_x defined in Eq. (15). The remaining scaling parameters $\{N_M, K, K_H, N_a\}$ are defined in the paragraph above Eq. (15).

	Operation	Quantum cost	
t_{eval}	Energy evaluation	$N_M K_H t_1$	Depending on measurement bases
	$\left\{ \begin{array}{l} \text{Analytic gradient} \\ \text{Numeric gradient (sym.)} \\ \text{Numeric gradient (asym.)} \end{array} \right.$	$(Kn)N_M K_H t_1$	Ancilla qubit required
		$2(Kn)N_M K_H t_1$	Parameter shift rule [27,49,50]
		$2nN_M K_H t_1$	Sensitive to noise
t_{grad}	SPSA gradient	$2N_M K_H t_1$	Additional samples improve precision
t_{Fubini}	$\left\{ \begin{array}{l} \text{Fubini matrix} \\ \text{BFGS} \end{array} \right.$	$(Kn)^2 N_a t_3 + (Kn) N_a t_2$	Ancilla qubit required
		$2(Kn)^2 N_a t_3 + (Kn) N_a t_2$	via projective measurements [51]
	ADAM	$t_{\text{grad}} + \gamma t_{\text{eval}}$	$\gamma = \mathcal{O}(n^{0 \leq \gamma < 1})$ expected
	NatGrad	t_{grad}	
		$t_{\text{grad}} + t_{\text{Fubini}}$	Cost for inverting F can be neglected

1. Epoch count and quantum run time

When comparing the cost of optimizers that access the same resources, the epoch count N_{epoch} is a sufficient figure of merit. The presented algorithms, however, use distinct sets of quantities such that the quantum run time t_Q is a better measure to compare them. It is important to keep the system specific scaling of computing the gradient and the Fubini matrix in mind. The presented spin chain systems and ansätze with translation symmetry contain $\mathcal{O}(1)$ terms to be measured in the Hamiltonian leading to $\mathcal{O}(n)$ cost for the gradient for n parameters in the ansatz. The layered structure and the symmetry of the used circuits leads to $\mathcal{O}(n^3)$ measurements for the Fubini matrix, generating a large overhead in NatGrad. On the other hand, chemical Hamiltonians, which constitute an important application of VQES, contain $\mathcal{O}(N^4)$ terms for N electrons, which can be measured roughly in $\mathcal{O}(N^3)$ bases [53,54] implying cost $\mathcal{O}(nN^3)$ of measuring the gradient. Meanwhile, typical circuit types contain gates with a moderate constant number of generators, leading to $\mathcal{O}(n^2)$ cost of measuring the Fubini matrix, which is considerably smaller than $\mathcal{O}(nN^3)$ for any realistic circuit depth.

In summary, we consider the quantum run time t_Q to deliver a more meaningful comparison between different optimizers but report N_{epoch} as well to characterize the algorithms in a less system-dependent measure. Assuming the epoch count to behave similarly in various VQE landscapes, this enables us to estimate the relative cost of the optimizers when applied to, e.g., quantum chemistry.

D. Models

1. Transverse field Ising model

Our main model is the TFIM on a spin chain with periodic boundary conditions (PBC). Its Hamiltonian reads

$$H_{\text{TFI}} = H_S + H_B := - \sum_{k=1}^N Z^{(k)} Z^{(k+1)} - t \sum_{k=1}^N X^{(k)}, \quad (16)$$

where we identify the sites 1 and $N + 1$ because of the PBC and t is the transverse field. For $t = 0$, the system is the classical Ising chain, which is also called ring of disagrees and

is a special case of the MAXCUT problem [1,34]. For $t \neq 0$, the problem is no longer motivated by a classical optimization task and for the critical point $t = 1$, the ground state exhibits long-ranged correlations.

The ground state of the TFIM is found analytically by mapping it to a system of noninteracting fermions, where the transformed Hamiltonian can be diagonalized exactly [55]. The translational invariance of the Hamiltonian is crucial for this step and it will be important that only a small number of different (Pauli) terms can be mapped to *noninteracting* fermions simultaneously. We show the explicit computation via the Jordan-Wigner transformation in Appendix A, it can also be found in, e.g., Ref. [34]. Here we summarize the action of the mapping on the terms in the Hamiltonian which also generate the QAOA circuit [see Eq. (20) for the definition of α_q]:

$$\sum_{k=1}^N Z^{(k)} Z^{(k+1)} \longrightarrow (N - 2r) + 2 \bigoplus_{q=1}^r [\cos \alpha_q Z + \sin \alpha_q Y], \quad (17)$$

$$\sum_{k=1}^N X^{(k)} \longrightarrow (N - 2r) + 2 \bigoplus_{q=1}^r Z \quad (18)$$

where the expressions on the right are understood in a *fermionic operator basis* and the number of fermions is given by $r = \lfloor \frac{N}{2} \rfloor$. The ground state of H_{TFI} is just the product of the single-fermion ground states in momentum basis and we can write out the state and its energy as

$$E_0 = -E' - 2 \sum_{q=1}^r \sqrt{1 + t^2 + 2t \cos \alpha_q} \quad \text{with} \quad (19)$$

$$\alpha_q := \begin{cases} (2q - 1)\pi/N & \text{for } N \text{ even} \\ 2q\pi/N & \text{for } N \text{ odd} \end{cases}, \quad (20)$$

$$E' := \begin{cases} 0 & \text{for } N \text{ even} \\ 1 + h & \text{for } N \text{ odd} \end{cases}. \quad (21)$$

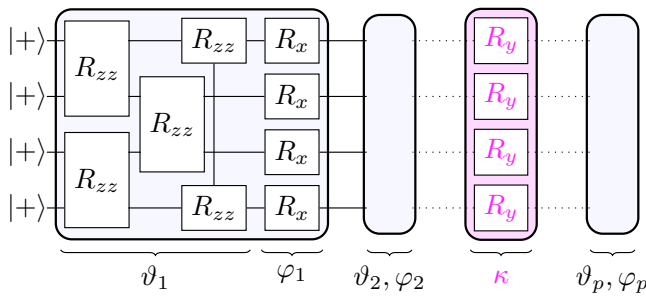


FIG. 1. The QAOA circuit for the TFIM on 4 qubits including an overparametrizing layer $L_y(\kappa)$. The first numerical experiment is performed without any Pauli Y layers L_y and in the second experiment overparametrization is investigated using one or two such layers.

Because of the free fermion mapping, we can not only obtain the exact ground state of the system but also justify the success of the modified QAOA circuit for the TFIM. As mentioned in Sec. II A 1 a, the original QAOA proposal would use the system Hamiltonian and a mixing term as generators for the parametrized gates. For the TFIM, however, separating the nearest-neighbour interaction terms H_S from the transverse field terms H_B recombines the latter with the mixing unitary next to it absorbing one variational parameter per block. The resulting parametrized circuit contains two types of translationally invariant layers, $L_x(\varphi)$ and $L_{zz}(\vartheta)$, of one- and two-qubit rotation gates, respectively. Starting in the ground state of H_B , that is $|\bar{\psi}\rangle = |+\rangle^{\otimes N}$, we alternately apply these two layers p times starting with L_{zz} . The resulting QAOA circuit is shown in Fig. 1. In the free fermion picture, this translates to $|\bar{\psi}\rangle = |0\rangle^{\otimes r}$ and to rotations of the r fermionic states about the z axis (L_x) and an axis $e_q = (0, \sin \alpha_q, \cos \alpha_q)^T$ which depends on the fermion momentum q (L_{zz}).

For $t = 0$, one can prove that this circuit can prepare the ground state exactly if and only if $p \geq r$ [14], whereas for the case $t \neq 0$ only numerical evidence and a nonrigorous explanation support this claim [35]. This explanation compares the number of independent parameters, $2p$ to the number of constraints from fixing the state of r free fermions, $2r$. While solvability would be implied for a linear system, the given problem is nonlinear and the argument remains on a nonrigorous level.

Finally, the equivalence to a system of free fermions has a practical implication for our simulations of the QAOA circuit: Storing the state of r free fermions just requires memory for $2r$ complex numbers. Applying the entire circuit needs $2pr$ two-dimensional matrix-vector multiplications, which is contrasted by $2pN$ matrix-vector multiplications in 2^N dimensions for a full circuit simulation in the qubit picture. Using the fermionic basis for numerical simulations, results on the VQE optimization problem for up to $N = 200$ and $p > 120$ have been obtained for $t = 0$ [14].

2. Heisenberg XXZ model

As a second model we consider the 1D XXZM with PBC which is defined by

$$H_{\text{XXZ}} = \sum_{k=1}^N [X^{(k)}X^{(k+1)} + Y^{(k)}Y^{(k+1)} + \Delta Z^{(k)}Z^{(k+1)}]. \quad (22)$$

Δ is the anisotropy parameter. As in the TFIM, the sites 1 and $N + 1$ are identified. The Bethe ansatz reduces the eigenvalue problem for the XXZM to a system of $N/2$ nonlinear equations that can be solved numerically with an iterative scheme [56,57]. This results in polynomial cost for computing the ground state energy but does not yield a simple ansatz class to construct the ground state on a quantum computer or a simulation scheme of reduced complexity.

We therefore use the XXZM as a second benchmark which models the application case more closely: We do not know a finite gate sequence that contains the ground state but instead employ circuits composed of symmetry-preserving layers which we found to be relatively successful in experiments. The ansatz we choose is the first-order Trotterized version of the unitary time evolution with the system Hamiltonian applied to an antiferromagnetic ground state:

$$|\psi(\theta)\rangle = \prod_{j=L}^1 L_{zz}(\vartheta_j)L_{yy}(\kappa_j)L_{xx}(\varphi_j)|\bar{\psi}\rangle, \quad (23)$$

$$|\bar{\psi}\rangle = \frac{1}{\sqrt{2}}(|01\rangle^{\otimes N/2} \pm |10\rangle^{\otimes N/2}), \quad (24)$$

where we only treat even N and $|\bar{\psi}\rangle$ is chosen symmetric under translation for $(N \bmod 4) = 0$ and antisymmetric for $(N \bmod 4) = 2$ in anticipation of the exact solution via the Bethe ansatz. We found this circuit to be more successful at finding the ground state than the QAOA circuit. Even though the terms $\sum_{k=1}^N X^{(k)}X^{(k+1)}$ and $\sum_{k=1}^N Y^{(k)}Y^{(k+1)}$ do not preserve the magnetization in the Z -basis in general they do so within the sector of the above ansatz.

E. Simulation details

The simulations of the QAOA circuit for the TFIM are done in the free fermion picture yielding a quadratic scaling of the energy evaluation in N . The circuits including L_y layers and for the XXZM do not obey the same symmetries and therefore are implemented as a full circuit simulation using PROJECTQ [58]. The depth of the QAOA circuit for the TFIM is fixed to the smallest value containing the exact ground state $p = N/2$, which gives us N variational parameters with one added per L_y in the second main experiment. For the XXZ model, we choose $p = N$ resulting in $3N$ variational parameters. All circuit simulations are performed exactly, i.e., without noise or sampling. Furthermore we use the SCIPY implementation [59] of the BFGS algorithm and in-house routines for ADAM and NatGrad. All variational parameters are initialized uniformly i.i.d. over the interval $[0.0001, 0.05]$ as this corresponds to initializing the circuit close to the identity and symmetric randomization around 0 has shown slightly worse performance in our experiments.

We bound the BFGS optimization to one period of the rotation parameters as this improves the line search efficiency and found only a small dependence on the position of the interval. For the ADAM optimizer we fixed $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-7}$ and vary η in $[0.005, 0.5]$ trying to build heuristics for the particular problems. We found nontrivial behavior of ADAM with respect to the learning rate, observing a strong influence on the optimization duration, for details see Sec. III A. Furthermore, an increased regularization constant ε did not

yield any improvements of ADAM. For NATGRAD, we use learning rates of 0.5, 0.05, and 0.2 and fix the Tikhonov regularization constant to $\varepsilon_T = 10^{-4}$ and 10^{-3} for the TFIM and XXZM, respectively. This is a choice based on numerical experiments in which we explored the hyperparameter spaces of the optimizers. Even though we did not perform a full study on the impact of ε_T regarding the convergence quality or duration, we gained the following intuitive insight on the regularization: Choosing ε_T to be very small or even deactivating the regularization may lead to very large eigenvalues of F^{-1} , which ultimately are bounded artificially by the method of (pseudo-)inverting F . Consequentially, the Natural Gradient might lead to unreasonably large updates when choosing a fixed moderate learning rate η . We confirmed this numerically and observed the jumps generated by this effect to significantly degrade the optimization quality. Choosing a strong regularization on the other hand reduces the impact of the Fubini-Study metric and the (renormalized) limit $\varepsilon_T \rightarrow \infty$ corresponds to the standard gradient descent in Eq. (7). We therefore chose ε_T such that NatGrad did not perform excessive jumps in our preliminary experiments while maintaining a significant contribution of F to the optimizer.

Employing (block) diagonal approximations to the Fubini-Study matrix as suggested in [19] was not successful due to long-range correlations between the variational parameters in the circuit.

III. MAIN RESULTS

In this section, we state and assess the main numerical results of the paper. For a detailed description of the optimizers and circuit models, see the Methods section above (Sec. II).

A. QAOA circuits for the TFIM

We start our numerical investigation with the QAOA circuit for the TFIM on N qubits with critical transverse field $t = 1$ and analyze the *accuracy, speed and stability* of all three optimizers BFGS, ADAM and NatGrad (see Sec. II A 1 a for the ansatz and Sec. II D 1 for the model). We consider circuits with a depth of $p = N/2$ blocks corresponding to $n = N$ parameters, which are sufficiently expressive to contain the ground state and respect the symmetries of the Hamiltonian. For each system size, we sample 20 points close to the origin in parameter space and initialize each optimizer at these positions (see Sec. II E for simulation details). This leads to statistically distributed performances of the algorithms and as we perform exact simulations without sampling and noise it is the only source of stochasticity. The minimal relative error δ_{\min} and the number of required epochs for each initial point and optimizer are shown in Fig. 2.

Before we analyze the results, recall that the optimization problem can be solved exactly, i.e., the ansatz contains the true ground state. This enables us to identify optimization results with precisions $\delta_{\min} \geq 10^{-3}$ as local minima and we consider them to be unsuccessful as they deviate from the ground state on a physically relevant scale. In practical applications, the precision reached in both local and global minima would be much lower and in particular results with $\delta \approx 10^{-10}$ are

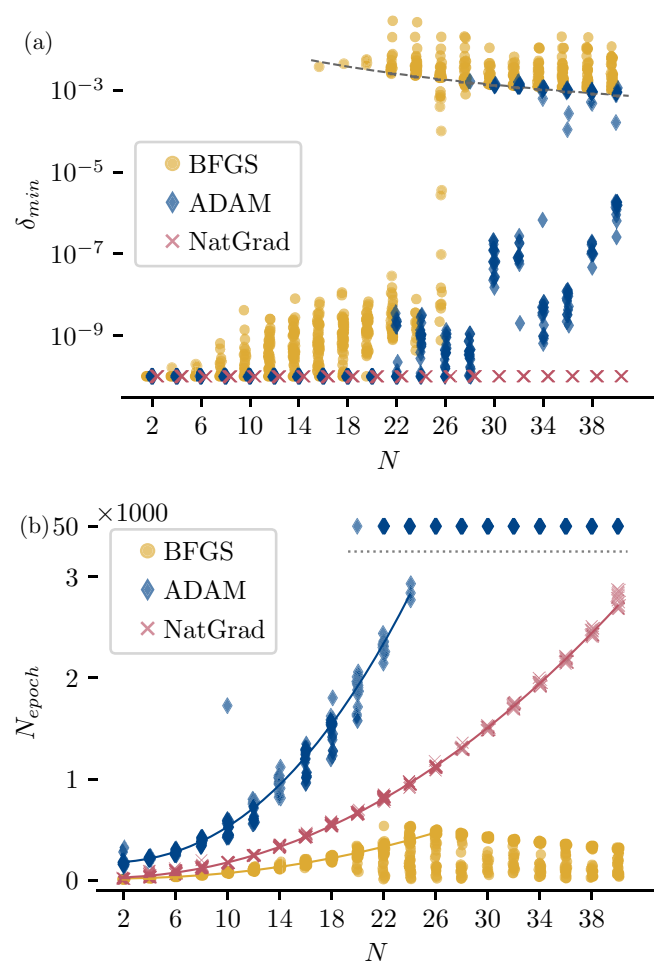


FIG. 2. Relative error δ_{\min} and epoch count N_{epoch} for the three optimizers initialized at 20 randomly chosen points close to the origin for the QAOA circuit with $n = N$ variational parameters. The ADAM optimizer is chosen with a learning rate of $\eta = 0.06$. (a) NatGrad reaches the ground state for all instances and all system sizes, while BFGS and ADAM start systematically getting stuck in local minima close to the first excited state (dashed line) beyond a system size of $N = 20$. (b) The monomial fits to the mean number of epochs to global minimization yield the scalings $N^{2.1}$ (BFGS), $N^{2.3}$ (ADAM), and $N^{2.1}$ (NatGrad). ADAM experiences a transition around $N = 22$ qubits, where the number of epochs to convergence jumps by an order of magnitude (separated by dotted line).

unreasonable to measure in quantum machines. This choice of benchmark is made in order to clearly reveal intrinsic features of the optimizers. For realistic applications, a systematic study of noise needs to be taken into account as well.

Our first observation is that the BFGS optimizer systematically fails to converge for systems sizes larger than $N = 20$. For small system sizes, however, it reaches a global minimum in the smallest number of epochs and at low cost per epoch (see Table I). The fast convergence is preserved for failed runs, which demonstrates that BFGS gets stuck in local minima, and can be attributed to the flexible parameter update size based on the line search subroutine. The runs of BFGS interrupted at a $\delta < 10^{-6}$ level could be improved to reach the

goal of $\delta = 10^{-10}$ by tuning the interrupt criterion. Therefore these runs are considered successful.

For ADAM, we here show the optimization results with $\eta = 0.06$, which similarly display a deterioration in accuracy for system sizes beyond $N = 26$. It is important to note that the failed ADAM runs are interrupted after 5×10^4 epochs and convergence with additional run time is not excluded in general. The question is then: How many update steps are needed for convergence? We observe a polynomial scaling of the required epochs in the system size up to a transition point $N^*(\eta)$, which depends on the chosen learning rate. Above this system size *both* successful and failing runs take much longer and exceed the set budget of 5×10^4 epochs.

The learning rate η imposes two main effects on the run time of the ADAM optimizer: On one hand, the transition point described above marks the system size at which a given learning rate leads to unpredictably high epoch numbers and increasing η shifts this point to smaller system sizes. On the other hand, a reduced learning rate slows down the optimization significantly, prolonging the optimization duration unnecessarily for all $N < N^*(\eta)$. This makes the choice of the learning rate for ADAM a system-dependent fine tuning problem, requiring additional heuristics and hyperparameter optimization. We present a more detailed analysis of the influence of the learning rate on the performance of ADAM in Appendix B.

In Fig. 2, we present the ADAM runs for a medium learning rate in order to demonstrate the described behavior but not the best possible performance of the ADAM optimizer.

NatGrad shows reliable convergence to a global minimum for all sampled initial parameters. The number of epochs to convergence scales polynomially with the system size and there is little variance in the required number of epochs.

For most of the unsuccessful runs, the relative error is very close to the (relative) gap of the Hamiltonian demonstrating that these local minima of the energy landscape correspond to excited states. This has been observed before in the context of digitized quantum annealing and QAOA [4] where the transition from ground to excited state is caused by a small energy gap of the (time-dependent) annealing Hamiltonian. The convergence to a local minimum reproduces this transition and demonstrates that the failed optimization runs yield deviations from the ground state not only on a level of numerical imprecisions but on a physically relevant scale, leading to wrong results of the VQE. For transverse fields other than $t = 1$, the similarity between the gap and the error due to local minima was not confirmed (see Appendix C) and, in particular, the latter is too small for the presented optimizer comparison for $t > 1$ and the optimization becomes too easy for $t < 1$.

Using the scalings as discussed in Sec. IIC and taking the translation symmetry of the TFIM into account, we show the expected optimization durations on a quantum computer in Fig. 3. Due to the increased cost per epoch and a similar scaling of the number of epochs for all optimizers, the cost for NatGrad are considerably higher than those for BFGS and ADAM in the regimes in which they converge and ADAM does not suffer from the sudden increase in required epochs. We expect the scaling for ADAM, which is truncated in Fig. 2 due to our epoch budget, to yield quantum run times

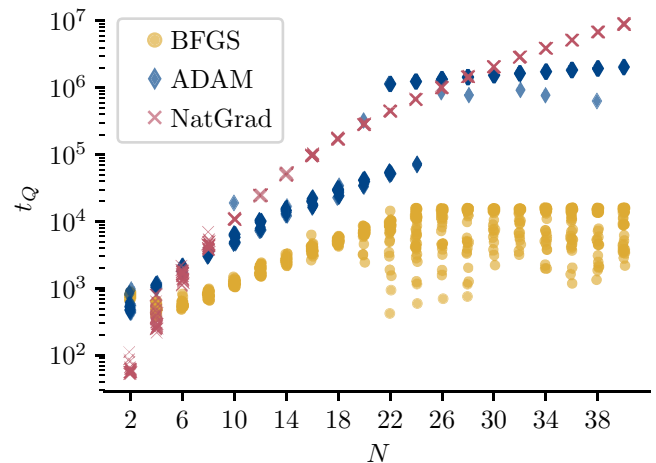


FIG. 3. Estimated run times t_Q on a quantum computer for the optimization runs shown in Fig. 2 based on the scalings in Table I. We note that none of the ADAM optimizations for $N \geq 30$ attained the full precision of 10^{-10} such that the scaling is truncated based on the epoch budget.

comparable to those of NatGrad. As we show in Appendix B, reducing the learning rate makes bigger system sizes accessible to ADAM, but also rather drastically increases run times because of slower convergence.

In summary, we find the BFGS optimizer to run into convergence problems already for medium sized systems, ADAM to take a large number of epochs with a transition into unpredictable cost at a certain system size and NatGrad to exhibit reliable convergence. While the estimated cost for running NatGrad on a real quantum computer are high, the number of epochs is much smaller than for ADAM. This implies that in applications like quantum chemistry which exhibit a more favourable scaling for measuring the Fubini matrix as compared to the gradient, NatGrad can be expected to be significantly cheaper overall (cf. Sec. IIC 1).

Furthermore, the success of both commonly used optimizers, BFGS and ADAM, strongly depends on the initial parameters whereas NatGrad shows stable convergence and a small variance of the optimization duration.

B. Overparametrization by adding Y layers

We now extend the optimal QAOA circuit for the TFIM by adding redundant layers of Pauli Y rotations. These additional rotations can be deactivated by setting their variational parameter κ to zero. This means in particular that the new ansatz classes still contain the ground state and simply introduce a form of overparametrization. Alternatively one can introduce additional degrees of freedom to the circuit by using more blocks in the QAOA circuit than minimally required, maintaining the symmetry of the model, which is shown in Sec. IIIC.

As single-qubit Pauli Y rotations cannot be represented in the free fermion basis of the Hamiltonian [see Eq. (17)], the overparametrized class can be seen as breaking a symmetry. This means that for any given $\kappa \neq 0$, the ansatz state will not be a global minimum and it will be crucial for an optimization

algorithm to find the subspace with $\kappa = 0$. This is clear for a single additional layer of gates, but we expect it to hold for multiple nonadjacent layers as well. Although the present situation is artificially constructed and the broken symmetry is manifest, similar behavior is expected in systems where we do not have an analytical solution. More generally, even for an ansatz class which is suitable to express the ground state a very specific configuration of the variational parameters is necessary to find that state and the chosen optimization algorithm consequentially should be resilient to local minima.

Our choice of overparametrization leads to such local minima, constructing an optimization problem that can be used as a test for the resilience of the optimizer. It furthermore is comparable to overparametrizing a classical neural network respecting translational symmetry but outside of the sector equivalent to free fermions as presented in, e.g., Ref. [42]. The presented experiment thus can be used to compare the optimizer performance for classical machine learning of quantum states and VQES.

We look at two configurations of the extended circuits with y -rotation layers included at positions $\{\lfloor \frac{N}{4} \rfloor\}$ and $\{\lfloor \frac{N}{4} \rfloor, \lfloor \frac{N}{2} \rfloor - 1\}$, respectively. With this choice we avoid special points in the circuit and expect these setups to properly emulate the problem of (additional) local minima.

Again we sample 20 positions in parameter space close to the origin and initialize the three optimizers at these points, resulting in the precisions and success ratios shown in Fig. 4 together with the estimated quantum computer run times in Fig. 5. We observe a clear distinction between the optimizations that succeed to find a global minimum and those which converge to a local minimum only, which makes the success ratio for this numerical experiment well-defined. In contrast to the results for the minimal QAOA circuit, no intermediate precisions caused by a finite epoch budget occur. All optimizers suffer from the introduced gates as they show convergence to local minima for system sizes they tackled successfully without overparametrization. The error of these attained local minima lies on a relevant scale but is smaller than the gap of the model by a factor of ~ 0.4 .

For BFGS, this effect appears for some system sizes for one layer of Pauli Y rotations but is much stronger for two additional layers, reducing the fraction of globally minimized runs to less than 50% for multiple system sizes. We do not claim a scaling behavior with the system size but note an alternating pattern for the configuration with two Y layers, demonstrating large fluctuations of the success ratio (cf. in particular system sizes 10 and 12 for two Y layers).

For the ADAM optimizer, we use a comparably small learning rate of $\eta = 0.02$, which pushes the jump of the optimization duration that we observed before well out of the treated system size range. Nonetheless, we observe runs stuck in local minima already for small systems without exceeding the epoch budget so in contrast to Sec. III A allowing for a longer run time would not improve the performance. Also for ADAM, the fraction of successful instances fluctuates with the system size but in particular for two Pauli Y rotation layers the effect becomes stronger for bigger systems and no successful runs were observed for $N \geq 14$.

The performance of NatGrad on the other hand, for which we reduced the learning rate to $\eta = 0.05$, is more reliable

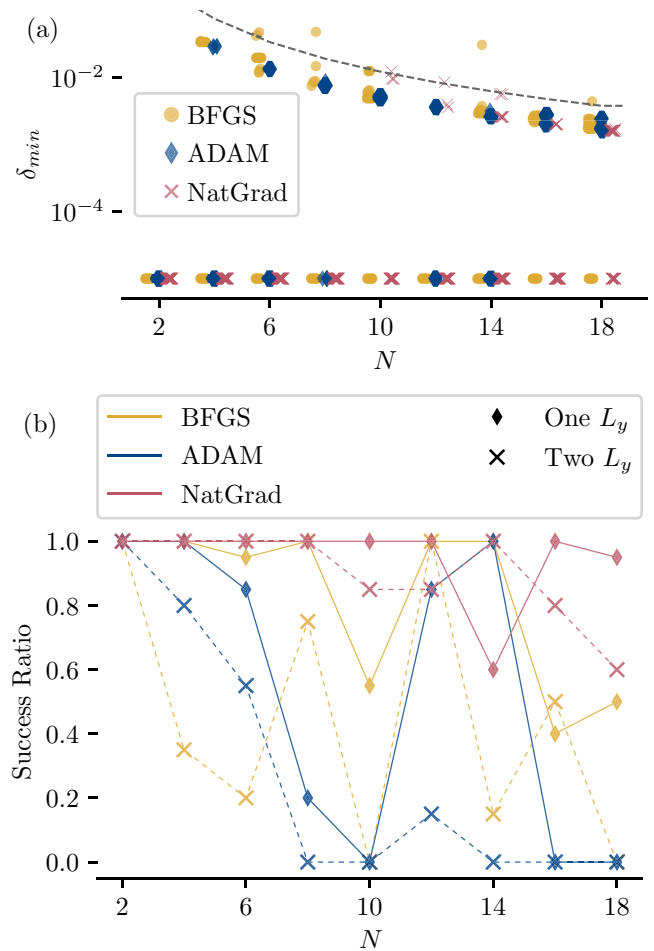


FIG. 4. (a) Achieved precisions δ_{\min} and (b) fraction of successful optimizations out of 20 runs with the three optimizers on QAOA circuits extended by one or two Pauli Y -rotation layers. Successful optimization runs and those only converging locally are separated by a gap in the attained minimal precision, which is smaller but on the scale of the gap of the model, and in contrast to Fig. 2 the epoch budget is almost never consumed entirely. Instead the optimization is completed, yielding either a global or a local minimum.

and the success rate is the best for most of the circuits, with few exceptions. In particular, there are only few system sizes with local convergence for one and two additional degrees of freedom each and overall the success rate of NatGrad does not drop below 60%. For 10 and 18 qubits and two additional layers, NatGrad solves 85% and 60% of the task instances, respectively, while BFGS and ADAM fail in *all* of them.

For all optimizers, we confirm that successful runs deactivate the additional Pauli Y rotation layers by setting the corresponding parameters to 0 and that all optimizations with worse precision failed to do so, leading to a local minimization only. The quantum run times demonstrate the expected scaling with NatGrad as the most expensive optimizer, where the small epoch count compensates the increased cost per epoch for small systems. However, the increased effort is rewarded with significantly higher success rates, making NatGrad a strong choice for (potentially) overparametrized VQE optimization. We want to stress that the relative cost of the Fubini

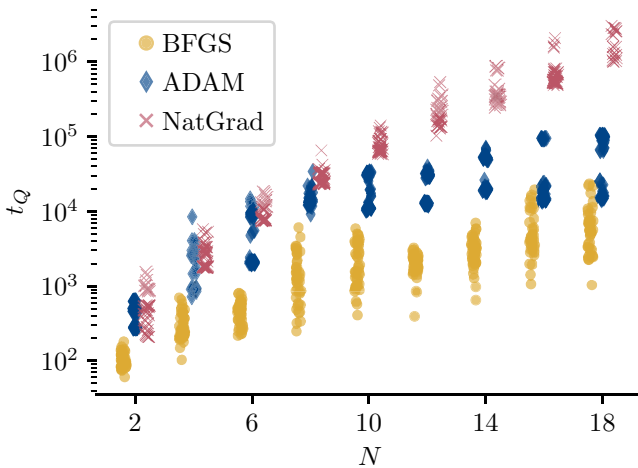


FIG. 5. Estimated run times t_Q on a quantum computer for the optimizations in Fig. 4 based on Table I and the same assumptions as in Fig. 3. For the ADAM optimizer the lower branch of data points corresponds to successful minimizations.

matrix are high for spin chain systems and that the reduced number of epochs required by NatGrad will have a bigger impact in other systems (see also Sec. II C 1).

Overall our numerical experiments with the extended QAOA circuits for the TFIM demonstrate the fragility of the three tested optimizers to perturbations of the ansatz class. A significant decrease in performance is caused by overparametrization outside of the symmetry sector of the model and the QAOA ansatz class. All algorithms were successful for the original QAOA circuits on the considered system sizes implying that the reduced success ratio can directly be attributed to the extension of the ansatz class. This is in contrast to machine learning settings where heavy overparametrization is essential to make the cost function landscape tractable to local optimizers like ADAM. The strong fluctuations over the tested system sizes indicate that more repetitions of the optimization would be required to resolve systematic behavior.

We note that the BFGS algorithm in some instances converges to a local minimum although it has access to nonlocal information via its line search subroutine. In particular, in the presence of two misleading parameters in the search space, the local information determining the one-dimensional subspace does not seem to suffice any longer to find the global minimum, even though the approximated Hessian is used. For the ADAM optimizer, the initial gradient leads to an activation of symmetry breaking layers and due to the restriction to local information the algorithm is not able to leave the resulting sector of the search space with local minima it enters initially. NatGrad also is affected by the limitation to local information but because of the access to geometric properties of the ansatz state class it was on average less likely to leave the Pauli Y -rotation layers activated. We attribute this to the fact that NatGrad performs the optimization in the locally undeformed Hilbert space by extracting the influence of the parametrization. As a consequence the optimizer does not follow the incentive to activate the Pauli Y rotations at the beginning when given the same gradient as ADAM, but stays within the minimal parameter subspace. A better foundation for this

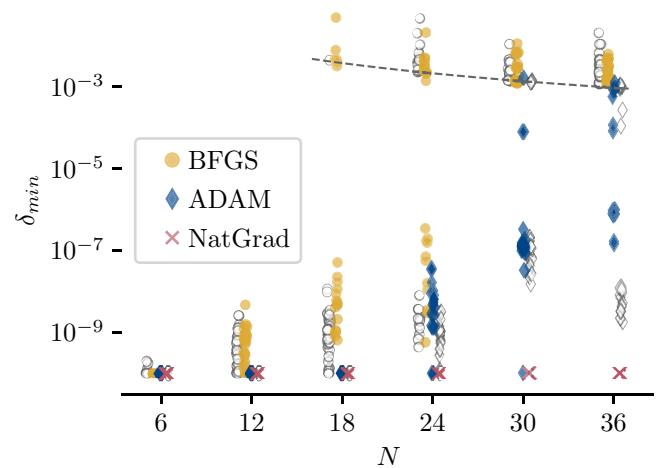


FIG. 6. Minimal attained relative errors δ_{\min} for the TFIM as in Fig. 2 but with the enlarged QAOA circuit containing 2 additional blocks. The empty markers show the results from Fig. 2 for comparison.

intuition and the observed exceptions will be subject to further investigations of NatGrad.

Our results also hint at possible hurdles for adaptive optimization strategies which construct the circuit ansatz iteratively: to obtain viable scaling with the problem size and parameter count, such algorithms have to rate the available gates based on local information in order to estimate their usefulness for the VQE. This rating however might suggest gates which introduce problematic local minima as in the case demonstrated here. When testing ADAPT-VQE [16] for the TFIM we indeed observed that rating gate layers by their gradient suggests using L_y , which—as demonstrated above—is harmful for the VQE.

C. Symmetry-preserving overparametrization

Here we discuss the effect of overparametrizing the QAOA ansatz for the TFIM with symmetry-preserving layers, i.e., by choosing the number of blocks p bigger than the minimum $\lfloor \frac{N}{2} \rfloor$ required to achieve the exact solution. To this end, we optimized the QAOA ansatz on the critical TFIM with two additional blocks, corresponding to four additional variational parameters while keeping all hyper- and simulation parameters fixed and present the attained relative precisions in Fig. 6.

All optimizers perform similarly to the optimizations of the minimal QAOA circuit (displayed with empty markers). The BFGS optimizer achieves slightly less precise results, ADAM obtains similar precisions within statistical fluctuations, showing singular improved convergence but many results with worse precision, and NatGrad solves all instances to requested precision as before. In particular, this means that overparametrization does not facilitate the optimization task but even tends to make it more difficult for the established optimizers. For the BFGS algorithm, this is in accordance with the intuition for large systems which links the poor performance to the high dimensionality of the parameter space and the unfit information access via line searches (see Sec. III A). For ADAM however, the results show a decisive difference

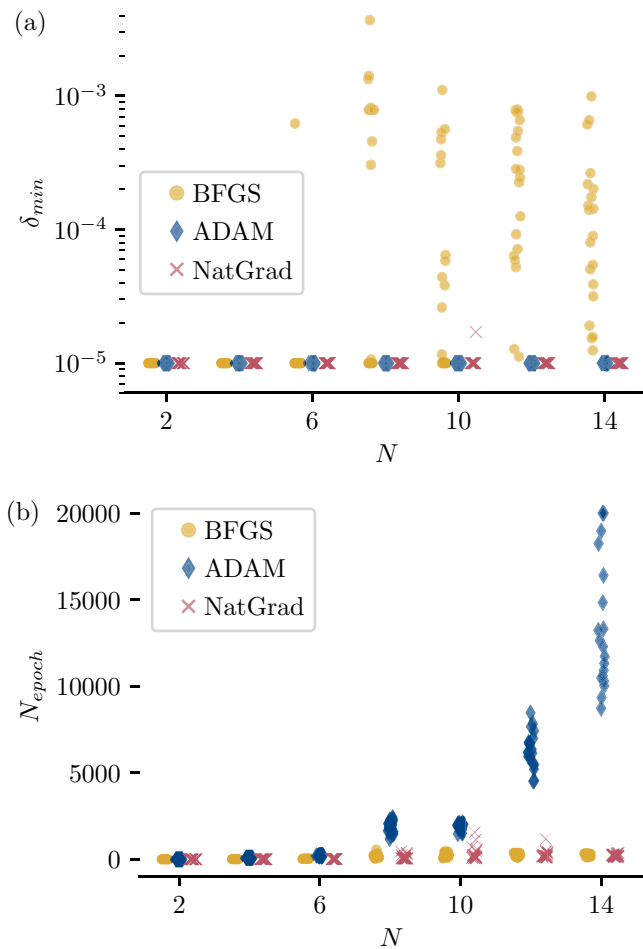


FIG. 7. (a) Minimal achieved precisions and (b) epoch count of the three optimizers and 20 runs on the ansatz in Eq. (23) for the XXZM at depth $p = N$. The circuit contains $n = 3N$ parameters and the learning rates are 0.03 and 0.2 for ADAM and NATGRAD, respectively. The epoch count is displayed on a logarithmic scale for these results.

between the classical machine learning setting and VQE as ADAM thrives on overparametrization in classical cost functions but struggles to exploit additional degrees of freedom in the ansatz circuit. Results of experiments on smaller systems (up to $N = 24$) indicate, that the optimizers behave as described above for stronger overparametrization (up to $n = \frac{4N}{3}$) as well.

D. Results on the Heisenberg model

To complement the study on scaling and overparametrization in the integrable TFIM, we present here numerical results on the XXZM with the ansatz discussed in detail in Sec. IID 2. The performance of the three optimizers, again initialized at 20 distinct points close to 0, is shown in Fig. 7 together with the number of epochs.

The BFGS optimizer shows problems in convergence for increasing circuit sizes but there seems to be a continuous transition between local and global minimum precisions such that a success rate can not be defined as easily. The low

number of epochs to convergence required by BFGS—for both global minima and low-quality results—makes it the cheapest optimizer but the unreliable optimization outcomes underline its infeasibility for large-scale VQES.

The behavior of ADAM is comparable to the one observed on the TFIM when using sufficiently small learning rates (cf. Appendix B): while the target precision of 10^{-5} is reached systematically for all problem sizes, the epoch count exhibits a rapid increase. It does not only appear to be exponential but additionally shows abrupt jumps e.g., when increasing the size from 6 to 8 and from 10 to 12 qubits.

The number of variational parameters at which the loss of precision of BFGS and the increase in epochs for ADAM occur is similar to that in the TFIM: The BFGS optimizer starts failing to reach the target precision at $n = 24$ and $n = 22$ for the XXZM and the TFIM, respectively. Likewise the cost of ADAM in Fig. 7 jumps abruptly at $n = 24$ and $n = 36$ and the runs with comparable learning rate for the TFIM show (less clear) transitions at $n = 26$ and $n = 30$ (see Fig. 9). The Hilbert space dimension however clearly differs at the transition points. While it is intuitively clear that the main influence should be due to the properties of the parameter space, the physical system size in general could affect the performance, too.

The reliable performance of NatGrad was confirmed for the XXZM, failing to converge globally only once for 10 qubits. These high quality results were obtained by modifying the regularization constant ε_T from 10^{-4} to 10^{-3} and setting the learning rate $\eta = 0.2$. This improvement is based on the observation that runs with a smaller learning rate and regularization were interrupted prematurely due to slow learning. We would like to emphasize that the presented choice is not the result of an extensive hyperparameter optimization but the best of a few tested settings, out of which only two were benchmarked on the full set of optimization tasks. The epoch count for the NatGrad optimizer shows more fluctuations than before but is much smaller than for ADAM. For 14 qubits, ADAM takes between 8721 and 20 000 epochs, while the count for NatGrad ranges from 132 to 361.

For a fair comparison of the optimizer cost, we again look at the estimated quantum computing run times t_Q in Fig. 8. Due to the small epoch count and comparably low cost per epoch, the unsuccessful BFGS runs clearly are cheapest. More interestingly, the difference in the number of epochs between NatGrad and ADAM discussed above equalizes the overhead in the cost per epoch of NatGrad due to the Fubini-Study matrix computation. This trend was indicated in the minimal QAOA circuit results for the TFIM (cf. Fig. 3) but distorted by the finite epoch budget.

The results for the Heisenberg model overall confirm the observations on the TFIM: NatGrad exhibits a favourable scaling in the epoch count which remedies the increased effort per epoch that is required to determine the Fubini matrix as compared to ADAM. Meanwhile, ADAM shows unpredictable behavior in its optimization cost but consistently attains the target precision whereas BFGS suffers from high dimensional search spaces, rendering it a cheap but unreliable method for VQES. We emphasize that the relative cost for measuring the Fubini-Study matrix in NatGrad is smaller for Hamiltonians with many terms as discussed in Sec. IIC 1. This means that

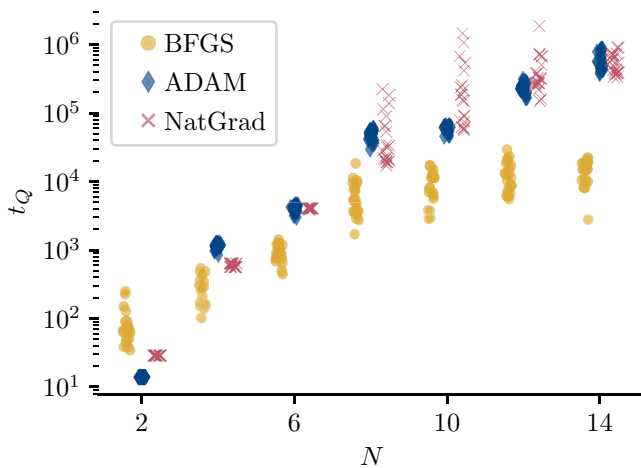


FIG. 8. Estimated quantum run times t_Q for the optimization tasks in Fig. 7 based on Table I and the same assumptions as in Fig. 3 and 5.

the quantum run times for NatGrad can be significantly better for such systems and the relative scaling of t_Q is comparable to the drastic distinction seen for N_{epoch} in Fig. 7 as the cost per epoch approach the ones for any gradient-based method.

IV. CONCLUSION

Our first main result shows that the BFGS optimizer, while quick and reliably for small systems, has an increased chance getting stuck in local minima already in medium sized VQEs that are comparable to present day and near future NISQ devices. This may be surprising as it has access to nonlocal information due to its line search subroutine. We suspect that this aspect of the algorithm becomes less helpful for finding a global minimum because of its sparsity in high-dimensional parameter spaces.

The ADAM optimizer on the other hand is able to find global minima also in larger parameter spaces (up to 42) for suitably small learning rates but this comes at the cost of a quickly increasing number of epochs to complete the optimization. In particular we observed two effects of the learning rate η on the run time of ADAM: On the one hand, there is a threshold size of the parameter space that depends on η above which the epoch count rapidly increases, which means that a small enough value of the learning rate is essential to avoid extremely long run times. On the other hand, the optimization duration for sizes below the threshold is significantly increased when reducing η making it undesirable to choose the learning rate smaller than strictly necessary. It thus appears that tedious hyperparameter tuning is necessary to balance these two effects.

The NatGrad optimizer recently proposed for VQE shows very reliable convergence to a global minimum for all tested system sizes within fewer epochs but at high cost per epoch. The problem of jumping into barren plateaus even after a suitable initialization can be fixed via Tikhonov regularization, which can be tuned with a continuous parameter to gradually trade the benefit from the information geometry for stability. This makes the algorithm a promising, although more ex-

pensive, candidate for the optimization of future VQES. The increased cost for determining the Fubini matrix at each step have a particularly strong effect on the estimated quantum run time for spin chain systems, for other systems with more favourable scaling NatGrad might not only be more reliable but additionally exhibit lower cost.

Our second main experiment treats overparametrization in VQE ansatz classes including an example of additional rotation gates that break the symmetry of the Hamiltonian as well as symmetry-preserving overparametrization. The BFGS optimizer fails to find a global minimum in some instances even for very small systems and in general exhibits a strongly fluctuating performance which decreases considerably with the number of additional gate layers.

Also ADAM showed strong susceptibility to the additional degrees of freedom. Beyond the implications on applications, this is interesting because overparametrization is heavily used in machine learning to make the cost function tractable for optimizers like ADAM and we therefore appear to observe a fundamental difference between classical machine learning and VQES.

Finally, NatGrad showed some failed optimization runs for selected system sizes as well but mostly remained successful even for multiple additional gate layers. It therefore rewards its increased cost per epoch with higher success rates and is the only tested optimization strategy that showed resilience to both big search spaces and local minima caused by overparametrization.

We therefore conclude that overparametrization which extends the effective Hilbert space is a serious problem for standard optimizers and even NatGrad as most resilient algorithm is disturbed by this issue. The simulation cost restricted the maximal system size for this second experiment but there is no reason to assume that a stronger overparametrization with more symmetry breaking layers would resolve these problems. This implies difficulties for adaptive ansatz techniques because standard rating strategies cannot detect this property and the gate set therefore has to be minimal in order to prevent this type of overparametrization.

For overparametrized ansatz classes *within* the symmetry sector of the TFIM, all optimizers behave similar to the minimal parametrization or show slightly worse convergence. This demonstrates that the optimization problem within VQES differs significantly from optimizations in classical machine learning, where overparametrization enhances the performance of ADAM.

In general, one could expect the cost function of VQES to behave differently than those in common machine learning models as the parameters enter in a very nonlinear manner via rotation gates. The restriction of NISQ devices to rather shallow circuits implies much smaller numbers of variational parameters than in machine learning and therefore NatGrad can be considered a viable option for VQE optimization while using second order resources.

The extension of our analysis to the XXZM confirmed the problems of the BFGS optimizer with big search spaces and the rapid run time growth for ADAM. NatGrad performed reliably on the XXZM as well and the reduced number of epochs compensates the cost per epoch such that the cost of the convergent optimizers ADAM and NatGrad are similar for

the tested system sizes. Additional experiments are in order to show further generalization to nonintegrable models, which would imply that a full VQE optimization on big systems in general is most affordable using NatGrad.

Our investigations have shown that NatGrad might enable VQES to solve more complex and bigger problems as it performs well on a test model with challenges representative of those in potential future applications of VQES. If reliability is more important than minimizing the quantum run time of a single optimization run we recommend NatGrad as optimizer of choice. Alternatively, whenever the Hamiltonian of interest contains many terms and thus is expensive to measure, the relative additional cost of obtaining the Fubini matrix become small (see Sec. II C 1) and the high reliability and low number of required epochs of NatGrad again make it the best method.

The observed differences between classical machine learning and VQES show that insights and heuristics from the former do not necessarily apply in the latter case and demonstrate the importance of understanding the optimization problem in VQES and the properties of the optimization algorithms.

ACKNOWLEDGMENTS

We would like to thank Chae-Yeun Park, David Gross, Gian-Luca Anselmetti, and Thorben Frank for helpful discussions. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy-Cluster of Excellence Matter and Light for Quantum Computing (ML4Q) EXC 2004/1-390534769. The authors would like to thank Covestro Deutschland AG, Kaiser Wilhelm Allee 60, 51373 Leverkusen, for the support with computational resources. The work was conducted while all three authors were affiliated with the Institute for Theoretical Physics of the University of Cologne.

APPENDIX A: EXACT SOLUTION OF THE TFIM

Here we derive the analytic solution of the TFIM by mapping it to noninteracting fermions, also see Ref. [15]. We start with the linear combinations $a_k := \frac{1}{2}(Z^{(k)} + iY^{(k)})$, which fulfill

$$X^{(k)} = 2a_k^\dagger a_k - 1, \quad Z^{(k)} = a_k^\dagger + a_k \quad (\text{A1})$$

and map them to the operators

$$b_k := \prod_{l=1}^{k-1} \mathcal{N}_l a_k, \quad \mathcal{N}_l := \exp[i\pi a_l^\dagger a_l], \quad (\text{A2})$$

which satisfy fermionic anticommutation relations:

$$\{b_k^\dagger, b_l\} = \delta_{kl}, \quad \{b_k, b_l\} = \{b_k^\dagger, b_l^\dagger\} = 0. \quad (\text{A3})$$

For the transformation of the Hamiltonians H_S and H_B , which comprise both the TFIM Hamiltonian and the generators for the unitaries in the QAOA ansatz, note that

$$\mathcal{N}_l^2 = \mathbb{1}, \quad \mathcal{N}_l^\dagger = \mathcal{N}_l = \mathcal{N}_l^{-1}, \quad (\text{A4})$$

$$\mathcal{N}_k b_k = b_k, \quad \mathcal{N}_k b_k^\dagger = -b_k^\dagger. \quad (\text{A5})$$

Using Eq. (A1) and the above properties the transformed Hamiltonians read

$$H_S = - \left[\sum_{k=1}^{N-1} (b_k^\dagger - b_k) b_{k+1}^\dagger - (b_N^\dagger - b_N) b_1^\dagger \mathcal{G} \right] + \text{H.c.}, \quad (\text{A6})$$

$$H_B = -t \sum_{k=1}^N 2b_k^\dagger b_k - 1, \quad (\text{A7})$$

where we denote by $\mathcal{G} := \prod_{l=1}^N \mathcal{N}_l$ the gauge factor in the term generated by the periodic boundary conditions and the nonlocal transformation (A3), which also has a reversed sign. \mathcal{G} interacts with the initial state of the QAOA ansatz $|\bar{\psi}\rangle$ and the Hamiltonian terms in the following way:

$$\mathcal{G}|\bar{\psi}\rangle = \exp\left[\frac{i\pi}{2}\left(-\frac{1}{t}H_B + N\right)\right]|+\rangle^{\otimes N} = e^{i\pi N}|\bar{\psi}\rangle, \quad (\text{A8})$$

$$[\mathcal{G}, H_B] = 0 = [\mathcal{G}, H_S], \quad (\text{A9})$$

where we used the ground state energy $-tN$ of H_B and Eq. (A5). This means that the reversed sign is canceled for odd N . Therefore we introduce an additional phase via the transformation

$$c_k := e^{ikv} b_k, \quad v := \begin{cases} \pi/N & \text{for } N \text{ even} \\ 0 & \text{for } N \text{ odd} \end{cases}, \quad (\text{A10})$$

$$H_S = - \left[\sum_{k=1}^N e^{iv} (c_k^\dagger e^{i2kv} - c_k) c_{k+1}^\dagger \right] + \text{H.c.}, \quad (\text{A11})$$

$$H_B = -t \sum_{k=1}^N 2c_k^\dagger c_k - 1, \quad (\text{A12})$$

where we defined v such that the result holds for both odd and even N . The last mapping we perform is a Fourier transformation with shifted momenta:

$$d_q := \frac{1}{\sqrt{N}} \sum_{k=1}^N e^{2\pi i(q-1)k/N} c_k, \quad (\text{A13})$$

$$H_S = - \left[\sum_{q=1}^N e^{-i\alpha_q} d_q^\dagger d_{-q}^\dagger - e^{i\alpha_q} d_q d_q^\dagger \right] + \text{H.c.}, \quad (\text{A14})$$

$$H_B = t \sum_{q=1}^N 2d_q^\dagger d_q - 1 \quad (\text{A15})$$

with mode-dependent angles and relabeled Fourier modes

$$\alpha_q := \begin{cases} (2q-1)\pi/N & \text{for } N \text{ even} \\ 2q\pi/N & \text{for } N \text{ odd} \end{cases}, \quad (\text{A16})$$

$$d_{-q} := \begin{cases} d_{N+1-q} & \text{for } N \text{ even} \\ d_{N+2-q} & \text{for } N \text{ odd} \end{cases}. \quad (\text{A17})$$

We finally can split up the sums, recollect the terms corresponding to the pairs $\{d_q, d_{-q}\}$ and rewrite the Hamiltonians

in a fermionic operator basis:

$$H_S = H'_S - 2 \left[\sum_{q=1}^r \cos \alpha_q (d_q^\dagger d_q - d_{-q} d_{-q}^\dagger) - i \sin \alpha_q (d_q^\dagger d_{-q}^\dagger - d_{-q} d_q) \right] \quad (\text{A18})$$

$$= -2 \sum_{q=1}^r \begin{pmatrix} d_q^\dagger & d_{-q} \\ i \sin \alpha_q & -\cos \alpha_q \end{pmatrix} \begin{pmatrix} d_q \\ d_{-q}^\dagger \end{pmatrix} + H'_S, \quad (\text{A19})$$

$$H_B = H'_B - 2t \sum_{q=1}^r d_q^\dagger d_q - d_{-q} d_{-q}^\dagger \quad (\text{A20})$$

$$= H'_B - 2t \sum_{q=1}^r \begin{pmatrix} d_q^\dagger & d_{-q} \\ 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} d_q \\ d_{-q}^\dagger \end{pmatrix}, \quad (\text{A21})$$

where $H'_B = H'_S = 0$ and $H_B/t = H'_S = -1$ for even and odd N , respectively, using $d_1^\dagger d_1 |\bar{\psi}\rangle = 1$ and $d_1 d_1^\dagger |\bar{\psi}\rangle = 0$ for the odd case.

In this shape, the simple structure of the model becomes apparent as we identify $r = \lfloor \frac{N}{2} \rfloor$ pairs of fermionic modes in momentum space which interact within but not between the pairs. The Hamiltonian can thus be written as a direct sum

$$H_{\text{TfI}} = -2 \bigoplus_{q=1}^r (t + \cos \alpha_q) Z + \sin \alpha_q Y - (1+t)(N-2r). \quad (\text{A22})$$

Due to the fact that H_B and H_S not only constitute H_{TfI} but also generate the (modified) QAOA ansatz, the simulation of the circuit can be carried out on a $2r$ -dimensional space that decomposes into the direct sum above. On the Bloch spheres of the free fermions, the two time evolution operators $e^{-i\theta H_S}$ and $e^{-i\theta H_B}$ correspond to rotations about the individual axes $e_q = (0, \sin \alpha_q, \cos \alpha_q)$ and the z axis, respectively. Furthermore we can manually solve for the ground state of the TFIM by computing the ground state in each subspace individually:

$$E_0 = E' - 2 \sum_{q=1}^r E_q, \quad |\psi_0\rangle = \bigoplus_{q=1}^r |\psi_{q,0}\rangle, \quad (\text{A23})$$

$$E_q = \sqrt{1 + t^2 + 2t \cos \alpha_q}, \quad (\text{A24})$$

$$|\psi_{q,0}\rangle = \frac{1}{\sqrt{2E_q(E_q - \cos \alpha_q - t)}} \begin{pmatrix} i \sin \alpha_q \\ E_q - \cos \alpha_q - t \end{pmatrix} \quad (\text{A25})$$

where E' is the eigenvalue of $H'_B + H'_S$.

APPENDIX B: LEARNING RATE INFLUENCE ON PERFORMANCE OF ADAM

In order to evaluate the systematically large optimization durations of the ADAM optimizer for the QAOA circuit of the TFIM, we tested it at multiple learning rates from the interval $[0.005, 0.1]$ observing a major influence on the run time, see Fig. 9. For a given learning rate η , the required number of epochs grows polynomially with the system size up to a size N^* above which ADAM takes much longer, exceeding

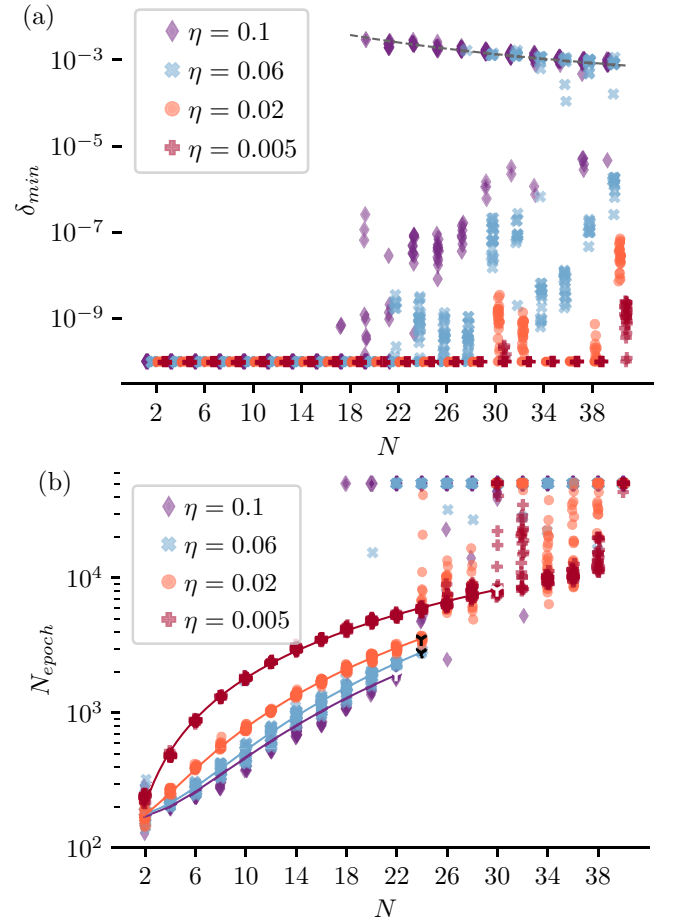


FIG. 9. Minimal attained relative errors δ_{\min} and epoch count N_{epoch} for the ADAM optimizer initialized at 20 distinct points close to zero and with different learning rates η . (a) The threshold size beyond which ADAM fails can be shifted by reducing η , delaying local convergence and output of an excited state (dashed line) to bigger systems. (b) The shown fits are based on *filtered* data in order to determine the apparent scaling for small system sizes and thus do *not* aim at describing the entire data. The biggest system size partially included in the fit is marked. For the shown learning rates in descending order, we obtain the exponents 2.3, 2.3, 1.9, and 1.4 but prefactors 1.8, 1.9, 7.3, and 74.7.

the budget of 5×10^4 epochs. In this second phase, we find the optimizer to require excessively many epochs both when succeeding and when getting stuck in a local minimum (see, e.g., $\eta = 0.06$), which prevents us from systematically distinguishing the two cases before convergence. The observed transition point $N^*(\eta)$ can be shifted towards bigger system sizes by decreasing the learning rate, i.e., $N^*(\eta)$ is monotonically decreasing. Meanwhile, reducing η increases the epoch count significantly for smaller system sizes without disrupting the convergence as is expected for well-behaved systems. Even though the scaling exponent is smaller for lower learning rates the optimization requires more epochs which is due to a large prefactor, increasing the cost for all system sizes before the jump. The observed dependencies of the run time on η result in a system size dependent optimal learning rate which trades off the systematically increased epoch counts for small η against the position of the jump in optimization duration.

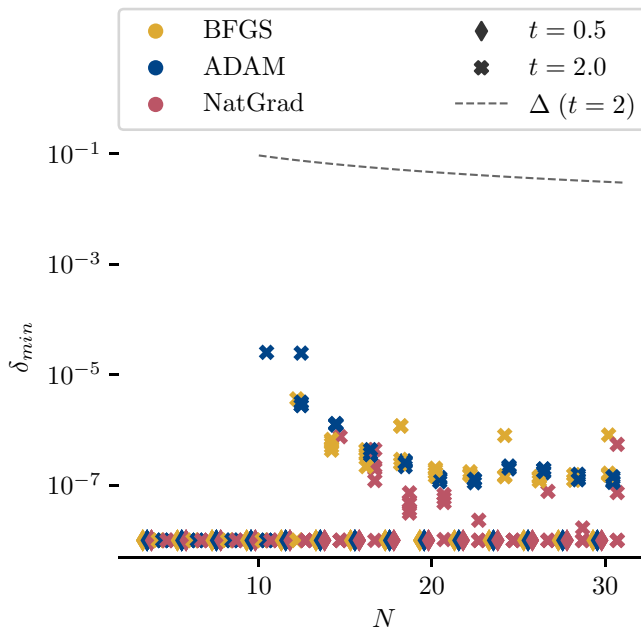


FIG. 10. Minimal attained relative errors δ_{\min} for the TFIM at sub- ($t = 0.5$) and supercritical ($t = 2$) transverse fields and energy gap Δ of the supercritical model.

This demonstrates that heuristics for ADAM are needed in order to achieve systematic global optimization and that the required number of optimization steps can be unpredictably large depending on the hyperparameters.

APPENDIX C: NONCRITICAL TFIM

In this section, we present numerical results for the optimization of the QAOA ansatz for the noncritical TFIM and demonstrate why the critical transverse field strength was chosen for the main investigations. As these experiments are performed for exploratory purposes, the maximal system size is reduced to 30, we choose one field strength for each phase and we sample 5 (instead of 20) initial parameter positions. As shown in Fig. 10, all optimizers succeed in finding global minima to the required precision for the subcritical transverse field strength but the supercritical model is harder to solve than both the sub- and the critical model. Convergence to local minima is observed at $t = 2$ for systems as small as 10 spins. However, we found that the error caused by convergence to local minima is three to five orders of magnitude smaller than the gap of the model, whereas optimizations for the critical model show errors very close to the gap (see Fig. 2). This improved separation of successful and failed optimization runs in the critical model makes it more suitable for the optimizer comparison.

- [1] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, [arXiv:1411.4028](#).
- [2] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. Sawaya *et al.*, Quantum chemistry in the age of quantum computing, *Chem. Rev.* **119**, 10856 (2019).
- [3] A. Smith, M. S. Kim, F. Pollmann, and J. Knolle, Simulating quantum many-body dynamics on a current digital quantum computer, *npj Quantum Inf.* **5**, 106 (2019).
- [4] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices, *Phys. Rev. X* **10**, 021067 (2020).
- [5] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, [arXiv:1412.6980](#).
- [6] A. Karpathy, A peek at trends in machine learning. Blog post, April (2017).
- [7] S. Ruder, An overview of gradient descent optimization algorithms, [arXiv:1609.04747](#).
- [8] M. Ostaszewski, E. Grant, and M. Benedetti, Quantum circuit structure learning, [arXiv:1905.09692](#).
- [9] C. G. Broyden, The convergence of a class of double-rank minimization algorithms 1. General considerations, *IMA J. Appl. Math.* **6**, 76 (1970).
- [10] R. Fletcher, A new approach to variable metric algorithms, *Comput. J.* **13**, 317 (1970).
- [11] D. Goldfarb, A family of variable-metric methods derived by variational means, *Math. Comput.* **24**, 23 (1970).
- [12] D. F. Shanno, Conditioning of quasi-Newton methods for function minimization, *Math. Comput.* **24**, 647 (1970).
- [13] G. Giacomo Guerreschi and M. Smelyanskiy, Practical optimization for hybrid quantum-classical algorithms, [arXiv:1701.01450](#).
- [14] G. B. Mbeng, R. Fazio, and G. Santoro, Quantum annealing: A journey through digitalization, control, and hybrid quantum variational schemes, [arXiv:1906.08948v3](#).
- [15] Z. Wang, N. C. Rubin, J. M. Dominy, and E. G. Rieffel, XY mixers: Analytical and numerical results for the quantum alternating operator Ansatz, *Phys. Rev. A* **101**, 012320 (2020).
- [16] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall, An adaptive variational algorithm for exact molecular simulations on a quantum computer, *Nat. Commun.* **10**, 3007 (2019).
- [17] J. Romero, R. Babbush, J. R. McClean, C. Hempel, P. J. Love, and A. Aspuru-Guzik, Strategies for quantum computing molecular energies using the unitary coupled cluster Ansatz, *Quantum Sci. Tech.* **4**, 014008 (2018).
- [18] B. T. Gard, L. Zhu, G. S. Barron, N. J. Mayhall, S. E. Economou, and E. Barnes, Efficient symmetry-preserving state preparation circuits for the variational quantum eigensolver algorithm, *npj Quantum Inf.* **6**, 10 (2020).
- [19] J. Stokes, J. Izaac, N. Killoran, and G. Carleo, Quantum natural gradient, *Quantum* **4**, 269 (2020).
- [20] S.-I. Amari, Natural gradient works efficiently in learning, *Neural Comput.* **10**, 251 (1998).
- [21] A. Harrow and J. Napp, Low-depth gradient measurements can improve convergence in variational hybrid quantum-classical algorithms, [arXiv:1901.05374](#).
- [22] J. Martens and I. Sutskever, Training deep and recurrent networks with hessian-free optimization, in *Neural Networks: Tricks of the Trade* (Springer, Berlin, Heidelberg, 2012), pp. 479–535.

- [23] R. Livni, S. Shalev-Shwartz, and O. Shamir, On the computational efficiency of training neural networks, in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., New York, NY, 2014), pp. 855–863.
- [24] D. Li, T. Ding, and R. Sun, Over-parameterized deep neural networks have no strict local minima for any continuous activations, [arXiv:1812.11039](https://arxiv.org/abs/1812.11039).
- [25] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, *Nat. Commun.* **9**, 4812 (2018).
- [26] C.-Y. Park and M. J. Kastoryano, Geometry of learning neural quantum states, *Phys. Rev. Res.* **2**, 023232 (2020).
- [27] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Evaluating analytic gradients on quantum hardware, *Phys. Rev. A* **99**, 032331 (2019).
- [28] H. Lun Tang, E. Barnes, H. R. Grimsley, N. J. Mayhall, and S. E. Economou, qubit-ADAPT-VQE: An adaptive algorithm for constructing hardware-efficient Ansatzes on a quantum processor, [arXiv:1911.10205](https://arxiv.org/abs/1911.10205).
- [29] A. G. Rattew, S. Hu, M. Pistoia, R. Chen, and S. Wood, A domain-agnostic, noise-resistant, hardware-efficient evolutionary variational quantum eigensolver, [arXiv:1910.09694](https://arxiv.org/abs/1910.09694).
- [30] M. E. S. Morales, J. Biamonte, and Z. Zimborás, On the universality of the quantum approximate optimization algorithm, *Quantum Info. Process.* **19**, 291 (2020).
- [31] S. Lloyd, Quantum approximate optimization is computationally universal, [arXiv:1812.11075](https://arxiv.org/abs/1812.11075).
- [32] M. B. Hastings, Classical and quantum bounded depth approximation algorithms, [arXiv:1905.07047](https://arxiv.org/abs/1905.07047).
- [33] E. Farhi and A. W. Harrow, Quantum Supremacy through the Quantum Approximate Optimization Algorithm, [arXiv:1602.07674](https://arxiv.org/abs/1602.07674).
- [34] Z. Wang, S. Hadfield, Z. Jiang, and E. G. Rieffel, Quantum approximation optimization algorithm for MaxCut: A fermionic view, *Phys. Rev. A* **97**, 022304 (2018).
- [35] W. W. Ho and T. H. Hsieh, Efficient variational simulation of nontrivial quantum states, *SciPost Physics* **6**, 029 (2019).
- [36] M. Y. Niu, S. Lu, and I. Chuang, Optimizing QAOA: Success probability and runtime dependence on circuit depth, [arXiv:1905.12134](https://arxiv.org/abs/1905.12134).
- [37] V. Akshay, H. Philathong, M. E. Morales, and J. D. Biamonte, Reachability Deficits in Quantum Approximate Optimization, *Phys. Rev. Lett.* **124**, 090504 (2020).
- [38] S. Hadfield, Z. Wang, B. O’Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, From the quantum approximate optimization algorithm to a quantum alternating operator Ansatz, *Algorithms* **12**, 34 (2019).
- [39] J. Duchi, E. Hazan, and Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, *J. Machine Learning Res.* **12**, 61 (2011).
- [40] G. Hinton, N. Srivastava, and K. Swersky, Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning (2012).
- [41] F. Becca and S. Sorella, *Quantum Monte Carlo Approaches for Correlated Systems* (Cambridge University Press, Cambridge, 2017).
- [42] G. Carleo and M. Troyer, Solving the quantum many-body problem with artificial neural networks, *Science* **355**, 602 (2017).
- [43] Y. Li and S. C. Benjamin, Efficient Variational Quantum Simulator Incorporating Active Error Minimization, *Phys. Rev. X* **7**, 021050 (2017).
- [44] P.-L. Dallaire-Demers, J. Romero, L. Veis, S. Sim, and A. Aspuru-Guzik, Low-depth circuit Ansatz for preparing correlated fermionic states on a quantum computer, *Quantum Science Technology* **4**, 045005 (2019).
- [45] S. McArdle, T. Jones, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan, Variational Ansatz-based quantum simulation of imaginary time evolution, *npj Quantum Inf.* **5**, 75 (2019).
- [46] B. Koczor and S. C. Benjamin, Quantum natural gradient generalised to nonunitary circuits, [arXiv:1912.08660](https://arxiv.org/abs/1912.08660).
- [47] N. Yamamoto, On the natural gradient for variational quantum eigensolver, 2019, [arXiv:1909.05074](https://arxiv.org/abs/1909.05074).
- [48] K. M. Nakanishi, K. Fujii, and S. Todo, Sequential minimal optimization for quantum-classical hybrid algorithms, *Phys. Rev. Res.* **2**, 043158 (2020).
- [49] J. Li, X. Yang, X. Peng, and C.-P. Sun, Hybrid Quantum-Classical Approach to Quantum Optimal Control, *Phys. Rev. Lett.* **118**, 150503 (2017).
- [50] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, *Phys. Rev. A* **98**, 032309 (2018).
- [51] K. Mitarai and K. Fujii, Methodology for replacing indirect measurements with direct measurements, *Phys. Rev. Res.* **1**, 013006 (2019).
- [52] J. C. Spall, Multivariate stochastic approximation using a simultaneous perturbation gradient approximation, *IEEE Trans. Autom. Control* **37**, 332 (1992).
- [53] P. Gokhale and F. T. Chong, $O(N^3)$ measurement cost for variational quantum eigensolver on molecular hamiltonians, [arXiv:1908.11857](https://arxiv.org/abs/1908.11857).
- [54] T.-C. Yen, V. Verteletskyi, and A. F. Izmaylov, Measuring all compatible operators in one series of single-qubit measurements using unitary transformations, *J. Chem. Theory Comput.* **16**, 2400 (2020).
- [55] E. Lieb, T. Schultz, and D. Mattis, Two soluble models of an antiferromagnetic chain, *Ann. Phys.* **16**, 407 (1961).
- [56] M. Karbach, G. Müller, H. Gould, and J. Tobochnik, Introduction to the bethe Ansatz i, *Comput. Phys.* **11**, 36 (1997).
- [57] M. Karbach, K. Hu, and G. Müller, Introduction to the bethe Ansatz ii, *Comput. Phys.* **12**, 565 (1998).
- [58] D. S. Steiger, T. Häner, and M. Troyer, Projectq: an open source software framework for quantum computing, *Quantum* **2**, 49 (2018).
- [59] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright *et al.*, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, *Nat. Methods* **17**, 261 (2020).
- [60] B. van Straaten and B. Koczor, Measurement cost of metric-aware variational quantum algorithms, [arXiv:2005.05172](https://arxiv.org/abs/2005.05172).

Correction: The acronym QAOA was mistakenly set during production as QAQA in the abstract and in the heading for Sec. II A 1 (a) and has been fixed.

Conclusion

In this chapter, I present a succinct summary of the main results. The interested reader may find more extensive summaries in the publications in Chap. 2 and 3. The first part of this thesis is concerned with the differentiation of functions that stem from the expectation value of some observable with respect to the quantum state prepared by a parametrized quantum circuit (PQC). Significant effort in the research community has been put into deriving parameter-shift rules and understanding the underlying structure of PQC-based functions, highlighting their trigonometric nature. Although this is a development within the last four years, this perspective has already been used to approach other relevant aspects of PQCs, such as their expressivity, classical surrogate models, tailored optimization methods or error mitigation. It is to be expected that the analysis of PQCs using Fourier analysis will continue to deliver insights into their computational power in practice and lead to heuristics for both circuit architectures and optimization procedures for variational quantum algorithms (VQAs).

Regarding the differentiation of PQCs itself, parameter-shift rules for many relevant classes of gates are covered in the literature by now, including existence and optimality proofs enabled by a proper mathematical foundation. As discussed in Sec. 2.1, there have been numerous works on shift rules with partially overlapping contributions, which makes it difficult to separate the included publication in Chap. 2 from the literature. Concretely, my work contributes to the optimal differentiation of a wide range of PQCs, which improves existing approaches, and the extension to non-equidistant Fourier spectra. Additionally, I analysed the cost of the generalized shift rules thoroughly for both quantum processing units (QPUs) and classical simulators. A specific result is that QAOA benefits from this gradient estimator and that bounds obtained with classical methods can be used to reduce the required resources in the quantum computation. The perspective on higher-order derivatives, concretely the Hessian and the metric tensor, of PQCs with gates other than Pauli rotations is a new contribution as well. It is a subject of current research to compare unbiased gradient estimators, like the parameter-shift rule, to established biased estimators and to more involved methods that exploit e.g. knowledge from past function and gradient evaluations. In connection to the latter, shot-frugal heuristics for VQAs that take the full optimization procedure into account have been investigated as well.

Considering VQAs in end-to-end analyses leads to strong dependencies on the considered benchmark problems and on many choices in the algorithm design, making it difficult

to arrive at statements that hold in general. Therefore it is an important open question to find metrics for the quality of gradient estimators – and of other subroutines – that hint at their usefulness in the algorithms and can be used as surrogates for benchmark simulations. This would allow for separating the choice of the estimator from the optimization and for evaluating the former without considering the complete VQA workflow. A rather generic metric, the (elementwise) mean squared error (MSE) of the gradient, is widely used to date², even though it does not cover some estimators that are commonly used in practice such as simultaneous perturbation stochastic approximation (SPSA). Moreover, it is not clear that minimizing the elementwise MSE actually leads to the most useful gradient estimator. Solving these shortcomings is a promising subject for future work.

The second part of this thesis is concerned with VQAs in a broader scope and the second publication addresses the choice of optimization algorithm in the variational quantum eigensolver (VQE) for spin chain problems. In this work I conducted several numerical experiments for a large range of hyperparameters and problem sizes in order to investigate the performance of two optimizers established in classical computing and of the quantum natural gradient optimizer (QNG). This offered early insights into the strengths of QNG and its behaviour in full VQA runs. Another key aspect of this publication is the consideration of symmetry-breaking gates in otherwise symmetry-preserving PQCs and of their impact on the optimization performance. Finally, I observed some relevant differences in optimization behaviour with respect to overparametrization in PQCs as compared to classical machine learning applications. This becomes especially clear for the Adam optimizer, which is widely used for the training of e.g. deep neural networks. As mentioned above, experiments like those presented in this publication require us to make a series of choices for the particular benchmark simulations. This limits the scope of the obtained insights and makes it difficult to extract reproducible statements that hold more generally and for a wide range of the many involved (hyper)parameters.

A review of the literature introducing and discussing new optimization techniques for VQAs goes well beyond the scope of this conclusion, but a particular example of the sensitivity to details is Ref. [129]. This work contains numerical experiments with notable similarity to my work in [15] regarding the considered PQC, optimizers and problem Hamiltonians, and leads to somewhat conflicting statements about the optimization behaviour if generalized carelessly. This demonstrates the importance of separating the evaluation of optimization routines from other aspects of the full VQA and analysing them independently in order to arrive at a robust characterization – much like for estimation subroutines discussed above. I will not aim for a broader conclusion regarding VQAs, as they are the topic of a large area of research. Whether or for which task they offer computational advantage over classical methods is an open key question that may decide whether noisy intermediate-scale quantum devices will be of practical use in applications or remain a bridge technology towards fault-tolerant quantum computing.

²In Chap. 1, for example.

Acknowledgements

I would like to thank everyone that supported, encouraged and advised me during my studies and the various projects in the last four years. I am grateful for the continued guidance, advice and encouragement by Christian Gogolin and David Gross, who did not hesitate to academically adopt me and supported me at all times, as well as the motivation and direction Michael Kastoryano gave me to dive into the exciting field of quantum computing. In the friendly and relaxed yet productive atmosphere of David's group I had the pleasure to meet, discuss with and become friends with many amazing colleagues. In particular I want to thank Yaiza Aragonés, Chae-Yeun Park, Markus Heinrich, Felipe Montealegre Mora, Mariela Boevska, Angelos Bampounis, Lukas Franken, Laurens Ligthart, Mariami Gachechiladze, Nikolai Miklin, Arne Heimendahl, Vahideh Eshaghian, Paulina Goedicke, Fabian Henze and Mark Goh³ for the great time that were the last four years. Warm thanks to Thorben Frank for all the discussions, a lot of fun and constant motivation even in hoffice times, and to Johan Åberg for always helping me out from near and far, be it with research riddles, teaching troubles or writing work.

Furthermore I want to thank my collaborators and friends: the Berlin band consisting of Johannes Jakob Meyer, Elies Gil-Fuster, Tom Hubregtsen, Paul Fährmann and Peter-Jan Derks together with their advisor Jens Eisert, the chemistry crew with Gian-Luca Anselmetti and Fotios Gkritis as well as Rob Parrish from QC Ware and Cedric Lin and Cody Wang from AWS. During my residency at Xanadu, a sustainable and very enjoyable discussion and collaboration atmosphere developed, and I would like to thank Josh Izaac, Maria Schuld and Nathan Killoran for that in particular, among the whole team and the residents of 2021. Thank you, Josh, for the many great chats about gradients, software and all the tangents we went on.

Special thanks to Johan, Laurens, Anni, Christian, Isabel, Paulina and Vahideh for their valuable feedback on this manuscript.

I also would like to thank my parents, siblings, aunt and cousins for always being there and supporting me. They always encouraged and trusted me to search for my path and I could not wish for a nicer family. Finally, I want to thank my partner Isabel; for being there when things go wrong and for celebrating with me when they work out. For your support, your honesty and opinion, the endless laughter and fun, and for your love.

³ and Julius Zeiss.

I also want to acknowledge the numerous open source tools that helped me in the research projects and the writing of this thesis – or made some undertakings possible in the first place – and their creators, contributors and maintainers. Explicitly I want to mention JAX [78], Matplotlib [130], PennyLane [80], PyTorch [77], Quantikz [131], SciPy [132] and Seaborn [133]. Thanks!

Lastly, I acknowledge the use of the website wombo.art in the making of the cover background image [134].

Bibliography

- [1] Craig Gidney and Martin Ekerå. “How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits”. *Quantum* **5**, 433 (2021). Appearances: [ix](#)
- [2] Isaac H. Kim, Ye-Hua Liu, Sam Pallister, William Pol, Sam Roberts, and Eunseok Lee. “Fault-tolerant resource estimate for quantum chemical simulations: Case study on Li-ion battery electrolyte molecules”. *Phys. Rev. Research* **4**, 023019 (2022). Appearances: [ix](#)
- [3] Thaddeus D Ladd, Fedor Jelezko, Raymond Laflamme, Yasunobu Nakamura, Christopher Monroe, and Jeremy Lloyd O’Brien. “Quantum computers”. *nature* **464**, 45–53 (2010). [arXiv:1009.2267](#). Appearances: [ix](#)
- [4] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. “Quantum supremacy using a programmable superconducting processor”. *Nature* **574**, 505–510 (2019). Appearances: [ix](#)
- [5] Jay Gambetta. “IBM’s roadmap for scaling quantum technology”. url: research.ibm.com. (accessed: 02.11.2022). Appearances: [ix](#)
- [6] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Sergio Boixo, Michael Broughton, Bob B. Buckley, David A. Buell, Brian Burkett, Nicholas Bushnell, Yu Chen, Zijun Chen, Benjamin Chiaro, and Roberto Collins. “Hartree-Fock on a superconducting qubit quantum computer”. *Science* **369**, 1084–1089 (2020). [arXiv:2004.04174](#). Appearances: [ix](#)
- [7] Thomas E O’Brien, G Anselmetti, Fotios Gkritis, VE Elfving, Stefano Polla, William J Huggins, Oumarou Oumarou, Kostyantyn Kechedzhi, Dmitry Abanin, Rajeev Acharya, et al. “Purification-based quantum error mitigation of pair-correlated electron simulations” (2022). [arXiv:2210.10799](#). Appearances: [ix](#)
- [8] John Preskill. “Quantum Computing in the NISQ era and beyond”. *Quantum* **2**, 79 (2018). Appearances: [ix](#)
- [9] Edward Farhi and Aram W Harrow. “Quantum supremacy through the quantum approximate optimization algorithm” (2016). [arXiv:1602.07674](#). Appearances: [x](#)

- [10] Sergey Bravyi, David Gosset, Robert Koenig, and Marco Tomamichel. “Quantum advantage with noisy shallow circuits”. *Nature Physics* **16**, 1040–1045 (2020). [arXiv:1904.01502](https://arxiv.org/abs/1904.01502). Appearances: [x](#)
- [11] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. “Variational quantum algorithms”. *Nature Reviews Physics* **3**, 625–644 (2021). [arXiv:2012.09265](https://arxiv.org/abs/2012.09265). Appearances: [x](#), [83](#), [84](#)
- [12] Jules Tilly, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia, Ying Li, Edward Grant, Leonard Wossnig, Ivan Rungger, George H. Booth, and Jonathan Tennyson. “The Variational Quantum Eigensolver: A review of methods and best practices”. *Physics Reports* **986**, 1–128 (2022). Appearances: [x](#), [31](#), [83](#), [84](#)
- [13] Gian-Luca R Anselmetti, David Wierichs, Christian Gogolin, and Robert M Parrish. “Local, expressive, quantum-number-preserving VQE ansätze for fermionic systems”. *New Journal of Physics* **23**, 113010 (2021). Appearances: [xi](#), [2](#), [45](#), [46](#), [48](#), [49](#), [50](#), [51](#), [83](#), [123](#)
- [14] David Wierichs, Josh Izaac, Cody Wang, and Cedric Yen-Yu Lin. “General parameter-shift rules for quantum gradients”. *Quantum* **6**, 677 (2022). Appearances: [xi](#), [46](#), [54](#), [123](#)
- [15] David Wierichs, Christian Gogolin, and Michael Kastoryano. “Avoiding local minima in variational quantum eigensolvers with the natural gradient optimizer”. *Phys. Rev. Research* **2**, 043246 (2020). Appearances: [xi](#), [83](#), [84](#), [86](#), [106](#), [123](#)
- [16] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. “A quantum approximate optimization algorithm” (2014). [arXiv:1411.4028](https://arxiv.org/abs/1411.4028). Appearances: [2](#), [83](#)
- [17] Bryan T Gard, Linghua Zhu, George S Barron, Nicholas J Mayhall, Sophia E Economou, and Edwin Barnes. “Efficient symmetry-preserving state preparation circuits for the variational quantum eigensolver algorithm”. *npj Quantum Information* **6**, 1–9 (2020). Appearances: [2](#), [83](#)
- [18] George S. Barron, Bryan T. Gard, Orien J. Altman, Nicholas J. Mayhall, Edwin Barnes, and Sophia E. Economou. “Preserving Symmetries for Variational Quantum Eigensolvers in the Presence of Noise”. *Phys. Rev. Applied* **16**, 034003 (2021). [arXiv:2003.00171](https://arxiv.org/abs/2003.00171). Appearances: [2](#)
- [19] “Compute resources of IBM Washington”. url: quantum-computing.ibm.com. (accessed: 31.10.2022). Appearances: [2](#)
- [20] J. Hilder, D. Pijn, O. Onishchenko, A. Stahl, M. Orth, B. Lekitsch, A. Rodriguez-Blanco, M. Müller, F. Schmidt-Kaler, and U. G. Poschinger. “Fault-Tolerant Parity

- Readout on a Shuttling-Based Trapped-Ion Quantum Computer". *Phys. Rev. X* **12**, 011032 (2022). Appearances: 2
- [21] "IonQ - Introducing the Native Gates". url: ionq.com/docs. (accessed: 01.11.2022). Appearances: 2
- [22] "Google devices - Two Qubit Gates". url: quantumai.google/cirq. (accessed: 01.11.2022). Appearances: 2
- [23] Francisco Javier Gil Vidal and Dirk Oliver Theis. "Input redundancy for parameterized quantum circuits". *Frontiers in Physics* **8**, 297 (2020). Appearances: 3
- [24] Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. "Effect of data encoding on the expressive power of variational quantum-machine-learning models". *Phys. Rev. A* **103**, 032430 (2021). [arXiv:2008.08605](https://arxiv.org/abs/2008.08605). Appearances: 3, 84
- [25] Javier Gil Vidal and Dirk Oliver Theis. "Calculus on parameterized quantum circuits" (2018). [arXiv:1812.06323](https://arxiv.org/abs/1812.06323). Appearances: 3, 45, 46, 47, 85
- [26] Oleksandr Kyriienko and Vincent E. Elfving. "Generalized quantum circuit differentiation rules". *Phys. Rev. A* **104**, 052417 (2021). [arXiv:2108.01218](https://arxiv.org/abs/2108.01218). Appearances: 3, 47
- [27] Dirk Oliver Theis. "Optimality of Finite Parameter Shift Rules for Derivatives of Variational Quantum Circuits" (2021). [arXiv:2112.14669](https://arxiv.org/abs/2112.14669). Appearances: 3, 9, 19, 20, 47
- [28] Robert M Parrish, Joseph T Iosue, Asier Ozaeta, and Peter L McMahon. "A Jacobi diagonalization and Anderson acceleration algorithm for variational quantum algorithm parameter optimization" (2019). [arXiv:1904.03206](https://arxiv.org/abs/1904.03206). Appearances: 3, 47, 85
- [29] Ken M. Nakanishi, Keisuke Fujii, and Syngae Todo. "Sequential minimal optimization for quantum-classical hybrid algorithms". *Phys. Rev. Research* **2**, 043158 (2020). Appearances: 3, 47, 85
- [30] Mateusz Ostaszewski, Edward Grant, and Marcello Benedetti. "Structure optimization for parameterized quantum circuits". *Quantum* **5**, 391 (2021). Appearances: 3, 47, 85
- [31] Bálint Koczor and Simon C. Benjamin. "Quantum analytic descent". *Phys. Rev. Research* **4**, 023017 (2022). Appearances: 3, 85
- [32] Kaito Wada, Rudy Raymond, Yuki Sato, and Hiroshi C Watanabe. "Full optimization of a single-qubit gate on the generalized sequential quantum optimizer" (2022). [arXiv:2209.08535](https://arxiv.org/abs/2209.08535). Appearances: 3, 85

- [33] Enrico Fontana, Ivan Rungger, Ross Duncan, and Cristina Cîrstoiu. “Spectral analysis for noise diagnostics and filter-based digital error mitigation” (2022). [arXiv:2206.08811](#). Appearances: 3
- [34] Franz J Schreiber, Jens Eisert, and Johannes Jakob Meyer. “Classical surrogates for quantum learning models” (2022). [arXiv:2206.11740](#). Appearances: 3, 4
- [35] Enrico Fontana, Ivan Rungger, Ross Duncan, and Cristina Cîrstoiu. “Efficient recovery of variational quantum algorithms landscapes using classical signal processing” (2022). [arXiv:2208.05958](#). Appearances: 3, 43
- [36] Leonardo Banchi and Gavin E. Crooks. “Measuring Analytic Gradients of General Quantum Evolution with the Stochastic Parameter Shift Rule”. *Quantum* **5**, 386 (2021). Appearances: 3, 17, 46, 47, 48
- [37] Dirk Oliver Theis. “Proper Shift Rules for Derivatives of Perturbed-Parametric Quantum Evolutions” (2022). [arXiv:2207.01587](#). Appearances: 3, 17, 19, 47
- [38] Adrián Pérez-Salinas, Alba Cervera-Lierta, Elies Gil-Fuster, and José I. Latorre. “Data re-uploading for a universal quantum classifier”. *Quantum* **4**, 226 (2020). Appearances: 4
- [39] Pranav Gokhale, Olivia Angiuli, Yongshan Ding, Kaiwen Gui, Teague Tomesh, Martin Suchara, Margaret Martonosi, and Frederic T Chong. “Minimizing state preparations in variational quantum eigensolver by partitioning into commuting families” (2019). [arXiv:1907.13623](#). Appearances: 6, 7
- [40] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. “The theory of variational hybrid quantum-classical algorithms”. *New Journal of Physics* **18**, 023023 (2016). Appearances: 6, 7
- [41] Nicholas C Rubin, Ryan Babbush, and Jarrod McClean. “Application of fermionic marginal constraints to hybrid quantum algorithms”. *New Journal of Physics* **20**, 053020 (2018). Appearances: 6
- [42] William J Huggins, Jarrod R McClean, Nicholas C Rubin, Zhang Jiang, Nathan Wiebe, K Birgitta Whaley, and Ryan Babbush. “Efficient and noise resilient measurements for quantum chemistry on near-term quantum computers”. *npj Quantum Information* **7**, 1–9 (2021). Appearances: 6, 7, 12, 25, 31, 83
- [43] Vojtěch Havlíček, Antonio D Córcoles, Kristan Temme, Aram W Harrow, Abhinav Kandala, Jerry M Chow, and Jay M Gambetta. “Supervised learning with quantum-enhanced feature spaces”. *Nature* **567**, 209–212 (2019). [arXiv:1804.11326](#). Appearances: 7

- [44] Stefano Polla, Gian-Luca R Anselmetti, and Thomas E O'Brien. "Optimizing the information extracted by a single qubit measurement" (2022). [arXiv:2207.09479](#). Appearances: 7, 23
- [45] Zhenyu Cai. "Resource-efficient purification-based quantum error mitigation" (2021). [arXiv:2107.07279](#). Appearances: 7, 23
- [46] Thomas E. O'Brien, Stefano Polla, Nicholas C. Rubin, William J. Huggins, Sam McArdle, Sergio Boixo, Jarrod R. McClean, and Ryan Babbush. "Error Mitigation via Verified Phase Estimation". *PRX Quantum* **2**, 020317 (2021). Appearances: 7, 23
- [47] Mingxia Huo and Ying Li. "Dual-state purification for practical quantum error mitigation". *Phys. Rev. A* **105**, 022427 (2022). [arXiv:2105.01239](#). Appearances: 7, 23
- [48] Bryan O'Gorman, William J Huggins, Eleanor G Rieffel, and K Birgitta Whaley. "Generalized swap networks for near-term quantum computing" (2019). [arXiv:1905.05118](#). Appearances: 7
- [49] Andrew Jena, Scott Genin, and Michele Mosca. "Pauli partitioning with respect to gate sets" (2019). [arXiv:1907.07859](#). Appearances: 7
- [50] Artur F. Izmaylov, Tzu-Ching Yen, and Ilya G. Ryabinkin. "Revising the measurement process in the variational quantum eigensolver: is it possible to reduce the number of separately measured operators?". *Chem. Sci.* **10**, 3746–3755 (2019). Appearances: 7
- [51] Artur F Izmaylov, Tzu-Ching Yen, Robert A Lang, and Vladyslav Verteletskyi. "Unitary partitioning approach to the measurement problem in the variational quantum eigensolver method". *Journal of chemical theory and computation* **16**, 190–195 (2019). [arXiv:1907.09040](#). Appearances: 7, 31
- [52] Vladyslav Verteletskyi, Tzu-Ching Yen, and Artur F. Izmaylov. "Measurement optimization in the variational quantum eigensolver using a minimum clique cover". *The Journal of Chemical Physics* **152**, 124114 (2020). [arXiv:1907.03358](#). Appearances: 7, 31
- [53] Tzu-Ching Yen, Vladyslav Verteletskyi, and Artur F Izmaylov. "Measuring all compatible operators in one series of single-qubit measurements using unitary transformations". *Journal of chemical theory and computation* **16**, 2400–2409 (2020). [arXiv:1907.09386](#). Appearances: 7
- [54] Alexis Ralli, Peter J. Love, Andrew Tranter, and Peter V. Coveney. "Implementation of measurement reduction for the variational quantum eigensolver". *Phys. Rev. Research* **3**, 033195 (2021). Appearances: 7

- [55] Ophelia Crawford, Barnaby van Straaten, Daochen Wang, Thomas Parks, Earl Campbell, and Stephen Brierley. “Efficient quantum measurement of Pauli operators in the presence of finite sampling error”. *Quantum* **5**, 385 (2021). Appearances: 7
- [56] Andrea Mari, Thomas R. Bromley, and Nathan Killoran. “Estimating the gradient and higher-order derivatives on quantum hardware”. *Phys. Rev. A* **103**, 012405 (2021). [arXiv:2008.06517](https://arxiv.org/abs/2008.06517). Appearances: 8, 9, 12, 20, 21, 32, 34, 35, 42, 46, 48, 49, 53
- [57] Lennart Bittel, Jens Watty, and Martin Kliesch. “Fast gradient estimation for variational quantum algorithms” (2022). [arXiv:2210.06484](https://arxiv.org/abs/2210.06484). Appearances: 9, 21, 32, 42
- [58] William J Huggins, Joonho Lee, Unpil Baek, Bryan O’Gorman, and K Birgitta Whaley. “A non-orthogonal variational quantum eigensolver”. *New Journal of Physics* **22**, 073009 (2020). Appearances: 12
- [59] Jonas M. Kübler, Andrew Arrasmith, Lukasz Cincio, and Patrick J. Coles. “An Adaptive Optimizer for Measurement-Frugal Variational Algorithms”. *Quantum* **4**, 263 (2020). Appearances: 12
- [60] Barnaby van Straaten and Bálint Koczor. “Measurement Cost of Metric-Aware Variational Quantum Algorithms”. *PRX Quantum* **2**, 030324 (2021). Appearances: 12, 31
- [61] Donald E. Knuth. “The Art of Computer Programming, Volume 1 (3rd Ed.): Fundamental Algorithms”. **Chapter 1.2.3, pages 38, Ex.40**. Addison Wesley Longman Publishing Co., Inc. USA (1997). Appearances: 14
- [62] Jun Li, Xiaodong Yang, Xinhua Peng, and Chang-Pu Sun. “Hybrid Quantum-Classical Approach to Quantum Optimal Control”. *Phys. Rev. Lett.* **118**, 150503 (2017). [arXiv:1608.00677](https://arxiv.org/abs/1608.00677). Appearances: 17, 45, 48
- [63] G. Ortiz, J. E. Gubernatis, E. Knill, and R. Laflamme. “Quantum algorithms for fermionic simulations”. *Phys. Rev. A* **64**, 022319 (2001). [arXiv:cond-mat/0012334](https://arxiv.org/abs/cond-mat/0012334). Appearances: 21
- [64] R. Somma, G. Ortiz, J. E. Gubernatis, E. Knill, and R. Laflamme. “Simulating physical phenomena by quantum networks”. *Phys. Rev. A* **65**, 042323 (2002). [arXiv:quant-ph/0108146](https://arxiv.org/abs/quant-ph/0108146). Appearances: 21
- [65] Artur K. Ekert, Carolina Moura Alves, Daniel K. L. Oi, Michał Horodecki, Paweł Horodecki, and L. C. Kwek. “Direct Estimations of Linear and Nonlinear Functionals of a Quantum State”. *Phys. Rev. Lett.* **88**, 217901 (2002). [arXiv:quant-ph/0203016](https://arxiv.org/abs/quant-ph/0203016). Appearances: 21

- [66] Gian Giacomo Guerreschi and Mikhail Smelyanskiy. “Practical optimization for hybrid quantum-classical algorithms” (2017). [arXiv:1701.01450](#). Appearances: 22
- [67] Ying Li and Simon C. Benjamin. “Efficient Variational Quantum Simulator Incorporating Active Error Minimization”. *Phys. Rev. X* **7**, 021050 (2017). Appearances: 22
- [68] Jonathan Romero, Ryan Babbush, Jarrod R McClean, Cornelius Hempel, Peter J Love, and Alán Aspuru-Guzik. “Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz”. *Quantum Science and Technology* **4**, 014008 (2018). [arXiv:1701.02691](#). Appearances: 22
- [69] Sirui Lu, Mari Carmen Bañuls, and J. Ignacio Cirac. “Algorithms for Quantum Simulation at Finite Energies”. *PRX Quantum* **2**, 020321 (2021). Appearances: 25
- [70] Andrew M Childs and Nathan Wiebe. “Hamiltonian simulation using linear combinations of unitary operations”. *Quantum Information and Computation* **12**, 0901–0924 (2012). [arXiv:1202.5822](#). Appearances: 26
- [71] J.C. Spall. “Multivariate stochastic approximation using a simultaneous perturbation gradient approximation”. *IEEE Transactions on Automatic Control* **37**, 332–341 (1992). Appearances: 28
- [72] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta. “Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets”. *Nature* **549**, 242–246 (2017). [arXiv:1704.05018](#). Appearances: 28, 84
- [73] Xavier Bonet-Monroig, Hao Wang, Diederick Vermetten, Bruno Senjean, Charles Moussa, Thomas Bäck, Vedran Dunjko, and Thomas E O’Brien. “Performance comparison of optimization methods on variational quantum algorithms” (2021). [arXiv:2111.13454](#). Appearances: 28, 84
- [74] Donghwa Lee, Jinil Lee, Seongjin Hong, Hyang-Tag Lim, Young-Wook Cho, Sang-Wook Han, Hyundong Shin, Junaid ur Rehman, and Yong-Su Kim. “Error-mitigated photonic variational quantum eigensolver using a single-photon ququart”. *Optica* **9**, 88–95 (2022). Appearances: 28
- [75] Julien Gacon, Christa Zoufal, Giuseppe Carleo, and Stefan Woerner. “Simultaneous Perturbation Stochastic Approximation of the Quantum Fisher Information”. *Quantum* **5**, 567 (2021). Appearances: 28
- [76] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, et al. (2015). code: [tensorflow/tensorflow](#). Appearances: 29, 30

- [77] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. “Pytorch: An imperative style, high-performance deep learning library”. *Advances in neural information processing systems* **32** (2019). code: [pytorch/pytorch](#). Appearances: [29](#), [108](#)
- [78] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang (2018). code: [google/jax](#). Appearances: [29](#), [108](#)
- [79] autodiff community authors (2021). code: [autodiff/autodiff](#). Appearances: [29](#)
- [80] Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, M Sohaib Alam, Shah Nawaz Ahmed, Juan Miguel Arrazola, Carsten Blank, Alain Delgado, Soran Jangiri, et al. “PennyLane: Automatic differentiation of hybrid quantum-classical computations” (2018). [arXiv:1811.04968](#). code: [PennyLaneAI/pennylane](#). Appearances: [29](#), [30](#), [36](#), [49](#), [54](#), [108](#)
- [81] Xiu-Zhe Luo, Jin-Guo Liu, Pan Zhang, and Lei Wang. “Yao.jl: Extensible, Efficient Framework for Quantum Algorithm Design”. *Quantum* **4**, 341 (2020). code: [QuantumBFS/Yao.jl](#). Appearances: [29](#), [30](#)
- [82] Michael Broughton, Guillaume Verdon, Trevor McCourt, Antonio J Martinez, Jae Hyeon Yoo, Sergei V Isakov, Philip Massey, Ramin Halavati, Murphy Yuezhen Niu, Alexander Zlokapa, et al. “Tensorflow quantum: A software framework for quantum machine learning” (2020). [arXiv:2003.02989](#). code: [tensorflow/quantum](#). Appearances: [29](#)
- [83] Qiskit community authors (2021). code: [Qiskit/qiskit](#). Appearances: [29](#)
- [84] Charles C Margossian. “A review of automatic differentiation and its efficient implementation”. *Wiley interdisciplinary reviews: data mining and knowledge discovery* **9**, e1305 (2019). [arXiv:1811.05031](#). Appearances: [29](#)
- [85] Dougal Maclaurin, David Duvenaud, and Ryan Adams. “Gradient-based hyperparameter optimization through reversible learning”. In International conference on machine learning. Pages 2113–2122. PMLR (2015). [arXiv:1502.03492](#). Appearances: [29](#)
- [86] Tyson Jones and Julien Gacon. “Efficient calculation of gradients in classical simulations of variational quantum algorithms” (2020). [arXiv:2009.02823](#). Appearances: [29](#), [30](#)

- [87] Amazon Web Services. “Amazon Braket”. url: aws.amazon.com/braket/. Appearances: 30
- [88] Ryan Sweke, Frederik Wilde, Johannes Meyer, Maria Schuld, Paul K. Faehrmann, Barthélemy Meynard-Piganeau, and Jens Eisert. “Stochastic gradient descent for hybrid quantum-classical optimization”. *Quantum* **4**, 314 (2020). Appearances: 31
- [89] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. “Barren plateaus in quantum neural network training landscapes”. *Nature communications* **9**, 1–6 (2018). Appearances: 36, 84
- [90] James Stokes, Josh Izaac, Nathan Killoran, and Giuseppe Carleo. “Quantum Natural Gradient”. *Quantum* **4**, 269 (2020). Appearances: 42
- [91] C. G. Broyden. “The Convergence of a Class of Double-rank Minimization Algorithms. 1. General Considerations”. *IMA Journal of Applied Mathematics* **6**, 76–90 (1970). Appearances: 42, 84
- [92] R. Fletcher. “A new approach to variable metric algorithms”. *The Computer Journal* **13**, 317–322 (1970). Appearances: 42, 84
- [93] Donald Goldfarb. “A family of variable-metric methods derived by variational means”. *Mathematics of Computation* **24**, 23–26 (1970). Appearances: 42, 84
- [94] D. F. Shanno. “Conditioning of quasi-Newton methods for function minimization”. *Mathematics of Computation* **24**, 647–656 (1970). Appearances: 42, 84
- [95] David Wierichs (2022). code: [dwierichs/Derivative-Estimator-Comparison](https://github.com/dwierichs/Derivative-Estimator-Comparison). Appearances: 43
- [96] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii. “Quantum circuit learning”. *Phys. Rev. A* **98**, 032309 (2018). [arXiv:1803.00745](https://arxiv.org/abs/1803.00745). Appearances: 45, 48
- [97] Jin-Guo Liu and Lei Wang. “Differentiable learning of quantum circuit Born machines”. *Phys. Rev. A* **98**, 062324 (2018). [arXiv:1804.04168](https://arxiv.org/abs/1804.04168). Appearances: 45
- [98] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. “Evaluating analytic gradients on quantum hardware”. *Phys. Rev. A* **99**, 032331 (2019). [arXiv:1811.11184](https://arxiv.org/abs/1811.11184). Appearances: 45, 48
- [99] Gavin E Crooks. “Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition” (2019). [arXiv:1905.13311](https://arxiv.org/abs/1905.13311). Appearances: 46
- [100] Patrick Huembeli and Alexandre Dauphin. “Characterizing the loss landscape of variational quantum circuits”. *Quantum Science and Technology* **6**, 025011 (2021). [arXiv:2008.02785](https://arxiv.org/abs/2008.02785). Appearances: 46

- [101] Jakob S. Kottmann, Abhinav Anand, and Alán Aspuru-Guzik. “A feasible approach for automatically differentiable unitary coupled-cluster on quantum computers”. *Chem. Sci.* **12**, 3497 (2021). Appearances: 46, 48, 51, 54
- [102] Thomas Hubregtsen, Frederik Wilde, Shozab Qasim, and Jens Eisert. “Single-component gradient rules for variational quantum algorithms”. *Quantum Science and Technology* **7**, 035008 (2022). [arXiv:2106.01388v1](https://arxiv.org/abs/2106.01388v1). Appearances: 46
- [103] Artur F. Izmaylov, Robert A. Lang, and Tzu-Ching Yen. “Analytic gradients in variational quantum algorithms: Algebraic extensions of the parameter-shift rule to general unitary transformations”. *Phys. Rev. A* **104**, 062443 (2021). [arXiv:2107.08131](https://arxiv.org/abs/2107.08131). Appearances: 46
- [104] Andrea Skolik, Jarrod R. McClean, Masoud Mohseni, Patrick van der Smagt, and Martin Leib. “Layerwise learning for quantum neural networks”. *Quantum Machine Intelligence* **3**, 1–11 (2021). Appearances: 47
- [105] Marcello Benedetti, Mattia Fiorentini, and Michael Lubasch. “Hardware-efficient variational quantum algorithms for time evolution”. *Phys. Rev. Research* **3**, 033083 (2021). Appearances: 47
- [106] Ernesto Campos, Aly Nasrallah, and Jacob Biamonte. “Abrupt transitions in variational quantum circuit training”. *Phys. Rev. A* **103**, 032607 (2021). [arXiv:2010.09720](https://arxiv.org/abs/2010.09720). Appearances: 47
- [107] Johannes Jakob Meyer, Johannes Borregaard, and Jens Eisert. “A variational toolbox for quantum multi-parameter estimation”. *npj Quantum Information* **7**, 1–5 (2021). Appearances: 48
- [108] Dave Wecker, Matthew B. Hastings, and Matthias Troyer. “Progress towards practical quantum variational algorithms”. *Phys. Rev. A* **92**, 042303 (2015). [arXiv:1507.08969](https://arxiv.org/abs/1507.08969). Appearances: 83
- [109] M. Ganzhorn, D.J. Egger, P. Barkoutsos, P. Ollitrault, G. Salis, N. Moll, M. Roth, A. Fuhrer, P. Mueller, S. Woerner, I. Tavernelli, and S. Filipp. “Gate-Efficient Simulation of Molecular Eigenstates on a Quantum Computer”. *Phys. Rev. Applied* **11**, 044092 (2019). [arXiv:1809.05057](https://arxiv.org/abs/1809.05057). Appearances: 83
- [110] Yordan S. Yordanov, David R. M. Arvidsson-Shukur, and Crispin H. W. Barnes. “Efficient quantum circuits for quantum computational chemistry”. *Phys. Rev. A* **102**, 062612 (2020). [arXiv:2005.14475](https://arxiv.org/abs/2005.14475). Appearances: 83
- [111] Zoë Holmes, Kunal Sharma, M. Cerezo, and Patrick J. Coles. “Connecting Ansatz Expressibility to Gradient Magnitudes and Barren Plateaus”. *PRX Quantum* **3**, 010313 (2022). Appearances: 84

- [112] Matthias C. Caro, Elies Gil-Fuster, Johannes Jakob Meyer, Jens Eisert, and Ryan Sweke. “Encoding-dependent generalization bounds for parametrized quantum circuits”. *Quantum* **5**, 582 (2021). Appearances: 84
- [113] Sumeet Khatri, Ryan LaRose, Alexander Poremba, Lukasz Cincio, Andrew T. Sornborger, and Patrick J. Coles. “Quantum-assisted quantum compiling”. *Quantum* **3**, 140 (2019). Appearances: 84
- [114] Kaoru Mizuta, Yuya O. Nakagawa, Kosuke Mitarai, and Keisuke Fujii. “Local Variational Quantum Compilation of Large-Scale Hamiltonian Dynamics”. *PRX Quantum* **3**, 040302 (2022). Appearances: 84
- [115] Edward Grant, Leonard Wossnig, Mateusz Ostaszewski, and Marcello Benedetti. “An initialization strategy for addressing barren plateaus in parametrized quantum circuits”. *Quantum* **3**, 214 (2019). Appearances: 84
- [116] Stefan H. Sack and Maksym Serbyn. “Quantum annealing initialization of the quantum approximate optimization algorithm”. *Quantum* **5**, 491 (2021). Appearances: 84
- [117] Jonathan Wurtz and Peter J. Love. “Counterdiabaticity and the quantum approximate optimization algorithm”. *Quantum* **6**, 635 (2022). Appearances: 84
- [118] Fernando GSL Brandao, Michael Broughton, Edward Farhi, Sam Gutmann, and Hartmut Neven. “For fixed control parameters the quantum approximate optimization algorithm’s objective function value concentrates for typical instances” (2018). [arXiv:1812.04170](https://arxiv.org/abs/1812.04170). Appearances: 84
- [119] Leo Zhou, Sheng-Tao Wang, Soonwon Choi, Hannes Pichler, and Mikhail D. Lukin. “Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices”. *Phys. Rev. X* **10**, 021067 (2020). Appearances: 84
- [120] Michael Streif and Martin Leib. “Training the quantum approximate optimization algorithm without access to a quantum processing unit”. *Quantum Science and Technology* **5**, 034008 (2020). [arXiv:1908.08862](https://arxiv.org/abs/1908.08862). Appearances: 84
- [121] Marco Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J Coles. “Cost function dependent barren plateaus in shallow parametrized quantum circuits”. *Nature communications* **12**, 1–12 (2021). Appearances: 84
- [122] Andrew Arrasmith, M. Cerezo, Piotr Czarnik, Lukasz Cincio, and Patrick J. Coles. “Effect of barren plateaus on gradient-free optimization”. *Quantum* **5**, 558 (2021). Appearances: 84
- [123] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization” (2014). [arXiv:1412.6980](https://arxiv.org/abs/1412.6980). Appearances: 84

- [124] J. A. Nelder and R. Mead. “A Simplex Method for Function Minimization”. *The Computer Journal* **7**, 308–313 (1965). Appearances: 85
- [125] Kevin J Sung, Jiahao Yao, Matthew P Harrigan, Nicholas C Rubin, Zhang Jiang, Lin Lin, Ryan Babbush, and Jarrod R McClean. “Using models to improve optimizers for variational quantum algorithms”. *Quantum Science and Technology* **5**, 044008 (2020). [arXiv:2005.11011](https://arxiv.org/abs/2005.11011). Appearances: 85
- [126] Harper R Grimsley, Sophia E Economou, Edwin Barnes, and Nicholas J Mayhall. “An adaptive variational algorithm for exact molecular simulations on a quantum computer”. *Nature communications* **10**, 1–9 (2019). Appearances: 85
- [127] Roeland Wiersema and Nathan Killoran. “Optimizing quantum circuits with riemannian gradient-flow” (2022). [arXiv:2202.06976](https://arxiv.org/abs/2202.06976). Appearances: 85
- [128] Stefan H Sack, Raimel A Medina, Richard Kueng, and Maksym Serbyn. “Transition states and greedy exploration of the QAOA optimization landscape” (2022). [arXiv:2209.01159](https://arxiv.org/abs/2209.01159). Appearances: 85
- [129] Roeland Wiersema, Cunlu Zhou, Yvette de Sereville, Juan Felipe Carrasquilla, Yong Baek Kim, and Henry Yuen. “Exploring Entanglement and Optimization within the Hamiltonian Variational Ansatz”. *PRX Quantum* **1**, 020319 (2020). Appearances: 106
- [130] J. D. Hunter. “Matplotlib: A 2D graphics environment”. *Computing in Science & Engineering* **9**, 90–95 (2007). Appearances: 108
- [131] Alastair Kay. “Tutorial on the quantikz package” (2018). [arXiv:1809.03842](https://arxiv.org/abs/1809.03842). code: <https://ctan.org/pkg/quantikz>. Appearances: 108
- [132] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, and SciPy 1.0 Contributors. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. *Nature Methods* **17**, 261–272 (2020). Appearances: 108
- [133] Michael L. Waskom. “seaborn: statistical data visualization”. *Journal of Open Source Software* **6**, 3021 (2021). Appearances: 108
- [134] “dream by Wombo”. url: wombo.art. (accessed: 24.10.2022). Appearances: 108, 123
- [135] Thomas Hubregtsen, David Wierichs, Elies Gil-Fuster, Peter-Jan H. S. Derks, Paul K. Faehrmann, and Johannes Jakob Meyer. “Training quantum embedding kernels on near-term quantum computers”. *Phys. Rev. A* **106**, 042431 (2022). [arXiv:2105.02276](https://arxiv.org/abs/2105.02276). Appearances: 123

Formalia

Zusammenfassung in deutscher Sprache

Universelles Quantenrechnen auf fehlerkorrigierenden Rechnern verspricht maßgebliche, beweisbare Vorteile gegenüber klassischen Rechnern. Auf dem Weg zu diesem Ziel befindet sich die heutige Quantentechnologie jedoch noch in der Frühphase. Nichtsdestotrotz könnte es sein, dass die fehlerbehafteten Quantengeräte, die in der nahen Zukunft gebaut werden werden, bereits relevante Berechnungen durchführen können, welche klassische Methoden in ihrer zeitlichen, räumlichen oder Energieeffizienz überbieten. Diese Frage, ob die ersten Quantengeräte neben ihrer Funktion als Brückentechnologie auch für Anwendungen nutzbar sein könnten, begründet die Forschung an sogenannten verrauschten Quantengeräten und -algorithmen der Übergangsgröße (NISQ-Geräte und -Algorithmen). Ein Großteil dieser Forschung behandelt variationelle Quantenalgorithmen (VQA), welche diese Geräte mit klassischen Rechnern zu einer hybriden Technologie verknüpfen. Dabei wird das zu lösende Problem als Observable, üblicherweise als Hamilton-Operator, formuliert, sodass eine Lösung gefunden würde, indem man den Grundzustand (oder seine Energie) ermittelte. Für die Suche nach einem geeigneten Zustand wählt der klassische Rechner einen Quantenschaltkreis aus einer Schar von Schaltkreisen aus und der Quantenrechner führt diesen aus, um den zugehörigen Zustand herzustellen. Eine solche Schar von Schaltkreisen ist gewöhnlich durch einen parametrisierten Quantenschaltkreis (PQS) gegeben, und die Wahl eines Schaltkreises entspricht einer Konfiguration seiner Parameter. Als Antwort auf eine Konfiguration erhält der klassische Rechner vom Quantengerät die Ergebnisse von Messungen ausgewählter Observablen. Im Anschluss optimiert der variationelle Algorithmus die Parameter anhand der erhaltenen Messwerte, um den Lösungszustand – oder eine hinreichende Näherung dessen – zu erhalten. Mittlerweile gibt es viele Varianten von VQA, die sich durch die verwendeten Teilmethoden sowie ihre Anwendungsfälle unterscheiden und zu einer modularen Struktur der Algorithmen geführt haben.

Das erste Kapitel dieser Arbeit behandelt Schätzer für Gradienten von Zielfunktionen, die auf PQS basieren. Dazu wird zu Beginn eine Reihe (komponentenweiser) Schätzer eingeführt und anschließend anhand einer weit verbreiteten Beispielklasse von PQS verglichen. Die Schlussfolgerungen aus diesem Kapitel decken sich mit denen aus aktuellen

Forschungsergebnissen und sind von Relevanz für die Praxis des Schätzens von Gradienten für VQA.

Das zweite Kapitel führt das Thema der Schätzung von Ableitungen weiter und befasst sich mit Ableitungen durch Parameterrückung, beginnend mit einer Diskussion der Literatur. Der anschließende Teil enthält eine Erweiterung dieser Methode zur Anwendung auf spezifische Quantengatter, die in Rechnungen zur Quantenchemie Verwendungen finden. Den Hauptteil des Kapitels stellt der Abdruck einer Publikation dar, welche weitere Verallgemeinerungen der Ableitung durch Parameterrückung behandelt. Ein besonderer Fokus liegt dabei auf der Analyse der Kosten für diese Schätzer, mit Bezug auf die Verwendung auf klassischen Simulatoren sowie auf Quantengeräten.

Das dritte und letzte Kapitel beginnt mit einer kurzen Einführung von VQA, welche den Zusammenhang zu einer zweiten Publikation herstellt. Letztere analysiert eine Reihe von Algorithmen, die in VQA in der Optimierungsphase auf dem klassischen Rechner Verwendung finden. Insbesondere werden anhand von numerischen Experimenten etablierte Methoden der klassischen nicht-konvexen Optimierung und des maschinellen Lernens mit dem Gradientenverfahren mittels des sogenannten natürlichen Gradienten verglichen. Dabei zeigt die spezialisierte Methode, die den natürlichen Gradienten nutzt, ein zuverlässigeres Verhalten während der Optimierung, indem sie die Konvergenz zu lokalen Minima vermeidet und auch symmetriebrechende Quantenschaltkreise erfolgreich optimiert. Die vorgestellten Experimente untersuchen dabei ein weit verbreitetes Beispiel eines VQA, den variationellen Eigenwertlöser (VQE), angewendet auf Spinkettensysteme als Testproblem.

Vielversprechende Themen für die zukünftige Forschung sind zum Einen die Untersuchung einzelner Bausteine von VQA und zum Anderen die Entwicklung akkurater Deskriptoren für diese Bausteine, die eine Vorhersage ihrer Eigenschaften im Kontext des gesamten Algorithmus ermöglichen. Dadurch könnten die Komplexität einzelner Forschungsfragen reduziert und Erkenntnisse ermöglicht werden, die an die modulare Struktur von VQA angepasst sind.

Erklärung zur Dissertation

Hiermit versichere ich an Eides statt, dass ich die vorliegende Dissertation selbstständig und ohne die Benutzung anderer als der angegebenen Hilfsmittel und Literatur angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Werken dem Wortlaut oder dem Sinn nach entnommen wurden, sind als solche kenntlich gemacht. Ich versichere an Eides statt, dass diese Dissertation noch keiner anderen Fakultät oder Universität zur Prüfung vorgelegen hat; dass sie – abgesehen von unten angegebenen Teilpublikationen und eingebundenen Artikeln und Manuskripten – noch nicht veröffentlicht worden ist sowie, dass ich eine Veröffentlichung der Dissertation vor Abschluss der Promotion nicht ohne Genehmigung des Promotionsausschusses vornehmen werde. Die Bestimmungen dieser Ordnung sind mir bekannt. Darüber hinaus erkläre ich hiermit, dass ich die Ordnung zur Sicherung guter wissenschaftlicher Praxis und zum Umgang mit wissenschaftlichem Fehlverhalten der Universität zu Köln gelesen und sie bei der Durchführung der Dissertation zugrundeliegenden Arbeiten und der schriftlich verfassten Dissertation beachtet habe und verpflichtete mich hiermit, die dort genannten Vorgaben bei allen wissenschaftlichen Tätigkeiten zu beachten und umzusetzen. Ich versichere, dass die eingereichte elektronische Fassung der eingereichten Druckfassung vollständig entspricht.

Teilpublikationen

Zusätzlich zu den vollständig eingebundenen Artikeln in den Kapiteln 2 [14] und 3 [15] habe ich den Appendix F der Publikation “Local, expressive, quantum-number-preserving VQE ansätze for fermionic systems” [13] verwendet und leicht modifiziert in Abschnitt 2.2 eingebunden. Der Vollständigkeit halber sei der Artikel “Training quantum embedding kernels on near-term quantum computers” [135] als Hilfsmittel erwähnt, den ich mitverfasst habe und der in *Physical Review A* publiziert wurde. Dieser Artikel ist jedoch nicht direkt inhaltlich in diese Dissertation eingeflossen, und insbesondere nicht eingebunden. Die Hintergrundgrafik der Titelseite wurde mit Hilfe von “wombo.art” erstellt [134]. Für die Bibliographie habe ich eine Variante des Bibliographiestils des Journals *Quantum* verwendet.