

Software Between Quantum and Machine Learning – And Down to Pulses

Maja Franz^{*§}, Melvin Strobl^{*†}, Jonathan Hunz[†], Lukas Scheller[†],

Lucas van der Horst[†], Eileen Kuehn[†], Achim Streit[†], Wolfgang Mauerer^{§¶}

[§] Technical University of Applied Sciences Regensburg, Germany, {maja.franz, wolfgang.mauerer}@othr.de

[†] Karlsruhe Institute of Technology, Germany, {melvin.strobl, lukas.scheller, eileen.kuehn, achim.streit}@kit.edu

[¶] Siemens AG, Technology, Munich, Germany

Abstract—Contemporary quantum computing platforms remain, in essence, programmable physical systems whose control is typically mediated through unitary gate abstractions. While such abstractions provide a uniform interface, they obscure important aspects of the underlying hardware and may limit the exploitation of its full capabilities. Direct operation at the level of control pulses offers a more expressive and physically faithful paradigm, enabling, for instance, the implementation of tailored error-mitigation and optimisation strategies. However, this increased expressivity comes at the cost of greater complexity from the perspective of quantum software development, necessitating structured and accessible tooling.

We present a software framework, integrated within the QML-ESSENTIALS package, that extends quantum machine learning (QML) methodologies to encompass pulse-level modelling. By embedding quantum optimal control (QOC) techniques within a QML setting, our approach enables the seamless combination of gate-based and pulse-level representations in a unified modelling paradigm. The framework provides a comprehensive suite of modelling and analytical capabilities. In particular, we introduce composable ansatz constructions based on interchangeable building blocks, alongside support for end-to-end optimisation of pulse parameters. Motivated by the central role of quantum Fourier models (QFMs), we further incorporate a range of Fourier-analytic diagnostics, including the Fourier fingerprint and the Fourier coefficient correlation (FCC), complemented by extended measures of entanglement. All performance-critical components are implemented in a high-performance environment using JAX and supported by a dedicated quantum simulator. Taken together, the framework facilitates reproducible and systematic investigations in QML research, while bridging the conceptual and practical divide between abstract circuit-based models and hardware-aware pulse-level optimisation. It provides a robust foundation for future developments at the intersection of quantum machine learning and quantum control.

Index Terms—Quantum Machine Learning, Quantum Computing, Quantum Optimal Control, Quantum Software Framework

I. INTRODUCTION

Parameterised quantum circuits (PQCs) constitute the core of contemporary quantum machine learning (QML), serving as function approximators across a variety of supervised and unsupervised machine learning (ML) tasks [1–7]. A Fourier-analytic perspective [8, 9] allows studying their expressive power (*i.e.*, the classes of functions that can be learned).

By characterising PQCs and their data encodings through their induced frequency spectra, *quantum Fourier models* (QFMs) provide a structured approach to understanding how quantum circuits realise function classes in terms of Fourier coefficients and frequencies. At the same time, a growing body of literature highlights the fundamental limitations in PQCs. For instance, *barren plateaus* [10, 11], or trainability degradation [12, 13] and restricted effective expressivity [14, 15] under realistic hardware constraints restrict the capabilities of quantum computing approaches in ML. Despite this increasing number of hardware-aware studies, we see a need for systematic tools that enable investigations beyond idealised quantum gates. In particular, while most QML studies assume unitary gate decompositions [16, 17], quantum processing units (QPUs), such as superconducting systems, operate at the level of control pulses. However, based on Refs. [18–24], we see a large potential in advancing QML research by directly learning pulse-level control sequences instead of gate angles.

We therefore aim to bridge the gap between the *high-level* circuit analysis metrics and the *low-level* pulse-level control with a software framework that (1) provides a modular and systematically composable QML model construction, (2) integrates rigorous diagnostic tools tailored to the Fourier-analytic perspective of QML, and (3) supports pulse-level modelling grounded in quantum optimal control (QOC). Rather than treating control pulses as an implementation detail, we expose them as differentiable objects, enabling end-to-end optimisation of pulse parameters. This enables the study of expressivity, trainability and other circuit properties, such as the entangling capability, directly at the level of physically realisable controls. Additionally, pulse-level learning paves the way for certain error mitigation techniques [25], where control signals are optimised to suppress coherent errors or reduce sensitivity to noise [26–28].

In this work, we introduce software tools in Python that provide a modular approach to variational ansatz design that can be instantiated by conventional unitary gates or as implicit pulse-level control sequences. Using this dual representation allows controlled comparisons between circuit-level models and hardware-aware implementations. Specifically, we extend the QML-ESSENTIALS package, introduced in Ref. [29], which already delivers some tools and metrics for circuit analysis from a Fourier-analytic perspective.

*Equal contribution

Building on QFMs as a central theme, we incorporate additional circuit metrics such as the Fourier fingerprint or the Fourier coefficient correlation (FCC) [30]. Furthermore, we expand the set of entangling capability measures by incorporating entanglement of formation (EF) and concentratable entanglement (CE), which are also applicable to mixed (noisy) quantum states.

To summarise, our software tools in QML-ESSENTIALS provide a comprehensive, reproducible [31] analysis and benchmarking suite for studying the relationship between entanglement, model expressivity, and efficient, hardware-agnostic circuit representations. Additionally, all performance-critical components are implemented with JAX [32] and supported by a dedicated and custom differentiable simulator for fast array computations.

The remainder of this article is structured as follows: [Sec. II](#) reviews the state of the art in QML-software with a focus on QFMs and pulse-level learning, followed by a brief introduction to QFMs in [Sec. III](#). In [Sec. IV](#), we then present an overview of our software and its integration in the QML-ESSENTIALS package from Ref. [29]. Throughout this article, novel features compared to QML-ESSENTIALS are marked with ⁺, and major changes with ⁽⁺⁾. A major part of our software is the tooling for pulse-level learning, which we present in [Sec. V](#). Subsequently, in [Sec. VI](#) we describe our implemented ansatz diagnostics metrics for the spectral properties, expressibility and entangling capability with some proof-of-concept results. Finally, we give an outlook on the possibilities of future work and experiments that utilise our toolkit and conclude in [Sec. VII](#).

II. RELATED WORK

The current state of the art in QML software can be roughly separated into three main categories, namely (1) general libraries for quantum circuit simulation or real hardware execution, at the core of established classical machine learning frameworks, (2) QML-tailored frameworks, which, in addition to quantum circuit execution, offer useful tools for ML-research, such as predefined datasets and differentiable programming, or (3) individual code setups accompanying research articles for specific tasks or reproduction, often leveraging tools from the first two categories.

General quantum frameworks, such as Qiskit [33], $t|ket\rangle$ [34], or Cirq [35], which offer comprehensive support for transpilation, real hardware execution, and stable circuit simulations, fall into the first category. These frameworks provide a variety of simulation, execution and analysis capabilities, and even some quantum algorithmic primitives, like Trotterisation and parameter optimisation functionality for the quantum approximate optimisation algorithm (QAOA), or variational quantum eigensolver (VQE). Furthermore, SimuQ [36], and QuTiP [37] support extended functionality for analysing quantum computers at pulse-level. Here, SimuQ focuses on Hamiltonian simulation, and QuTiP provides several algorithms for QOC.

However, customisation and effort are required to utilise quantum circuits defined in one of those frameworks in a QML setting.

The second category contains more QML-tailored approaches, including frameworks like PennyLane [38], TensorFlow Quantum [39], or Qibo [40] with its recent QML-library [41]. They provide a comprehensive toolset that extends the capabilities of the quantum circuit simulation or hardware interface with, for instance, quantum and classical ML datasets, different gradient computation methods, or a native integration into classical ML frameworks, like PyTorch [42], or TensorFlow [43]. The recent approach AutoQML [44] bundles the functionality of these QML frameworks to generate automated pipelines for non-domain experts and industrial use cases, as a more specialised method compared to, for instance, Refs. [45, 46], which orchestrate general quantum experiments. The two frameworks, PennyLane and Qibo, which are most similar to our work, additionally integrate differentiable pulse-level control within a QML workflow, enabling a full-stack approach from application to hardware or simulator execution. Our work differs from these two in that we particularly focus on the analysis of the interplay between QFMs and their properties.

Since it has been established that PQCs can be represented as truncated Fourier series [8, 47] (*cf.* [Sec. III](#)), the Fourier spectrum poses an important tool to determine the expressivity of a QML model. From a fundamental research perspective, some relations between other properties of quantum circuits and their Fourier spectrum are known, such as the expressibility [48, 49] (*i.e.*, the ability of a model to explore the Hilbert space; *cf.* [Sec. VI-B](#)), or dequantisability [50, 51] (*i.e.*, the ability to find classical alternatives to represent a quantum model). Furthermore, a variety of publications [14, 15, 30, 52–56] investigate the Fourier spectrum to improve QML algorithms, for instance, to find suitable circuit structures or data encodings [57]. However, if code accompanies these kinds of research articles, it mainly consists of individual code setups, which may allow reproducing individual numerical experiments, yet are difficult to build upon. Therefore, these software packages fall into our third category of QML software.

The QML-ESSENTIALS package, introduced in Ref. [29], aims to close the gap between the categories (2) and (3): It provides a set of standard tools, less general than category (2) frameworks, but tailored to QFMs investigations and suited as a backend for category (3) numerical experiments. However, we find that the unitary-centred approach in Ref. [29] masks the possibilities of lower-level quantum computations in QML to directly access and learn the pulses of an operation. With our QOC implementation, we aim to create a differentiable interface for gradient-based pulse optimisation, such as algorithms suggested in Refs. [58–60], or more recent approaches that, for instance, use reinforcement learning [23].

Additionally, we propose further improvements to QML-ESSENTIALS [29] to mature them into a software framework that comprehensively tackles QFMs.

All performance-critical computations (including all quantum computational simulations) are implemented using the JAX [32] library, and a variety of other features (cf. Sec. IV).

III. PRELIMINARIES ON QUANTUM FOURIER MODELS

The goal of QML aligns with classical machine learning: *learning* a function f using a function approximator \tilde{f} parametrised by θ . The parameters θ are *trained* to minimise the difference between $f(x)$ and $\tilde{f}(x, \theta)$ according to some metric, for a D -dimensional input $x = (x_1, \dots, x_D) \in \mathbb{R}^D$, using classical optimisation routines, such as gradient descent [1].

In QML, the function approximator is an n -qubit quantum circuit whose expectation value, using the observable \mathcal{M} , is given by

$$\tilde{f}(x, \theta) = \langle 0 |^{\otimes n} \hat{U}^\dagger(x, \theta) \mathcal{M} \hat{U}(x, \theta) | 0 \rangle^{\otimes n}, \quad (1)$$

parametrised by the trainable parameter vector $\theta = (\theta_1, \dots, \theta_{L+1}) \in [\mathbb{R}^K]^{L+1}$ with L layers of K parameters per layer¹.

We build the eventual unitary $\hat{U}(x, \theta)$ as products of multiple layers ($\ell \in [1, L]$) of consecutive encoding unitaries $\hat{S}^{(\ell)}(x)$ and trainable unitaries $\hat{W}^{(\ell)} := \hat{W}^{(\ell)}(\theta_\ell)$ as follows:

$$\hat{U}(x) = \hat{W}^{(L+1)} \hat{S}^{(L)}(x) \hat{W}^{(L)} \dots \hat{W}^{(2)} \hat{S}^{(1)}(x) \hat{W}^{(1)} \quad (2)$$

where an additional last trainable layer $\hat{W}^{(L+1)}$ is added. To allow for an arbitrary circuit structure and placement of unitary and encoding gates, we construct $\hat{S}^{(\ell)}(x)$ to comprise combinations of D unitaries $\hat{S}_i^{(\ell)}(x)$, each of which encodes one single input feature x_i :

$$\hat{S}_i^{(\ell)}(x) = \exp(-i e_i^T x \hat{G}_i^{(\ell)}) = \exp(-i x_i \hat{G}_i^{(\ell)}) \quad (3)$$

with Hermitian generator $\hat{G}_i^{(\ell)}$ and standard basis vector e_i .

As established in Refs. [8, 9], this architecture allows rewriting Eq. 1 as a partial Fourier series

$$\tilde{f}(x, \theta) = \sum_{\omega \in \Omega} c_\omega(\theta) e^{i\omega^T x}, \quad (4)$$

where Ω contains the unique frequencies resulting from the eigenvalues of each generating encoding Hamiltonian $\hat{G}_i^{(\ell)}$. The corresponding complex Fourier coefficients are given by $\{c_\omega(\theta)\}_{\omega \in \Omega}$.

From Eq. 4 it can be inferred that the Fourier frequencies derive from the encoding gates (*i.e.*, the feature map), while the Fourier coefficients, parametrised by the parameters θ of the circuit, solely depend on the circuit structure, which we refer to as *ansatz* in the further course of this work. In Sec. VI-A, we describe our implemented metrics for circuit analysis that derive from the Fourier spectrum and how we compute coefficients in the framework.

¹Without loss of generality, we assume identical numbers of parametrised gates per layer.

TABLE I: Python module overview.

Module	Description
model.py ⁽⁺⁾	Data-reuploading model class with various options to parameter initialisation, encoding schemes, measurement type and noise.
ansaeetze.py ⁽⁺⁾	Predefined ansätze and encodings to be used within a model.
gates.py ⁽⁺⁾ , unitary.py, pulses.py ⁺	Collection of (noisy) gates implemented as both unitary and pulse variant.
topologies.py ⁺	Collection of circuit topologies.
qoc.py ⁺	Tools for QOC and pulse-level based optimisation (cf. Sec. V).
coefficients.py ⁽⁺⁾	Calculation of Fourier coefficients and frequencies (FFT or analytical), as well as the FCC. Also includes a Fourier series dataset generator.
expressibility.py	Calculation of expressibility (cf. Sec. VI-B for implemented metrics).
entanglement.py ⁽⁺⁾	Calculation of entangling capability (cf. Sec. VI-C for implemented metrics).
tape.py ⁺ , script.py ⁺ , jaqsi.py ⁺	JAX-based quantum circuit simulator.
operations.py ⁺ , math.py ⁽⁺⁾ , utils.py ⁽⁺⁾ , drawing.py ⁺	Collection of utilities in the framework for <i>e.g.</i> quantum operations, mathematical tools, or drawing.

IV. PACKAGE OVERVIEW

An overview of the different components included in QML-ESSENTIALS and a code example are shown in Fig. 1. At a more technical level, the Python modules of the package are summarised in Tab. I. In the following, we give a more detailed description of the modules and components utilised in this work, while a full API documentation is delivered with the software package. The source code is hosted on GitHub², linked to Zenodo [62], and distributed via PyPI³.

A. Model and Ansatz Overview

The Model forms the core of the package, built upon a given ansatz. An ansatz can be imported from the ansatz-library, which contains, among others, a *strongly-entangling ansatz* (SEA) [47] (shown as an example at the top of Fig. 1), a *hardware-efficient ansatz* (HEA) [63] and almost all ansätze introduced in Ref. [48]. Alternatively, we refactored custom ansatz definitions to allow composition via multiple Block instances, as shown on the right-hand side of Fig. 1. A Block takes a set of quantum gates and a topology, which describes the order and layout of two-qubit operations in a circuit.

The resulting ansatz represents one variational layer of the model of size `n_qubits`, which is repeated multiple times, depending on the `n_layer` argument of the Model constructor, to build the full circuit. For QML use cases, the unitary ansatz comprises multiple rotational gates (*e.g.*, \hat{R}_X , \hat{R}_Y , \hat{R}_Z), parametrised by at least one angle.

²<https://github.com/cirKITers/qml-essentials>

³<https://pypi.org/project/qml-essentials/>

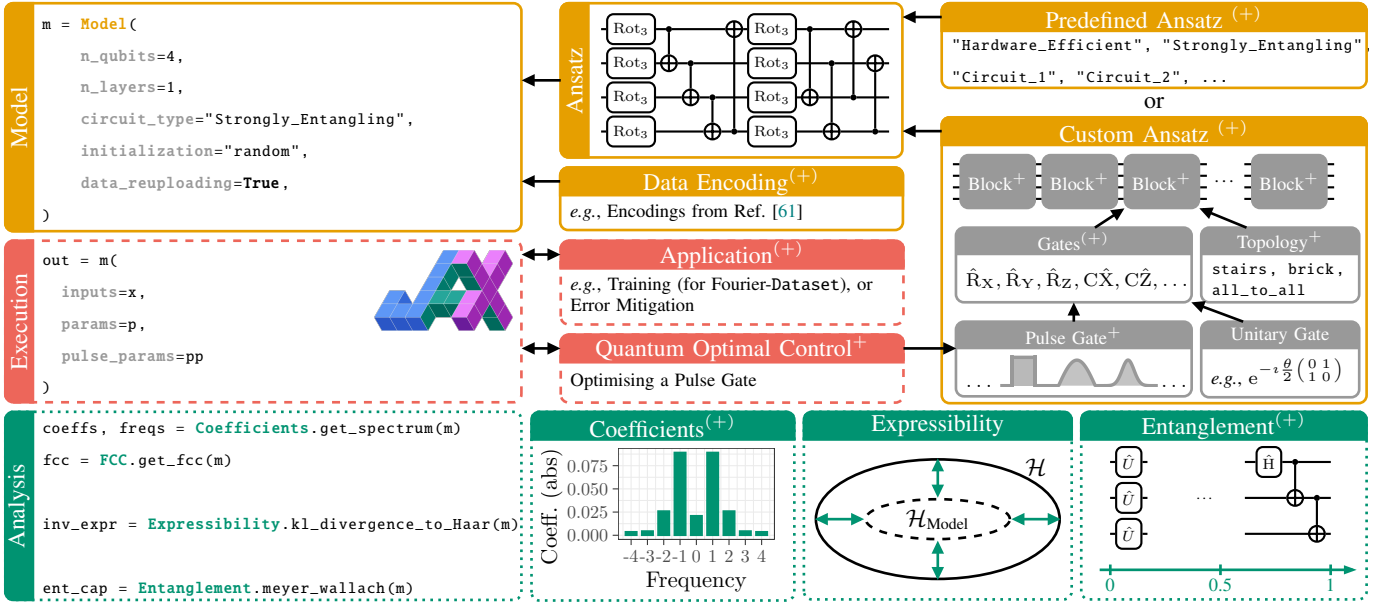


Fig. 1: Overview of our contributions to QML-ESSENTIALS with $+$ indicating novel additions, and $(+)$ major changes. The code example in Python on the left-hand side describes the main functionality of the framework. The **solid yellow** blocks on the right hand show the possibilities for defining a quantum Fourier **model**, based on different **ansätze** and **encodings**. **Dashed red** blocks show the possibilities of utilising the **execution** of a model, which encompasses the JAX-based [32] classical simulation of a quantum circuit and its measurement. Different input, parameter and pulse parameter values open up possibilities for a variety of **applications** and **QOC**. **Dotted green** blocks correspond to built-in metrics for analysing the model, including computing the **frequencies** and **coefficients** and **FCC** of the corresponding Fourier spectrum, as well as different metrics for **expressibility** and the **entanglement** capability.

We refer to the set of angles corresponding to all rotational gates in the circuit as the *parameters* of the model, which may be subject to optimisation in a training process. For the initialisation of the parameters, we provide multiple approaches, such as setting the parameters to specific values (zero or π), or uniformly at random in a specific domain.

Following the data re-uploading mechanism from Eq. 3, we support input encodings interleaved with the variational layers, where encoding positions can be controlled with the `data_reuploading` argument of the `Model` constructor. While the default input encoding scheme is an angle encoding (using a \hat{R}_X gate), we also implemented the encoding schemes of Ref. [61]: As introduced in Sec. III, the number of frequencies (and therefore the number of coefficients) depends on the chosen feature map. Hence, we provide an interface to choose between different encoding methods in addition to providing a custom feature map: (1) Hamming encoding (default, linear spectrum), (2) binary encoding (exponential spectrum), and (3) ternary encoding (exponential spectrum). In addition, trainable parameters can be applied in the feature map as introduced in Ref. [54].

Additionally, we allow to view ansätze from a lower abstraction level than unitary gates, by providing the feature to parametrise an ansatz directly with pulses, using *pulse parameters* (cf. Sec. V), instead of rotational angles.

B. Execution with Input- and Parameter-Batching

Depending on the application or metric that is to be evaluated for a model, different modes of execution may be required. In QML-ESSENTIALS, this mode can be selected by setting the `execution_type` argument when calling the model. In particular, this value can be set to (1) "expval", which gives the expectation value of an observable (e.g., the default is \hat{Z} , acting on a specified set of qubits), (2) "density", which gives the density matrix, (3) "state", which gives the state-vector representation, and (4) "probs", which gives the output probability distribution over a specified set of qubits, using a finite number of shots.

To evaluate models with concurrent sets of inputs and (pulse) parameters, we handle batch dimensions internally to cover all possible combinations.

We utilise our own simulator to carry out the computation of the circuit with the specified observables. This simulator and all implementations of the metrics presented in this work are entirely built upon the JAX [32] library for high-performance array computations and utilisation of parallelisation where possible. To solve the differential equation required for pulse-level simulation, our framework supports adaptive Runge-Kutta methods (e.g., Dormand-Prince of order 8 [64] via `DiffraX` [65]) as well as commutator-free Magnus integrators tailored for highly oscillatory Hamiltonian dynamics.

Our simulator is continuously tested against the PennyLane [38] simulator to ensure correctness of the computations.

C. Other Features

The support for pulse-parameters and QOC is detailed in Sec. V. Furthermore, the QML-ESSENTIALS package offers circuit analysis metrics for computing the Fourier spectrum, expressibility, or entangling capability. In Sec. VI we describe further which metrics are part of the original framework and how we contribute towards improving the existing state. In addition, we added other notable features, which are summarised in this subsection.

1) *Dataset Generator*: To reduce the overhead required to run simple trainings of QFMs, we provide a Fourier series dataset generator. Given a model, its intrinsic frequencies are inferred based on the encoding strategy chosen upon initialisation. Afterwards, corresponding coefficients are uniformly sampled within a unit circle. The resulting Fourier series is then used to obtain pairs of data points (x and $\mathcal{F}(x)$), which can then be used in a training environment.

2) *Visualisation*: To obtain a visualisation for a model's circuit, our framework offers an export of circuit diagrams as text-based or Matplotlib figures. Additionally, we support exporting to the Quantikz \LaTeX package [66]. The resulting \LaTeX code can be easily copied into a paper or thesis. Pulse-sequence schedules can also be visualised via the same API.

V. PULSE-LEVEL SIMULATION

Generally, unitary gates are used to train QFM models. However, there exist lower-level parameters that allow controlling the exact pulse of each gate applied to a quantum system, to which we refer as *pulse parameters*. These parameters are usually subject to a fine-tuning process carried out by the hardware provider in a process commonly termed QOC. In our framework, we provide an interface to conduct experiments that require access to the pulse parameters, therefore enabling research that is more tailored to the actual quantum hardware.

A. Pulses of Basis Gates

Generally, the dynamics of a closed quantum system are described by the unitary time evolution of a state ψ . For many platforms (*e.g.*, superconducting transmons), dynamics from time t_0 to $t_0 + \delta t$ are governed by the time-dependent Schrödinger equation [40, 67] (assuming the convention $\hbar = 1$):

$$|\psi(t_0 + \delta t)\rangle = \mathcal{T} \exp\left(-i \int_{t_0}^{t_0 + \delta t} \hat{\mathcal{H}}(t, \mathbf{p}) dt\right) |\psi(t_0)\rangle, \quad (5)$$

where \mathcal{T} denotes the time-ordering operator, and $\hat{\mathcal{H}}(t, \mathbf{p})$ is the time-dependent Hamiltonian parametrised by pulse parameters \mathbf{p} (distinct from the Hadamard gate \hat{H}).

Typically, the Hamiltonian decomposes into a static operator $\hat{\mathcal{H}}_{\text{static}}$ (unperturbed energy levels) and m control operators $\{\hat{\mathcal{H}}_j\}_{j=1}^m$ driven by time-dependent signals $S_j(t; \mathbf{p})$, such that

$$\hat{\mathcal{H}}(t, \mathbf{p}) = \hat{\mathcal{H}}_{\text{static}} + \sum_{j=1}^m S_j(t; \mathbf{p}) \hat{\mathcal{H}}_j. \quad (6)$$

Throughout our work, we choose $\hat{\mathcal{H}}_{\text{static}} = \frac{\omega_q}{2} \hat{Z}$ for general periodically driven two-level systems [68], with the qubit frequency $\omega_q = 10\pi$ adapted as needed for a specific hardware. Notably, both the exact static Hamiltonian and control Hamiltonians depend on the specific hardware that is used [69, 70]. The control signals are typically composed of an envelope function $E_j(t; \mathbf{p})$ and a carrier wave: $S_j(t; \mathbf{p}) = E_j(t; \mathbf{p}) \cos(\omega_c t + \phi_c)$, where ω_c is the carrier frequency and ϕ_c is the carrier phase.

To simplify the evolution in Eq. 5, it is common to move to the interaction picture with respect to $\hat{\mathcal{H}}_{\text{static}}$, factoring out the free evolution. Our software framework supports the exact numerical integration of these full interaction-picture dynamics, either directly in the laboratory frame or via a product-to-sum drive frame decomposition that explicitly resolves slow and fast oscillatory modes. However, for the numerical experiments in this work, we primarily utilise the rotating-wave approximation (RWA), which drops the fast-oscillating counter-rotating terms. The RWA significantly reduces the integration stiffness and accelerates simulation while maintaining high accuracy for near-resonant drives.

Selecting the control envelopes $E_j(t; \mathbf{p})$ such that the resulting time evolution implements a target unitary operation is then the goal of QOC. To this end, we provide a set of commonly known pulse envelope shapes, listed in Tab. II.

TABLE II: Pulse envelope functions, parametrised by the time t and pulse parameters \mathbf{p} , consisting of combinations of Amplitude A , pulse width σ , the central pulse time t_c , and, in case of the derivative removal by adiabatic gate (DRAG) pulse, a scaling factor β .

Envelope	$E(t; \mathbf{p})$
Gaussian	$A \exp\left(-1/2 \left(\frac{t-t_c}{\sigma}\right)^2\right)$
Rectangle	$A \mathbf{1}(t - t_c \leq \sigma/2)$
Raised Cosine	$A \cos\left(\pi \frac{t-t_c}{\sigma}\right) \quad \forall t - t_c \leq \sigma/2$
DRAG [71]	$A \exp\left(-1/2 \left(\frac{t-t_c}{\sigma}\right)^2\right) \left[1 - i\beta \frac{t-t_c}{\sigma^2}\right]$
Hyperbolic Secant	$A \left(\cosh\left(\frac{t-t_c}{\sigma}\right)\right)^{-1}$

Given the envelope and parameters subject to QOC (*cf.* Sec. V-C), a pulse sequence diagram can be used to visualise the envelope, carrier frequency and duration of the pulses corresponding to the particular gates of a given circuit. An exemplary diagram is shown in Fig. 2 where shaded areas represent the envelope and dashed vertical lines fence the duration of the pulse, in between which the carrier frequency is rendered. Note that in case of the \hat{R}_Z and $\hat{C}\hat{Z}$ gates, no actual pulse is applied; instead, the system evolves freely to rotate the phase accordingly.

B. Pulse-Gate Interface

Carefully optimising the fidelities of a set of basis gates allows building composed gates upon this foundation.

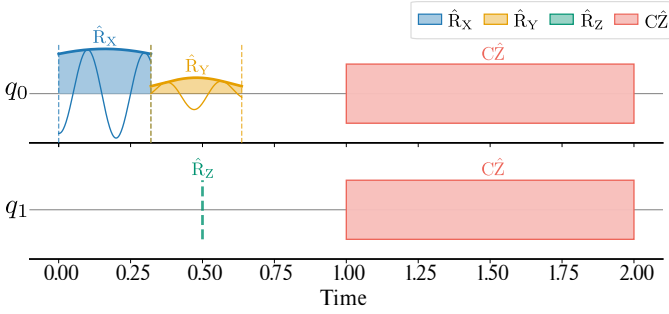


Fig. 2: Pulse sequence diagram showing an \hat{R}_X and \hat{R}_Y gate on the first-, a \hat{R}_Z on the second, and a \hat{CZ} gate on both qubits. Shaded areas show envelopes with carrier frequencies marked by thin lines. Vertical lines delimit the gate duration.

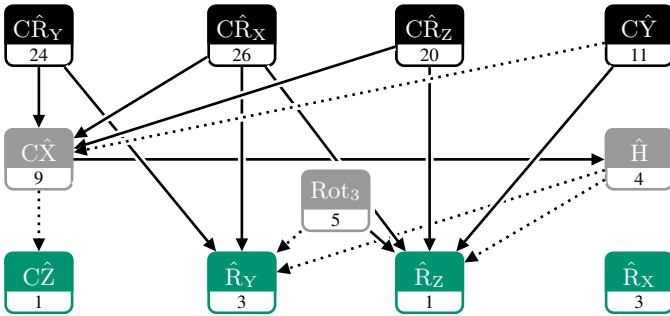


Fig. 3: Overview of the connectivity between basis gates and composed gates. The line type (dotted/solid) represents the number of child gates (1/2) the parent gate depends on. Labels at the bottom of each node represent the total number of pulse parameters including redundancies.

In the context of this work, we choose the universal gate set $\{\hat{R}_X, \hat{R}_Y, \hat{R}_Z, \hat{CZ}\}$ as our basis gates, on which we build other single- and two-qubit gates. While these basis gates are directly described by the pulses introduced in Sec. V-A, Fig. 3 depicts the corresponding dependence of higher-level gates. Here, the number below each node represents the overall number of pulse parameters available to this specific gate. As an example, the $\hat{C}\hat{R}_Y$ gate is composed of two \hat{R}_Y gates and two $\hat{C}\hat{X}$ gates which in turn consist of two \hat{H} and one $\hat{C}\hat{Z}$ gate. There may be several repetitions of a basis gate within a composed gate, which can introduce redundant parameters. Our pulse-level interface, shown in Fig. 3, uses a graph-based structure that allows controlling either the pulse parameters of the basis gates or all the parameters, including redundant ones from the repeated basis gates. This approach enables more specialised QOC routine (cf. Sec. V-C) and potentially enables increasing the fidelity of composed gates.

C. Quantum Optimal Control

We provide a QOC routine to learn the pulse parameters of a pulse-based gate. In QOC, overviewed in Fig. 4, we optimise a multi-objective cost function using Adam [72] with

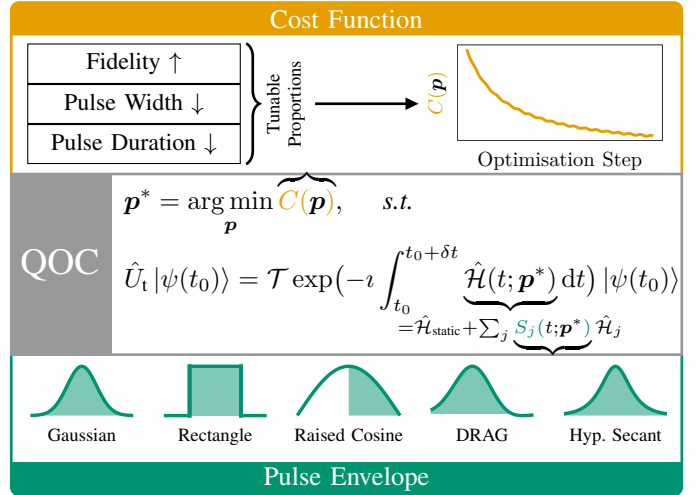


Fig. 4: Overview of our QOC procedure to find optimal pulse parameters \mathbf{p}^* by minimising some cost function C . Resulting pulse parameters determine, for instance, the amplitude, width or duration of the pulse envelope, for which the available possibilities are shown at the bottom. For an ideal optimisation, the state after evolution with the constructed pulse gate matches that of the target unitary gate \hat{U}_t .

weight decay regularisation and a cosine decay learning rate scheduler [73] with a warm-up phase. This optimisation is embedded into a multi-stage procedure: In the first stage, a grid scan is performed to evaluate parameter candidates, each refined through a few gradient steps. The best-performing candidate is then used to initialise the second stage, which consists of multiple independent optimisation restarts. The first restart begins from the selected candidate, while subsequent restarts apply Gaussian-distributed random perturbations to the parameter vector.

Maximising the fidelity $|\langle \psi | \phi \rangle|^2$ (absolute and phase difference) between the standard unitary state $|\psi\rangle$ and the pulse-based state $|\phi\rangle$ is the primary objective. Furthermore, minimising pulse duration keeps the sequence as short as possible, which is advantageous due to T1 and T2 coherence time constraints. In addition to these objectives, our QOC routine can be extended with custom cost functions accounting for factors like Gaussian pulse width or spectral density. The state after applying a given pulse sequence can typically only be obtained by measuring multiple shots of a quantum system in an actual hardware environment. However, as a proof of concept, we compute the pulse-based state by numerically simulating the pulse sequences and compare it to the standard unitary state.

We conducted an optimisation run on the basis gate set described in Sec. V-B. For the optimisation, we use a mixed-objective cost function consisting of the three parts shown at the top of Fig. 4, where we weighted the pulse width and duration with 5 and 15 ($\times 10^{-9}$), respectively.

The absolute and phase differences are weighted equally in this optimisation with the remainder to sum up to one (*i.e.*, $1 - 2 \times 10^{-8}$). For the rotational basis gates \hat{R}_X , \hat{R}_Y and \hat{R}_Z we calculate the average fidelity over 20 different rotation angles uniformly distributed in $[0, 2\pi]$. The $C\hat{Z}$ entangling gate is evaluated in combination with leading \hat{R}_Y and \hat{H} unitary gates applied on the control and target qubits, respectively, where the \hat{R}_Y gate is swept similarly to the single-qubit optimisation.

Tab. III lists mean results for the absolute ($\Delta|\cdot|$) and phase ($\Delta\angle\cdot$) differences of the basis gates after an optimisation over the parameter samples. As the optimisation also minimises the pulse width, which contradicts the fidelity, the results converge to exactly this weighting between the two objectives.

TABLE III: Mean gate infidelity (with respective standard deviations) over the variational parameter domain for all of the optimised pulse gates in QOC, compared to unitary gates.

Gate	\hat{R}_X	\hat{R}_Y	\hat{R}_Z	$C\hat{Z}$
$\Delta \cdot $	$(1.6 \pm 1.2) \times 10^{-13}$	$(3.6 \pm 2.9) \times 10^{-14}$	$(6.3 \pm 10) \times 10^{-17}$	$(1.6 \pm 1.2) \times 10^{-8}$
$\Delta\angle\cdot$	$(0.0 \pm 0) \times 10^{-13}$	$(0.0 \pm 0) \times 10^{-14}$	$(0.0 \pm 0) \times 10^{-17}$	$(17 \pm 6.1) \times 10^{-5}$

While the infidelity might appear sufficiently small at first sight, it is important to note that the error accumulates with the size of the circuit. We demonstrate this effect in Fig. 5, where we show the infidelity (average of absolute and phase differences) for all available ansätze with circuit depths up to twelve variational layers. Comparing the results with the circuit structure, it becomes apparent that the number of non-basis gates in a circuit correlates with the infidelity. Notably, even the highest infidelity in Fig. 5 is orders of magnitude below practically achieved infidelities on real devices, which currently record 10^{-7} [74] for single qubit gate fidelities (*cf.* Tab. III). Our QOC procedure is therefore sufficiently precise in ideal and non-noisy environments.

VI. METRICS

In this section, we describe the analysis metrics shown at the bottom of Fig. 1, discuss the corresponding implementation details, and show some proof-of-concept results for each metric.

A. Fourier Spectral Properties

Calculating the Fourier spectrum of QFMs is a central tool for analysing the capabilities of different ansätze and data encodings [15, 30, 49, 51, 55] (*cf.* Sec. III). Therefore, computing Fourier coefficients, frequencies and other metrics that derive from them are a key feature of the QML-ESSENTIALS package, which we describe in the following.

1) *Coefficients*: Fourier coefficients can be obtained with two different approaches in the QML-ESSENTIALS package: Either an analytical expression for the coefficients can be derived for a model, or the FFT is used.

For the former, the analytical method uses an algorithm proposed by Nemkov et al. [53], extended in Ref. [55].

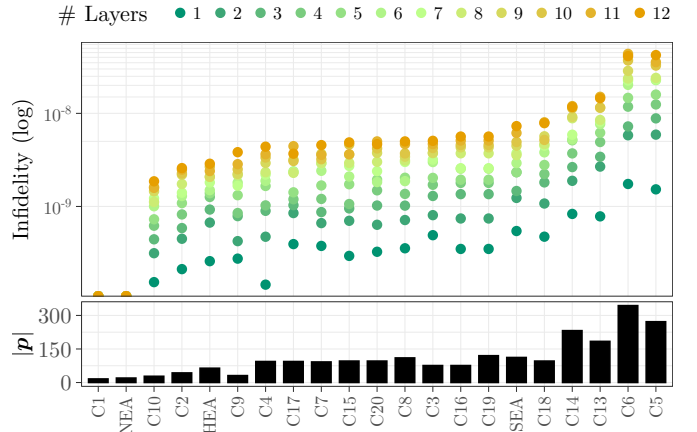


Fig. 5: (top) Infidelities for all available ansätze, ordered by the mean infidelity over all numbers of layers. (bottom) the number of pulse parameters for each ansatz. Ansätze of the format C* are taken from Ref. [48]. The additional ansätze, NEA, SEA and HEA refer to a *non-entangling*, *strongly-entangling* and *hardware-efficient* ansatz, respectively. Note that the infidelity is zero for all layers of C1 and NEA.

Here, the expectation value from Eq. 1 is expanded in terms of trigonometric polynomials. However, in its current state, this method is comparatively slow and only implemented for a single input feature, serving as an evaluation of the second, FFT-based method. In this second method, the FFT over a range of input samples yields a numerical approximation of the coefficients based on the expectation value in Eq. 1, given the number of frequencies for a model, for which an upper bound can be derived from the input encoding [61].

2) *Fourier Coefficient Correlation*: The FCC is a metric to quantify the correlation between the intrinsic Fourier coefficients of a QFM [30]. This is computed numerically by first executing the FFT to obtain a set of Fourier coefficients while sampling $\theta \in \Theta$ and then calculating the Pearson correlation coefficient [75]

$$r_{\Theta}(\omega, \omega') = \frac{\sum_{\theta \in \Theta} (c_{\omega}(\theta) - \bar{c}_{\omega})(c_{\omega'}(\theta) - \bar{c}_{\omega'})}{\sqrt{\sum_{\theta \in \Theta} (c_{\omega}(\theta) - \bar{c}_{\omega})^2 \sum_{\theta \in \Theta} (c_{\omega'}(\theta) - \bar{c}_{\omega'})^2}} \quad (7)$$

between each pair of Fourier coefficients. Here, \bar{c}_{ω} and $\bar{c}_{\omega'}$ represent the average value of the coefficient at frequency ω and ω' , respectively. This pairwise correlation is termed Fourier fingerprint, and averaging over all pairs results in the FCC

$$\text{FCC}_{\Theta} := \frac{1}{|\Omega|} \sum_{\omega, \omega' \in \Omega} |r_{\Theta}(\omega, \omega')|. \quad (8)$$

Our contribution covers the implementation of both, the Fourier fingerprint and the FCC with the default parameters set to replicate the results from Ref. [30].

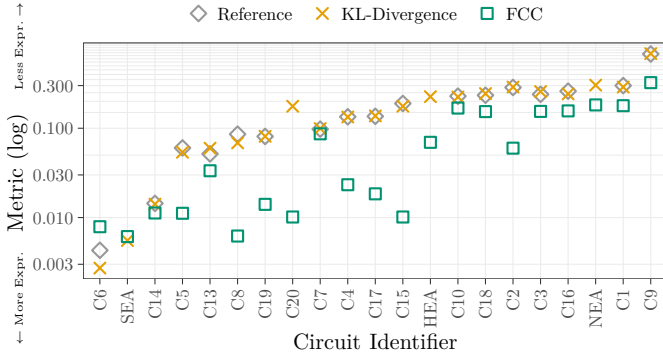


Fig. 6: Inverse of the expressibility, that is the KL-divergence to Haar states and the FCC (cf. Apx. A and Sec. VI-A2, respectively), on all predefined ansätze, which are ordered by the mean across all measures. The reference measures are taken from Ref. [48] and circuit identifiers are described in the caption of Fig. 5.

B. Expressibility

In QML-ESSENTIALS, the expressibility is computed by taking the inverse of the Kullback-Leibler (KL)-divergence between states sampled from a given ansatz and the Haar measure (cf. Apx. A for details). Therefore, the expressibility metric introduced in Ref. [48] evaluates how well an ansatz can approximate random states in Hilbert space. However, expressibility does not reflect whether circuits are suitable for learning settings, as functional constraints or input encodings are not considered in this approach. As shown in Ref. [30], the FCC addresses this limitation by revealing coefficient dependencies that limit the effective number of learnable frequency components.

The implementation of the expressibility based on the KL-divergence from QML-ESSENTIALS and our complementary FCC implementation allows us to measure and compare both metrics for all available ansätze. In particular, we compute the expressibility for 10 000 sample pairs and the FCC for an \hat{R}_Y angle encoding and 10 000 samples of uniformly randomly initialised parameters for each four-qubit ansatz. A comparison between the KL-divergence-based expressibility and FCC is shown in Fig. 6, also including the reference results for the expressibility from Ref. [48]. As shown in the figure, the KL-divergence from QML-ESSENTIALS and Ref. [48] match closely, while the FCC yields rather different values for most of the ansätze. This indicates that while some ansätze do not seem to be very expressive (e.g., C8, C15, or C20), the actual trainability might be better, as the Fourier coefficients are not highly correlated, and vice versa, thus confirming the discrepancy between these metrics as shown in [30].

To further provide a brief benchmark of our implementation, we measured the runtime required to compute expressibility and FCC for an increasing number of qubits. To this end, we evaluated both metrics for 50 parameter samples (-pairs) on all available ansätze with one and three variational layers on the same machine equipped with AMD EPYC 9454 processors.

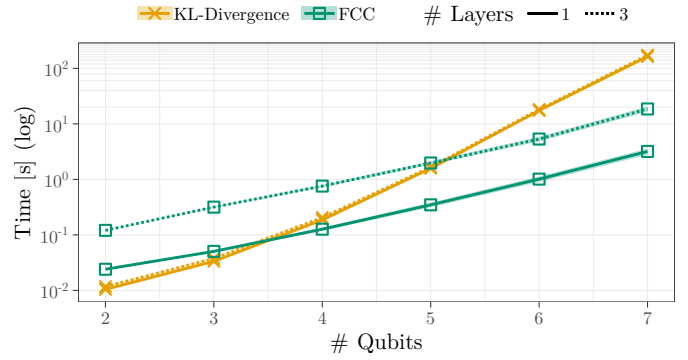


Fig. 7: Time required to compute the KL-divergence based expressibility and FCC for different circuit sizes, using our JAX-based simulator, averaged over ten executions for each configuration. The lines and points refer to the mean, and the shaded areas to the standard error of the mean over all available ansätze.

The results of these benchmark runs, which are shown in Fig. 7, demonstrate that the runtime increases exponentially with the number of qubits for both metrics, which is consistent with the exponential growth of the simulated quantum system. However, steeper curves for the expressibility computation indicate a less favourable scaling behaviour than for the FCC. This aligns with the complexity discussions of Ref. [30], where expressibility requires extensive sampling and pairwise fidelity evaluations, while the FCC avoids these quadratic comparisons. Notably, the time for computing expressibility is largely unaffected by the number of layers, while it becomes apparent that the FCC scales with the number of intrinsic frequencies of the model and therefore shows an additional computational overhead for the three- versus one-layer configuration. Overall, these results support the complexity analysis in Ref. [30] and also demonstrate the fast array computation capabilities of our JAX-based simulator.

C. Entangling Capability

The entangling capability aims to describe the amount of entanglement a circuit can create. It can also be understood as a measure of the *quantumness* of a particular circuit, as entanglement is one of the central properties that distinguish a quantum system from classical surrogates. As there are different ways of quantifying entanglement, there also exist various entangling capability metrics [76, 77], of which the Meyer-Wallach (MW) measure [78, 79] and Bell measurements (BMs) [80, 81] are included in QML-ESSENTIALS. The definitions and implementation details for these two metrics are summarised in Apx. B. In addition, we implement the EF [82] and CE [83] to also support noisy states.

1) *Entanglement of Formation*: Following Ref. [77], the EF [82] is usually defined as

$$E_{\text{EF}}(|\psi\rangle) = \inf \left\{ \sum_i p_i E(|\psi_i\rangle\langle\psi_i|) : \rho = \sum_i p_i |\psi_i\rangle\langle\psi_i| \right\}, \quad (9)$$

and represents the averaged entanglement of a pure state decomposition of a density matrix ρ , where the minimum over all possible decompositions then results in the EF. The entanglement for pure states is calculated using the MW measure as introduced in Eq. 13. Finding a pure state decomposition, however, is a non-trivial task. In our implementation, we utilise NumPy’s [84] eigenvalue and eigenstate decomposition of the density matrix ρ . Therefore, this metric is not applicable when using real devices. Based on the resulting decomposition, we proceed with calculating the entanglement of each eigenstate while weighting it by its eigenvalue as described in the left part of the right-hand side of Eq. 9. However, the eigenvalue decomposition is not unique, resulting in several potential decompositions, which can lead to different values for the entangling capability. Hence, the resulting values of the EF represent an upper bound.

2) *Concentratable Entanglement*: CE is a new family of multipartite entanglement measures introduced by Beckey et al. in [83]. Let $|\psi\rangle \in \mathbb{H}$ be a pure state over n qubits labelled by the set $S = \{1, \dots, n\}$. For any non-empty subset $s \subseteq S$ of qubit labels, the *Concentratable Entanglement* is defined as

$$\mathcal{C}_{|\psi\rangle}(s) = 1 - \frac{1}{2^{|s|}} \sum_{\alpha \in 2^s} \text{Tr}[\rho_\alpha^2], \quad (10)$$

where 2^s denotes the power set of s and ρ_α is the result of tracing out all qubits not in α . CEs can be efficiently computed on a quantum computer using an n -qubit parallelised SWAP test [83]. This method requires two copies of the state $|\psi\rangle$ and n additional ancillary qubits, resulting in $3n$ qubits in total. Beckey et al. show in [85] that CEs can be approximated from the results of Bell-basis measurements, hence using a circuit without additional ancillas. CEs, as described, are defined only for pure states. As in the case of the Entanglement of Formation, an entanglement measure for mixed states can be derived from CEs using a convex roof extension:

$$\mathcal{C}_\rho(s) = \inf \left\{ \sum_i p_i \mathcal{C}_{|\psi_i\rangle}(s) : \rho = \sum_i p_i |\psi_i\rangle \langle \psi_i| \right\}. \quad (11)$$

In general, computing convex roof extension measures presents a hard optimisation problem. Efficient lower and upper bounds for this value are proven by Beckey et al. [85] and Foulds et al. [86]. Experimentally, however, we observe that computation of the CEs using the n -qubit parallelised SWAP test still yields suitable results under the influence of lower levels of noise.

3) *Experimental Validation*: We compare our implementations against the existing measures in QML-ESSENTIALS and the results presented in Ref. [48]. Specifically, we evaluate 1000 samples of uniformly randomly initialised parameters for each measure at one layer of each predefined four-qubit ansatz. The measures can be separated into those that only work on pure, noiseless states (including the reference [48] and implemented MW measure), and the remaining measures that are also applicable to noisy states; for the latter, we additionally apply a 1% depolarising noise model.

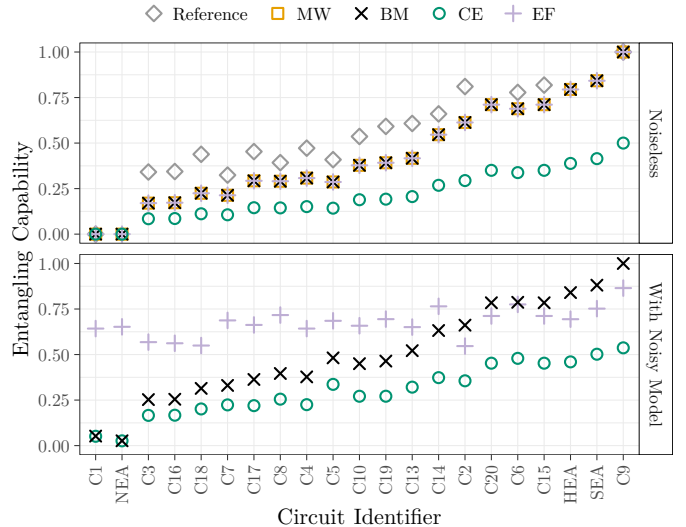


Fig. 8: Comparison of different entangling measures on all predefined ansätze, which are ordered by the mean entangling capability across all measures. The reference is taken from Ref. [48] and circuit identifiers are described in the caption of Fig. 5.

The results of this validation are shown in Fig. 8. While for pure states, the individual amount and scaling of entangling capabilities differ, the ansätze follow a consistent order across all measures. When applying the noise model, the entangling capability for BM and CE still follows this order, yet yields slightly higher values than in the noiseless case. In the noisy setting, the EF measure does not follow this increasing pattern in the entangling capability, but reaches higher values for the *lowly* entangled and lower values for the *highly* entangled ansätze, according to the other measures. However, as our EF implementation represents a valid upper bound for entangling capability and is based on the same pure state foundation as the BM, the EF might be a more reliable measure in cases where the EF is lower than the BM.

This comparison highlights that it can be useful to consider different metrics for differing comparison baselines. For instance, the BM or CE measures can be suitable for showing entangling capability differences across ansätze at one specific noise level. However, when conducting comparisons across different noise levels, it might be useful to incorporate entanglement measures such as the EF [15].

VII. CONCLUSION AND OUTLOOK

The software library introduced in this work provides a comprehensive environment for studying QML models from both a Fourier-analytic and hardware-aware perspective. By combining modular ansatz construction, pulse-level modelling, Fourier-spectral diagnostics, entanglement analysis, and a differentiable simulation backend, our software enables a reproducible workflow for a broad range of controlled numerical experiments within a single package.

This paves the way to investigate architectural ansatz choices and physical constraints in the context of QML. At the same time, we acknowledge that significant computational costs are associated with training QML models, particularly in pulse-level and noise-aware settings. While our framework incorporates performance optimisations to mitigate computational overheads, the primary contribution of this work is scientific rather than immediately application-driven.

In particular, we view the framework as a tool for the systematic and reproducible investigation of pulse-level learning with QFMs. While both perspectives have independently received considerable attention in the QML literature, with pulse-level approaches for improving hardware control and Fourier analysis for understanding the functional structure of circuits, to our best knowledge, our software is the first to allow for their combined investigation. With the foundation provided in this work, we leave experiments and analysis to connect the physical implementation of QML models with their spectral properties to future work.

Our framework further provides the basis for a variety of systematic studies. For instance, it can be investigated to a greater extent how specific data encoding strategies and ansätze, with their intrinsic entangling capability and expressibility, influence trainability and set quantum models apart from classical ML models. Additionally, the differentiable pulse-level backend enables experiments for the direct optimisation of pulse parameters in certain learning tasks, as well as the exploration of error mitigation strategies. Notably, our implementation allows combining pulse-level computation with any of the available noise models, therefore enabling a hardware-aware study of QML models. In future work, this may include adding hardware-aware system dynamics, such as damping due to hardware limitations. Ultimately, we aim to provide a comprehensive toolbox for reproducible QML research.

ACKNOWLEDGMENT

We thank Gabriel Mejía Ruiz for valuable discussions and comments to this work. MS, EK and AS acknowledge support by the state of Baden-Württemberg through bwHPC. LS acknowledges the support by the Doctoral School Karlsruhe School of Elementary and Astroparticle Physics: Science and Technology. MF and WM acknowledge support by German Research Foundation, grant MA 9739/1-1, the German Federal Ministry of Research, Technology and Space (BMFTR), funding program “Research Program Quantum Systems”, grant number 13N17387 and the High-Tech Agenda of the Free State of Bavaria.

APPENDIX A EXPRESSIBILITY METRIC

This section provides the definitions for the expressibility metrics implemented in QML-ESSENTIALS. The calculation is based on the KL-divergence between the distributions (1) obtained by sampling from the Haar integral

$\int_{\text{Haar}} d\psi (|\psi\rangle\langle\psi|)^{\otimes t}$ of a state t -design, and (2) obtained from the model $\int_{\theta} d\theta (|\psi_{\theta}\rangle\langle\psi_{\theta}|)^{\otimes t}$ [48, 87]:

$$D_{\text{KL}}(P_{\text{Model}}(F; \theta) \| P_{\text{Haar}}(F)). \quad (12)$$

Here, the fidelity $F := |\langle\psi_1|\psi_2\rangle|^2$ is based on the state overlap of pairs of sampled states $(|\psi_1\rangle, |\psi_2\rangle)$, and the distribution is then $P(F)$.

This metric yields zero if $P_{\text{Model}}(F; \theta) = P_{\text{Haar}}(F)$, meaning the states sampled from the QFM are Haar distributed. For the least expressive case, that is, the empty circuit $U = 1$, the KL divergence becomes $\ln(n_{\text{bins}})$ where n_{bins} describes the number of bins that are used for discretising the probability distribution using a histogram. Expressibility is thus regarded as the inverse KL-divergence.

APPENDIX B ENTANGLING CAPABILITY METRICS

This section provides the definitions for the implemented entangling capability metrics in QML-ESSENTIALS.

A. Meyer-Wallach Measure

The MW entangling capability [78, 79] of a state $|\psi\rangle$ is defined as

$$E(|\psi\rangle) = 2 \left(1 - \frac{1}{n} \sum_{k=0}^{n-1} \text{Tr} [\rho_k^2] \right), \quad (13)$$

based on partial density matrices ρ_k , where k denotes the subsystem (*i.e.*, qubit index).

This metric has the property that if $\text{Tr} [\rho_j^2] = 1 \quad \forall j$, implying $E = 0$, $|\psi\rangle$ is a product state whereas $E = 1$ iff $\text{Tr} [\rho_k^2] = 1/2 \quad \forall k$ meaning the state is maximally mixed. Notably, this metric is restricted to pure states and cannot be used for mixed states as they occur in decoherent noisy circuits.

B. Bell Measurement

The BM [80, 81] provides an alternative way to compute $\text{Tr} [\rho_k^2]$ in Eq. 13. Here, a circuit is considered, where the state of interest is vertically prepared twice, thereby requiring n additional ancillary qubits, and an inverse Bell-state is applied onto each pair of qubits between the two subsystems.

As shown in Ref. [81], the squared trace of the density matrix is linearly dependent on the probability of measuring the parity between each of the qubits in the individual subsystems:

$$\text{Tr} [\rho_k^2] = 1 - 2 \cdot P_{\text{odd},k}, \quad (14)$$

where $P_{\text{odd},k}$ is the probability of odd, non-zero parity in the outcomes of the k th qubit on each copy. Inserting this in Eq. 13, yields the same estimate of the entangling capability as when using the MW metric, yet uses a physical observable measurement applicable to real quantum systems. However, when using mixed states, impurities from the interaction with the environment may overestimate the entangling capability [86]. Therefore, the resulting values of the BM for mixed states rather present an upper bound for the entangling capability.

REFERENCES

- [1] M. Schuld and F. Petruccione, *Supervised Learning with Quantum Computers* (Quantum Science and Technology), en. Cham: Springer International Publishing, 2018. DOI: [10.1007/978-3-319-96424-9](https://doi.org/10.1007/978-3-319-96424-9)
- [2] M. Franz, L. Wolf, M. Periyasamy, et al., “Uncovering instabilities in variational-quantum deep q-networks,” *Journal of The Franklin Institute*, Aug. 2022. DOI: [10.1016/j.jfranklin.2022.08.021](https://doi.org/10.1016/j.jfranklin.2022.08.021)
- [3] T. Yue, W. Mauerer, S. Ali, et al., “Challenges and opportunities in quantum software architecture,” in *Software Architecture: Research Roadmaps from the Community*, P. Pelliccione, R. Kazman, I. Weber, et al., Eds. Cham: Springer Nature Switzerland, 2023, pp. 1–23. DOI: [10.1007/978-3-031-36847-9_1](https://doi.org/10.1007/978-3-031-36847-9_1)
- [4] D. Heimann, G. Schönhoff, E. Mounzer, et al., *Learning capability of parametrized quantum circuits*, Mar. 2024. DOI: [10.48550/arXiv.2209.10345](https://doi.org/10.48550/arXiv.2209.10345) arXiv: [2209.10345](https://arxiv.org/abs/2209.10345) [quant-ph].
- [5] C. Carbonelli, M. Felderer, M. Jung, et al., “Challenges for quantum software engineering: An industrial application scenario perspective,” in *Quantum Software: Aspects of Theory and System Design*, I. Exman, R. Pérez-Castillo, M. Piattini, et al., Eds. Cham: Springer Nature Switzerland, 2024, pp. 311–335. DOI: [10.1007/978-3-031-64136-7_12](https://doi.org/10.1007/978-3-031-64136-7_12)
- [6] J. Eisert and J. Preskill, *Mind the gaps: The fraught road to quantum advantage*, 2025. DOI: [10.48550/arxiv.2510.19928](https://doi.org/10.48550/arxiv.2510.19928) arXiv: [2510.19928](https://arxiv.org/abs/2510.19928) [quant-ph].
- [7] H.-Y. Huang, M. Broughton, N. Eassa, et al., *Generative quantum advantage for classical and quantum problems*, 2025. DOI: [10.48550/arxiv.2509.09033](https://doi.org/10.48550/arxiv.2509.09033) arXiv: [2509.09033](https://arxiv.org/abs/2509.09033) [quant-ph].
- [8] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, et al., “Data re-uploading for a universal quantum classifier,” *Quantum*, vol. 4, p. 226, Feb. 2020. DOI: [10.22331/q-2020-02-06-226](https://doi.org/10.22331/q-2020-02-06-226) arXiv: [1907.02085](https://arxiv.org/abs/1907.02085).
- [9] M. Schuld, R. Sweke, and J. J. Meyer, “The effect of data encoding on the expressive power of variational quantum machine learning models,” *Physical Review A*, vol. 103, no. 3, p. 032430, Mar. 2021. DOI: [10.1103/PhysRevA.103.032430](https://doi.org/10.1103/PhysRevA.103.032430) arXiv: [2008.08605](https://arxiv.org/abs/2008.08605).
- [10] S. Wang, E. Fontana, M. Cerezo, et al., “Noise-Induced Barren Plateaus in Variational Quantum Algorithms,” *Nature Communications*, vol. 12, no. 1, p. 6961, Nov. 2021. DOI: [10.1038/s41467-021-27045-6](https://doi.org/10.1038/s41467-021-27045-6) arXiv: [2007.14384](https://arxiv.org/abs/2007.14384).
- [11] M. Larocca, S. Thanasilp, S. Wang, et al., “Barren plateaus in variational quantum computing,” *Nature Reviews Physics*, vol. 7, no. 4, pp. 174–189, Mar. 2025. DOI: [10.1038/s42254-025-00813-9](https://doi.org/10.1038/s42254-025-00813-9)
- [12] G. De Palma, M. Marvian, C. Rouzé, et al., “Limitations of variational quantum algorithms: A quantum optimal transport approach,” *PRX Quantum*, vol. 4, p. 010309, 1 Jan. 2023. DOI: [10.1103/PRXQuantum.4.010309](https://doi.org/10.1103/PRXQuantum.4.010309)
- [13] T. Ahmed, M. Kashif, A. Marchisio, et al., “A comparative analysis and noise robustness evaluation in quantum neural networks,” *Scientific Reports*, vol. 15, no. 1, Sep. 2025. DOI: [10.1038/s41598-025-17769-6](https://doi.org/10.1038/s41598-025-17769-6)
- [14] E. Fontana, I. Rungger, R. Duncan, et al., “Spectral analysis for noise diagnostics and filter-based digital error mitigation,” arXiv, Tech. Rep., Nov. 2022. DOI: [10.48550/arXiv.2206.08811](https://doi.org/10.48550/arXiv.2206.08811) arXiv: [2206.08811](https://arxiv.org/abs/2206.08811).
- [15] M. Franz, M. Strobl, L. Chaichenets, et al., *Out of tune: Demystifying noise-effects on quantum fourier models*, 2025. DOI: [10.48550/arXiv.2506.09527](https://doi.org/10.48550/arXiv.2506.09527) arXiv: [2506.09527](https://arxiv.org/abs/2506.09527) [quant-ph].
- [16] M. Cerezo, G. Verdon, H.-Y. Huang, et al., “Challenges and opportunities in quantum machine learning,” *Nature Computational Science*, vol. 2, no. 9, pp. 567–576, Sep. 2022. DOI: [10.1038/s43588-022-00311-3](https://doi.org/10.1038/s43588-022-00311-3)
- [17] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, et al., “Noisy intermediate-scale quantum algorithms,” *Rev. Mod. Phys.*, vol. 94, p. 015004, 1 Feb. 2022. DOI: [10.1103/RevModPhys.94.015004](https://doi.org/10.1103/RevModPhys.94.015004)
- [18] P. Krantz, M. Kjaergaard, F. Yan, et al., “A Quantum Engineer’s Guide to Superconducting Qubits,” *Applied Physics Reviews*, vol. 6, no. 2, p. 021318, Jun. 2019. DOI: [10.1063/1.5089550](https://doi.org/10.1063/1.5089550) arXiv: [1904.06560](https://arxiv.org/abs/1904.06560) [quant-ph].
- [19] O. R. Meitei, B. T. Gard, G. S. Barron, et al., *Gate-free state preparation for fast variational quantum eigensolver simulations: Ctrl-VQE*, May 2021. DOI: [10.48550/arXiv.2008.04302](https://doi.org/10.48550/arXiv.2008.04302) arXiv: [2008.04302](https://arxiv.org/abs/2008.04302) [quant-ph].
- [20] Z. Liang, H. Wang, J. Cheng, et al., “Variational quantum pulse learning,” in *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 2022, pp. 556–565. DOI: [10.1109/QCE53715.2022.00078](https://doi.org/10.1109/QCE53715.2022.00078)
- [21] A. Melo, N. Earnest-Noble, and F. Tacchino, “Pulse-efficient quantum machine learning,” *Quantum*, vol. 7, p. 1130, Oct. 2023. DOI: [10.22331/q-2023-10-09-1130](https://doi.org/10.22331/q-2023-10-09-1130)
- [22] H.-X. Tao and R.-B. Wu, “On the role of controllability in pulse-based quantum machine learning models,” in *2024 IEEE 63rd Conference on Decision and Control (CDC)*, 2024, pp. 13–18. DOI: [10.1109/CDC56724.2024.10886070](https://doi.org/10.1109/CDC56724.2024.10886070)
- [23] B. Sarma and M. J. Hartmann, “Designing fast quantum gates using optimal control with a reinforcement-learning ansatz,” *Phys. Rev. Appl.*, vol. 23, p. 014015, 1 Jan. 2025. DOI: [10.1103/PhysRevApplied.23.014015](https://doi.org/10.1103/PhysRevApplied.23.014015)
- [24] H.-X. Tao, X. Wang, and R.-B. Wu, *On the design of expressive and trainable pulse-based quantum machine learning models*, 2025. DOI: [10.48550/arxiv.2508.05559](https://doi.org/10.48550/arxiv.2508.05559) arXiv: [2508.05559](https://arxiv.org/abs/2508.05559) [quant-ph].
- [25] I. Henao, J. P. Santos, and R. Uzdin, “Adaptive quantum error mitigation using pulse-based inverse evolutions,” *npj Quantum Information*, vol. 9, no. 1, Nov. 2023. DOI: [10.1038/s41534-023-00785-7](https://doi.org/10.1038/s41534-023-00785-7)
- [26] F. Greiwe, T. Krüger, and W. Mauerer, “Effects of imperfections on quantum algorithms: A software engineering perspective,” in *2023 IEEE International Conference on Quantum Software (QSW)*, 2023, pp. 31–42. DOI: [10.1109/QSW59989.2023.00014](https://doi.org/10.1109/QSW59989.2023.00014)
- [27] S. Thelen, H. Safi, and W. Mauerer, “Approximating under the influence of quantum noise and compute power,” in *Proceedings of WHQP@IEEE QCE*, Sep. 2024. DOI: [10.1109/QCE60285.2024.10291](https://doi.org/10.1109/QCE60285.2024.10291) arXiv: [2408.02287](https://arxiv.org/abs/2408.02287).
- [28] S. R. Maschek, J. Schwittalla, M. Franz, et al., “Make some noise! measuring noise model quality in real-world quantum software,” en, in *Proceedings of the IEEE International Conference on Quantum Software (QSW)*, Jul. 2025. DOI: [10.48550/arxiv.2506.03636](https://doi.org/10.48550/arxiv.2506.03636) arXiv: [2506.03636](https://arxiv.org/abs/2506.03636).
- [29] M. Strobl, M. Franz, E. Kuehn, et al., “QML-essentials – a framework for working with quantum fourier models,” in *2025 IEEE International Conference on Quantum Software (QSW)*, 2025, pp. 238–243. DOI: [10.1109/QSW67625.2025.00035](https://doi.org/10.1109/QSW67625.2025.00035)
- [30] M. Strobl, M. E. Sahin, L. van der Horst, et al., *Fourier fingerprints of ansatzes in quantum machine learning*, 2025. DOI: [10.48550/arXiv.2508.20868](https://doi.org/10.48550/arXiv.2508.20868) arXiv: [2508.20868](https://arxiv.org/abs/2508.20868) [quant-ph].
- [31] W. Mauerer and S. Scherzinger, “1-2-3 reproducibility for quantum software experiments,” in *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2022, pp. 1247–1248. DOI: [10.1109/SANER53432.2022.00148](https://doi.org/10.1109/SANER53432.2022.00148)
- [32] J. Bradbury, R. Frostig, P. Hawkins, et al., *JAX: Composable transformations of Python+NumPy programs*, version 0.4.28, 2024.
- [33] A. Javadi-Abhari, M. Treinish, K. Krsulich, et al., *Quantum computing with Qiskit*, Jun. 2024. DOI: [10.48550/arXiv.2405.08810](https://doi.org/10.48550/arXiv.2405.08810) arXiv: [2405.08810](https://arxiv.org/abs/2405.08810) [quant-ph].
- [34] S. Sivarajah, S. Dilkes, A. Cowtan, et al., “Tket: A retargetable compiler for nisq devices,” *Quantum Science and Technology*, vol. 6, no. 1, p. 014003, Nov. 2020. DOI: [10.1088/2058-9565/ab8e92](https://doi.org/10.1088/2058-9565/ab8e92)
- [35] C. Developers, *Cirq*, Aug. 2025. DOI: [10.5281/ZENODO.4062499](https://doi.org/10.5281/ZENODO.4062499)
- [36] Y. Peng, J. Young, P. Liu, et al., “SimuQ: A framework for programming quantum hamiltonian simulation with analog compilation,” *Proc. ACM Program. Lang.*, vol. 8, no. POPL, Jan. 2024. DOI: [10.1145/3632923](https://doi.org/10.1145/3632923)
- [37] B. Li, S. Ahmed, S. Saraogi, et al., “Pulse-level noisy quantum circuits with QuTiP,” *Quantum*, vol. 6, p. 630, Jan. 2022. DOI: [10.22331/q-2022-01-24-630](https://doi.org/10.22331/q-2022-01-24-630)
- [38] V. Bergholm, J. Izaac, M. Schuld, et al., *PennyLane: Automatic differentiation of hybrid quantum-classical computations*, Jul. 2022. DOI: [10.48550/arXiv.1811.04968](https://doi.org/10.48550/arXiv.1811.04968) arXiv: [1811.04968](https://arxiv.org/abs/1811.04968) [quant-ph].
- [39] M. Broughton, G. Verdon, T. McCourt, et al., *Tensorflow quantum: A software framework for quantum machine learning*, 2021. DOI: [10.48550/arxiv.2003.02989](https://doi.org/10.48550/arxiv.2003.02989) arXiv: [2003.02989](https://arxiv.org/abs/2003.02989) [quant-ph].
- [40] S. Efthymiou, S. Ramos-Calderer, C. Bravo-Prieto, et al., “Qibo: A framework for quantum simulation with hardware acceleration,” *Quantum Science and Technology*, vol. 7, no. 1, p. 015018, Jan. 2022. DOI: [10.1088/2058-9565/ac39f5](https://doi.org/10.1088/2058-9565/ac39f5) arXiv: [2009.01845](https://arxiv.org/abs/2009.01845) [quant-ph].
- [41] M. Robbati, A. Papaluca, A. Pasquale, et al., *Qiboml: Towards the orchestration of quantum-classical machine learning*, 2025. DOI: [10.48550/arxiv.2510.11773](https://doi.org/10.48550/arxiv.2510.11773) arXiv: [2510.11773](https://arxiv.org/abs/2510.11773) [quant-ph].

- [42] A. Paszke, S. Gross, F. Massa, et al., *Pytorch: An imperative style, high-performance deep learning library*, 2019. DOI: [10.48550/arxiv.1912.01703](https://doi.org/10.48550/arxiv.1912.01703) arXiv: [1912.01703](https://arxiv.org/abs/1912.01703) [cs.LG].
- [43] Martín Abadi, Ashish Agarwal, Paul Barham, et al., *TensorFlow: Large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015.
- [44] M. Roth, D. A. Kreplin, D. Basilewitsch, et al., “Autoqml: A framework for automated quantum machine learning,” in *2025 IEEE International Conference on Quantum Software (QSW)*, 2025, pp. 81–91. DOI: [10.1109/QSW67625.2025.00019](https://doi.org/10.1109/QSW67625.2025.00019)
- [45] J. R. Fingar, P. Ross, L. Hölscher, et al., “Quark: A framework for quantum computing application benchmarking,” in *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 2022, pp. 226–237. DOI: [10.1109/QCE53715.2022.00042](https://doi.org/10.1109/QCE53715.2022.00042)
- [46] V. Gierisch and W. Mauerer, *Qef: Reproducible and exploratory quantum software experiments*, 2025. DOI: [10.48550/arxiv.2511.04563](https://doi.org/10.48550/arxiv.2511.04563) arXiv: [2511.04563](https://arxiv.org/abs/2511.04563) [quant-ph].
- [47] M. Schuld, A. Bocharov, K. Svore, et al., “Circuit-centric quantum classifiers,” *Physical Review A*, vol. 101, no. 3, p. 032308, Mar. 2020. DOI: [10.1103/PhysRevA.101.032308](https://doi.org/10.1103/PhysRevA.101.032308) arXiv: [1804.00633](https://arxiv.org/abs/1804.00633) [quant-ph].
- [48] S. Sim, P. D. Johnson, and A. Aspuru-Guzik, “Expressibility and Entangling Capability of Parameterized Quantum Circuits for Hybrid Quantum-Classical Algorithms,” en, *Advanced Quantum Technologies*, vol. 2, no. 12, p. 1900070, 2019. DOI: [10.1002/qute.201900070](https://doi.org/10.1002/qute.201900070)
- [49] H. Mhiri, L. Monbroussou, M. Herrero-Gonzalez, et al., *Constrained and Vanishing Expressivity of Quantum Fourier Models*, Mar. 2024. DOI: [10.48550/arxiv.2403.09417](https://doi.org/10.48550/arxiv.2403.09417) arXiv: [2403.09417](https://arxiv.org/abs/2403.09417).
- [50] F. J. Schreiber, J. Eisert, and J. J. Meyer, “Classical surrogates for quantum learning models,” *Phys. Rev. Lett.*, vol. 131, p. 100803, 10 Sep. 2023. DOI: [10.1103/PhysRevLett.131.100803](https://doi.org/10.1103/PhysRevLett.131.100803)
- [51] R. Sweke, E. Recio-Armengol, S. Jerbi, et al., “Potential and limitations of random Fourier features for dequantizing quantum machine learning,” *Quantum*, vol. 9, p. 1640, Feb. 2025. DOI: [10.22331/q-2025-02-20-1640](https://doi.org/10.22331/q-2025-02-20-1640) arXiv: [2309.11647](https://arxiv.org/abs/2309.11647) [quant-ph].
- [52] B. Casas and A. Cervera-Liarta, “Multidimensional fourier series with quantum circuits,” *Phys. Rev. A*, vol. 107, p. 062612, 6 Jun. 2023. DOI: [10.1103/PhysRevA.107.062612](https://doi.org/10.1103/PhysRevA.107.062612)
- [53] N. A. Nemkov, E. O. Kiktenko, and A. K. Fedorov, “Fourier expansion in variational quantum algorithms,” *Physical Review A*, vol. 108, no. 3, p. 032406, Sep. 2023. DOI: [10.1103/PhysRevA.108.032406](https://doi.org/10.1103/PhysRevA.108.032406) arXiv: [2304.03787](https://arxiv.org/abs/2304.03787) [quant-ph].
- [54] B. Jaderberg, A. A. Gentile, Y. A. Berrada, et al., “Let Quantum Neural Networks Choose Their Own Frequencies,” *Physical Review A*, vol. 109, no. 4, p. 042421, Apr. 2024. DOI: [10.1103/PhysRevA.109.042421](https://doi.org/10.1103/PhysRevA.109.042421) arXiv: [2309.03279](https://arxiv.org/abs/2309.03279) [quant-ph].
- [55] M. Wiedmann, M. Periyasamy, and D. D. Scherer, *Fourier Analysis of Variational Quantum Circuits for Supervised Learning*, Nov. 2024. DOI: [10.48550/arxiv.2411.03450](https://doi.org/10.48550/arxiv.2411.03450) arXiv: [2411.03450](https://arxiv.org/abs/2411.03450).
- [56] C. Duffy and M. Jastrzebski, *Spectral bias in variational quantum machine learning*, 2026. DOI: [10.48550/arxiv.2506.22555](https://doi.org/10.48550/arxiv.2506.22555) arXiv: [2506.22555](https://arxiv.org/abs/2506.22555) [quant-ph].
- [57] M. Gogeissl, H. Safi, and W. Mauerer, “Quantum data encoding patterns and their consequences,” in *Proceedings of the 1st Workshop on Quantum Computing and Quantum-Inspired Technology for Data-Intensive Systems and Applications*, ser. Q-Data '24, Santiago, AA, Chile: Association for Computing Machinery, 2024, pp. 27–37. DOI: [10.1145/3665225.3665446](https://doi.org/10.1145/3665225.3665446)
- [58] N. Khaneja, T. Reiss, C. Kehlet, et al., “Optimal control of coupled spin dynamics: Design of nmr pulse sequences by gradient ascent algorithms,” *Journal of Magnetic Resonance*, vol. 172, no. 2, pp. 296–305, 2005. DOI: [10.1016/j.jmr.2004.11.004](https://doi.org/10.1016/j.jmr.2004.11.004)
- [59] S. Machnes, E. Assémat, D. Tannor, et al., “Tunable, flexible, and efficient optimization of control pulses for practical qubits,” *Phys. Rev. Lett.*, vol. 120, p. 150401, 15 Apr. 2018. DOI: [10.1103/PhysRevLett.120.150401](https://doi.org/10.1103/PhysRevLett.120.150401)
- [60] O. V. Morzhin and A. N. Pechen, “Krotov method for optimal control of closed quantum systems,” *Russian Mathematical Surveys*, vol. 74, no. 5, p. 851, Oct. 2019. DOI: [10.1070/RM9835](https://doi.org/10.1070/RM9835)
- [61] E. Peters and M. Schuld, “Generalization despite overfitting in quantum machine learning models,” *Quantum*, vol. 7, p. 1210, Dec. 2023. DOI: [10.22331/q-2023-12-20-1210](https://doi.org/10.22331/q-2023-12-20-1210)
- [62] M. Franz and M. Strobl, *Qml-essentialsa framework for working with quantum fourier models*. DOI: [10.5281/zenodo.18503388](https://doi.org/10.5281/zenodo.18503388)
- [63] A. Kandala, A. Mezzacapo, K. Temme, et al., “Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets,” *Nature*, vol. 549, no. 7671, pp. 242–246, Sep. 2017. DOI: [10.1038/nature23879](https://doi.org/10.1038/nature23879)
- [64] J. R. Dormand and P. J. Prince, “A family of embedded Runge-Kutta formulae,” *Journal of Computational and Applied Mathematics*, vol. 6, no. 1, pp. 19–26, Mar. 1980. DOI: [10.1016/0771-050X\(80\)90013-3](https://doi.org/10.1016/0771-050X(80)90013-3)
- [65] P. Kidger, *On neural differential equations*, 2022. DOI: [10.48550/arxiv.2202.02435](https://doi.org/10.48550/arxiv.2202.02435) arXiv: [2202.02435](https://arxiv.org/abs/2202.02435) [cs.LG].
- [66] A. Kay, *Tutorial on the quantikz package*, 2023. DOI: [10.48550/arxiv.1809.03842](https://doi.org/10.48550/arxiv.1809.03842) arXiv: [1809.03842](https://arxiv.org/abs/1809.03842) [quant-ph].
- [67] T. Rothe Santos, “Evaluation of pulse level quantum fourier models,” 46.21.02; LK 01, Abschlussarbeit - Bachelor, Karlsruher Institut für Technologie (KIT), 2025. DOI: [10.5445/IR/1000184129](https://doi.org/10.5445/IR/1000184129)
- [68] Y. Wu and X. Yang, “Strong-Coupling Theory of Periodically Driven Two-Level Systems,” *Physical Review Letters*, vol. 98, no. 1, p. 013601, Jan. 2007. DOI: [10.1103/PhysRevLett.98.013601](https://doi.org/10.1103/PhysRevLett.98.013601)
- [69] T. Alexander, N. Kanazawa, D. J. Egger, et al., “Qiskit pulse: Programming quantum computers through the cloud with pulses,” *Quantum Science and Technology*, vol. 5, no. 4, p. 044006, Aug. 2020. DOI: [10.1088/2058-9565/aba404](https://doi.org/10.1088/2058-9565/aba404)
- [70] M. Votto, J. Zeiher, and B. Vermersch, “Universal quantum processors in spin systems via robust local pulse sequences,” *Quantum*, vol. 8, p. 1513, Oct. 2024. DOI: [10.22331/q-2024-10-29-1513](https://doi.org/10.22331/q-2024-10-29-1513)
- [71] F. Motzoi, J. M. Gambetta, P. Rebentrost, et al., “Simple pulses for elimination of leakage in weakly nonlinear qubits,” *Physical Review Letters*, vol. 103, no. 11, p. 110501, Sep. 2009. DOI: [10.1103/PhysRevLett.103.110501](https://doi.org/10.1103/PhysRevLett.103.110501) arXiv: [0901.0534](https://arxiv.org/abs/0901.0534) [cond-mat].
- [72] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2017. DOI: [10.48550/arxiv.1412.6980](https://doi.org/10.48550/arxiv.1412.6980) arXiv: [1412.6980](https://arxiv.org/abs/1412.6980) [cs.LG].
- [73] I. Loshchilov and F. Hutter, *Sgdr: Stochastic gradient descent with warm restarts*, 2017. DOI: [10.48550/arxiv.1608.03983](https://doi.org/10.48550/arxiv.1608.03983) arXiv: [1608.03983](https://arxiv.org/abs/1608.03983) [cs.LG].
- [74] M. C. Smith, A. D. Leu, K. Miyanishi, et al., “Single-qubit gates with errors at the 10^{-7} level,” *Phys. Rev. Lett.*, vol. 134, p. 230601, 23 Jun. 2025. DOI: [10.1103/PhysRevLett.134.230601](https://doi.org/10.1103/PhysRevLett.134.230601)
- [75] D. Freedman, R. Pisani, and R. Purves, “Statistics (international student edition),” *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*, 2007.
- [76] V. Vedral, M. B. Plenio, M. A. Rippin, et al., “Quantifying entanglement,” *Phys. Rev. Lett.*, vol. 78, pp. 2275–2279, 12 Mar. 1997. DOI: [10.1103/PhysRevLett.78.2275](https://doi.org/10.1103/PhysRevLett.78.2275)
- [77] M. B. Plenio and S. Virmani, “An introduction to entanglement measures,” *Quantum Info. Comput.*, vol. 7, no. 1, pp. 1–51, Jan. 2007.
- [78] D. A. Meyer and N. R. Wallach, “Global entanglement in multiparticle systems,” *Journal of Mathematical Physics*, vol. 43, no. 9, pp. 4273–4278, Sep. 2002. DOI: [10.1063/1.1497700](https://doi.org/10.1063/1.1497700) arXiv: [quant-ph/0108104](https://arxiv.org/abs/quant-ph/0108104) [quant-ph].
- [79] G. K. Brennen, “An observable measure of entanglement for pure states of multi-qubit systems,” arXiv, Tech. Rep., Nov. 2003. DOI: [10.48550/arxiv.quant-ph/0305094](https://doi.org/10.48550/arxiv.quant-ph/0305094) arXiv: [quant-ph/0305094](https://arxiv.org/abs/quant-ph/0305094) [quant-ph].
- [80] S. Foulds, V. Kendon, and T. Spiller, “The controlled SWAP test for determining quantum entanglement,” *Quantum Science and Technology*, vol. 6, no. 3, p. 035002, Jul. 2021. DOI: [10.1088/2058-9565/abe458](https://doi.org/10.1088/2058-9565/abe458) arXiv: [2009.07613](https://arxiv.org/abs/2009.07613) [quant-ph].
- [81] T. Haug and M. S. Kim, “Scalable measures of magic resource for quantum computers,” *PRX Quantum*, vol. 4, no. 1, p. 010301, Jan. 2023. DOI: [10.1103/PRXQuantum.4.010301](https://doi.org/10.1103/PRXQuantum.4.010301) arXiv: [2204.10061](https://arxiv.org/abs/2204.10061).
- [82] S. A. Hill and W. K. Wootters, “Entanglement of a pair of quantum bits,” *Phys. Rev. Lett.*, vol. 78, pp. 5022–5025, 26 Jun. 1997. DOI: [10.1103/PhysRevLett.78.5022](https://doi.org/10.1103/PhysRevLett.78.5022)
- [83] J. L. Beckey, N. Gigena, P. J. Coles, et al., “Computable and operationally meaningful multipartite entanglement measures,” *Phys. Rev. Lett.*, vol. 127, p. 140501, 14 Sep. 2021. DOI: [10.1103/PhysRevLett.127.140501](https://doi.org/10.1103/PhysRevLett.127.140501)
- [84] C. R. Harris, K. J. Millman, S. J. van der Walt, et al., “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. DOI: [10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2)
- [85] J. L. Beckey, G. Pelegrí, S. Foulds, et al., “Multipartite entanglement measures via Bell-basis measurements,” *Physical Review A*, vol. 107, no. 6, p. 062425, Jun. 2023. DOI: [10.1103/PhysRevA.107.062425](https://doi.org/10.1103/PhysRevA.107.062425)

- [86] S. Foulds, O. Prove, and V. Kendon, “Generalising multipartite concentratable entanglement for practical applications: Mixed, qudit, and optical states,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 382, no. 2287, p. 20240411, Dec. 2024. DOI: [10.1098/rsta.2024.0411](https://doi.org/10.1098/rsta.2024.0411) arXiv: [2112.04333](https://arxiv.org/abs/2112.04333) [quant-ph].
- [87] S. Kullback and R. A. Leibler, “On Information and Sufficiency,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, Mar. 1951, Publisher: Institute of Mathematical Statistics. DOI: [10.1214/aoms/1177729694](https://doi.org/10.1214/aoms/1177729694)