

Open and Closed Loop Approaches for Energy Efficient Quantum Optimal Control

Fauquenot, Sebastiaan; Sarkar, Aritra; Feld, Sebastian

DOI

[10.1002/qute.202400690](https://doi.org/10.1002/qute.202400690)

Publication date

2025

Document Version

Final published version

Published in

Advanced Quantum Technologies

Citation (APA)

Fauquenot, S., Sarkar, A., & Feld, S. (2025). Open and Closed Loop Approaches for Energy Efficient Quantum Optimal Control. *Advanced Quantum Technologies*. <https://doi.org/10.1002/qute.202400690>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Open and Closed Loop Approaches for Energy Efficient Quantum Optimal Control

Sebastiaan Fauquenot, Aritra Sarkar, and Sebastian Feld*

This research investigates the possibility of using quantum optimal control techniques to co-optimize the energetic cost and the process fidelity of a quantum unitary gate. The energetic cost is theoretically defined, and thereby, the gradient of the energetic cost for pulse engineering is derived. The Pareto optimality is empirically demonstrated in the trade-off between process fidelity and energetic cost. Thereafter, two novel numerical quantum optimal control approaches are proposed: i) energy-optimized gradient ascent pulse engineering (EO-GRAPE) as an open-loop gradient-based method, and ii) energy-optimized deep reinforcement learning for pulse engineering (EO-DRLPE) as a closed-loop method. The performance of both methods is probed in the presence of increasing noise. It is found that the EO-GRAPE method performs better than the EO-DRLPE methods with and without a warm start for most experimental settings. Additionally, for one qubit unitary gate, the correlation between the Bloch sphere path length and the energetic cost is illustrated.

1. Introduction

The field of quantum computing (QC) is undergoing rapid development in both theoretical and practical aspects from both academic and industrial stakeholders. At its foundation, QC involves orchestrating quantum mechanical properties for information processing^[1,2] via quantum algorithms, thereby promising a significant reduction in computational resources^[3] for specific applications over their classical counterparts. Exemplary use cases include simulating quantum mechanical systems^[4] toward novel discovery in material sciences and breaking cryptographic

protocols^[5] ubiquitous for secure transactions over the internet. The required orchestration of quantum information toward QC is physically implemented by engineering quantum processing units (QPU), which are currently prototyped using a myriad of technologies like superconducting circuits, trapped ions, photonics, electron spin, etc. A significant challenge in the QC field^[6–8] is to manufacture better quality and scalable QPU against the fragility of quantum information from environmental noises and operational imperfections.

Akin to classical computers, the interface between the QC applications and the processor is organized into translation layers, called the quantum computation stack,^[9,10] as shown in **Figure 1**. From top to bottom, first, the application is formulated as a quantum algorithm and expressed in a quantum programming language. Then, a

quantum compiler decomposes and optimizes the high-level code into native operations supported by the target QPU. Thereafter, the quantum microarchitecture schedules and issues the low-level instructions in real-time. These instructions (like initialization, unitary gates, and measurements) further need to be translated to corresponding analog pulses that optimally control the accessible degrees of freedom of the quantum system. These electromagnetic analog signals perform the necessary transformation for synthesizing quantum unitary gates on specific addressable qubits while mitigating the undesirable effects of noise. Eventually, these hardware-aware signals implement the desired logical operation dictated by the hardware-agnostic quantum algorithm on the target QPU.

The QC stack, as discussed above, provides abstraction layers to organize the research and development. Thus, operations, including a computationally universal set of gates, initialization, and measurements, are provided as primitives for the microarchitecture layer. In the ideal scenario, the pulse-level implementation details of these operations are predetermined in the control layer during the characterization and calibration of the QPU prior to deployment. However, in the current noisy intermediate-scale quantum (NISQ) computing era, this abstraction^[11] is preventing further optimization of the QPU performance. Pulse-level control, when exposed to higher stack layers, allows advanced operating procedures that can mitigate the effect of noise dynamics and fine-tune a more extensive set of unitary gates with respect to other dependencies (e.g., cross-talk). While there are many advantages to pulse control, such as a high degree of flexibility

S. Fauquenot, A. Sarkar, S. Feld
Quantum Computing Division
QuTech
Delft 2628 CJ, The Netherlands
E-mail: s.feld@tudelft.nl

S. Fauquenot, A. Sarkar, S. Feld
Department of Quantum & Computer Engineering
Delft University of Technology
Delft 2628 CD, The Netherlands

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/qute.202400690>

© 2025 The Author(s). Advanced Quantum Technologies published by Wiley-VCH GmbH. This is an open access article under the terms of the [Creative Commons Attribution](#) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/qute.202400690

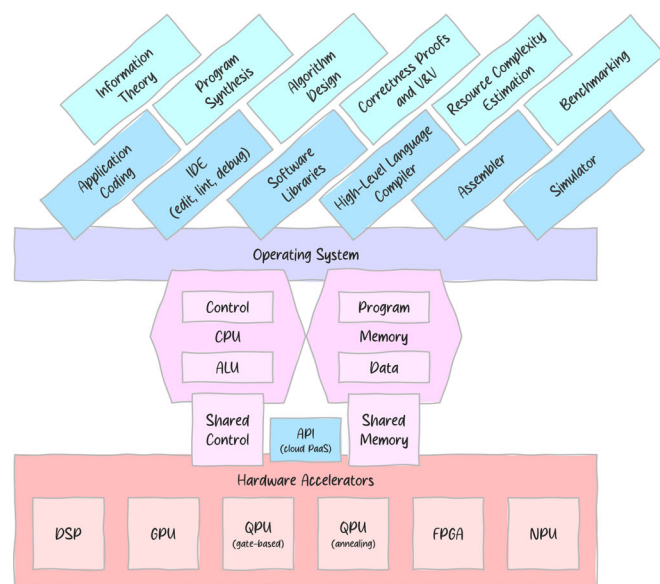


Figure 1. Overview of the system design of a quantum accelerator with classical control and various software modules required for research and development is shown on the left. The different abstraction layers for full-stack quantum computing are shown on the right. This research pertains to the optimal control layer highlighted with a dotted outline.

and higher speed of gate execution, there are also certain downsides to using pulse control, such as calibration requirements and overhead and scalability bottlenecks. Nevertheless, advancing research in this direction is not only imperative for current QC roadmaps to extract maximal performance but also allows a principled theoretical understanding of the limits of quantum information processing using classical control algorithms and electronics.

Current pulse control approaches focus primarily on optimizing the fidelity score. Motivated by the theoretical possibility and operational need, this research concerns a novel approach for pulse-level unitary gate synthesis, that co-optimizes the fidelity along with the pulse energy. Within the allied field of quantum thermodynamics, there is a growing interest in the possibility of achieving quantum advantage through the perspective of energy efficiency^[12] instead of computational resources like runtime and memory needs. This is imperative for at least two specific reasons. First, one of the earliest motivations of quantum computation stems from reversible computation^[13] with minimal energetic cost instead of a computational complexity advantage. Second, quantum processors within a quantum accelerator stack are benchmarked in performance against state-of-the-art classical high-performance computing (HPC) systems. In such HPC systems, optimizing and benchmarking the energetic cost^[14] is motivated by both operational costs and environmental sustainability directives. While this direction is gaining traction, specific research in the energetic cost of a quantum computational process is both sparse and theoretical at present. For example,^[15] proves an inequality bounding the change of Shannon information encoded in the logical quantum states by quantifying the energetic cost of Hamiltonian gate operations. Subsequently,^[16] showed that optimal control problems can be solved within the powerful framework on quantum speed limits and derive state-independent lower bounds on energetic cost. Recent work in quantum optimal control (QOC) theory has primarily focused on

developing control to carry out quantum processes with the highest fidelity possible. These processes include quantum processes such as state initialization, quantum measurements, and implementing quantum unitary gates. However, in the context of the growing impetus in achieving quantum advantage through energy efficiency,^[17] it seems crucial to investigate the energy efficiency of quantum operations, in particular, unitary quantum gates within a pragmatic quantum compiler framework. To this effect, we address three main research questions in this article:

- RQ_1 : How can we estimate the energetic cost of synthesizing a quantum unitary gate?
- RQ_2 : What is the relation between the fidelity of unitary synthesis and the energetic cost associated with the pulse?
- RQ_3 : How can fidelity and the energetic cost be co-optimized within existing quantum optimal control strategies?

The core contributions of this research are summarized below:

- 1) A theoretical formulation of the energetic cost of implementing a quantum unitary gate using discrete control pulses and the gradient of the energetic cost with respect to the control parameters that are required to optimize the cost.
- 2) Development of a modified version of the gradient ascent pulse engineering (GRAPE) algorithm to co-optimize a quantum unitary gate's fidelity and energetic cost. The proposed novel energy-optimized algorithm is called EO-GRAPE.
- 3) Identification and analysis of the trade-off between the fidelity and energetic cost of implementing a quantum unitary gate.
- 4) Development of a deep reinforcement learning (DRL) agent able to learn and generate energy-optimized control pulses for a universal set of quantum unitary gates. The proposed novel energy-optimized method is called EO-DRLPE.

- 5) Benchmarks to evaluate the performance of the EO-GRAPE algorithm and EO-DRLPE method with increasing noise levels.
- 6) Evaluation of the optimality of the geodesic path cost of the two methods for synthesizing 1-qubit unitary gates.

The rest of the article is organized as follows. Section 2 provides background information on pulse engineering for unitary synthesis. Thereafter, a non-exhaustive survey and classification of quantum optimal control techniques is provided. The energetic cost of a quantum operation is introduced. In Section 3, the GRAPE algorithm is introduced. Thereafter, we present our novel EO-GRAPE algorithm that co-optimizes the energy and process fidelity based on the derived gradient. The trade-off between these factors is explored. Section 4 introduces the alternative closed-loop approach, EO-DRLPE and its consecutive performance analysis. In Section 5, we provide an analysis of the correlation between the energetic cost and Bloch sphere path length for single qubit unitary synthesis. Section 6 concludes the article and provides suggestions for future research directions.

2. Background and Related Work

In this section, the formulation of pulse-level unitary gate synthesis is presented. Thereafter, a brief overview of existing QOC techniques is discussed to highlight the rationale behind the two methods that are proposed for energetic cost optimization.

2.1. Pulse-Level Unitary Gate Synthesis

Quantum systems evolve over time according to the time-dependent Schrödinger equation (TDSE),^[18]

$$i\hbar \frac{d}{dt} |\psi(t)\rangle = \hat{H} |\psi(t)\rangle \quad (1)$$

where, \hat{H} represents the Hamiltonian of the quantum system. A general solution for this equation is given by,

$$|\psi(t)\rangle = e^{-i\hat{H}t/\hbar} |\psi(t_0)\rangle \equiv U(t) |\psi(t_0)\rangle \quad (2)$$

where,

$$U(t) = e^{-i\hat{H}t/\hbar} \quad (3)$$

$\hbar = \frac{h}{2\pi}$ is the reduced Planck constant and is typically set to 1 in natural units.

Quantum logic gates are implemented by carefully tuning the Hamiltonian of the quantum system over time. Thus, if we want to perform a specific unitary (such as a Hadamard or CNOT), we have to apply a specific aggregated Hamiltonian to the system. This aggregated Hamiltonian is effectuated over a specific duration by carefully tuning the accessible/controllable Hamiltonians of the system over that duration.

We oftentimes make a distinction between the so-called drift Hamiltonian \hat{H}_D and control Hamiltonian \hat{H}_C . The drift Hamiltonian pertains to the actual qubit(s), usually consisting of an individual term for the eigen-energy and eigenstates of the qubits,

as well as a coupling term between different qubits. The control Hamiltonian describes the external control fields that can be applied to the qubit(s).

Let us consider a simple case of a single qubit without any interaction terms. The qubit is described by the following Hamiltonian,

$$\hat{H}_D = \hbar\omega_0\hat{\sigma}_z \quad (4)$$

This implies the qubit will precess about the \hat{z} -axis with frequency ω_0 .

If the control field is applied on the $\hat{x}\hat{y}$ -plane, and rotates around the \hat{z} -axis with frequency ω_f , we can define the control Hamiltonian by,

$$\hat{H}_C = \hbar\omega_1 (\cos \omega_f t \hat{\sigma}_x + \sin \omega_f t \hat{\sigma}_y) \quad (5)$$

and we can thus rewrite our total Hamiltonian as,

$$\hat{H} = \hat{H}_D + \hat{H}_C = \hbar\omega_0\hat{\sigma}_z + \hbar\omega_1 (\cos \omega_f t \hat{\sigma}_x + \sin \omega_f t \hat{\sigma}_y) \quad (6)$$

The control fields or the time-dependent function of the control Hamiltonian operators are referred to as the control pulses that one can apply to the quantum system. A control pulse field k typically has three associated parameters: i) the control field amplitude a_k , ii) the control field frequency ω_k , and iii) the control field phase ϕ_k . Thus, the control field can be described as a function of these three parameters acting on a control Hamiltonian operator $\hat{\sigma}_k$ on the target qubits,

$$\hat{H}_k = f(a_k(t), \omega_k(t), \phi_k(t)) \hat{\sigma}_k = u_k(t) \hat{\sigma}_k \quad (7)$$

Adjusting these three parameters over time is the operational definition of pulse control. We will use the shorthand $u_k(t)$ to denote these three variable functions to generalize over control formalisms for other qubit modalities with different control parameters. Typically, only the amplitude $a_k(t)$ is tuned to achieve the required control, with the other tunable parameters kept fixed.

2.2. A Classification of Quantum Optimal Control Techniques

The collection of techniques focused on achieving high-quality control of quantum systems is referred to as quantum optimal control (QOC). It involves methods to design and implement electromagnetic field configurations that can effectively steer quantum processes at the atomic or molecular scale in the best way possible.^[19,20]

QOC techniques can be broadly classified as analytical and numerical methods.^[21] As the names suggest, the analytical methods use mathematical theory and representations of the quantum system to analytically solve for the optimal pulse, while numerical methods leverage the power of discretization and linearization to allow the use of numerical methods and algorithms. An example of an analytical method for QOC is Pontryagin's maximum principle.^[22] It models any optimal control together with the optimal state trajectory as a 2-point boundary value problem with a maximum condition of the control Hamiltonian. Using this description, one can use a time-varying Lagrangian description and

multiplier vector to solve the problem. Another method to analytically devise optimal control pulses is through a generalization of adiabatic evolution, for example, via derivative removal by adiabatic gates (DRAG).^[23] If the system is more complex, such as multiple qubits, interactions, or noise systems, often a perturbative expansion^[24] is needed to solve it analytically. SU(2) Lie algebra^[25] can also be used to devise rules analytically when a quantum system is fully reachable or controllable.

While analytical methods are, by default, open-loop, numerical methods can further be classified into closed-loop and open-loop methods. Closed-loop methods do not require an existing theoretical model of the system and use experimental feedback loops, with measurement data from the quantum system, to devise and refine the control pulses. Contrarily, open-loop methods rely on using an existing theoretical model of the quantum system and process in question and, accordingly, design optimal control pulses based on that theoretical model that is directly applied to the system.

A popular closed-loop method is a sub-class of machine learning (ML) called reinforcement learning (RL). In RL, a so-called agent is allowed to take certain actions and apply them to the environment. The environment then outputs a certain state, accompanied by a certain reward based on the action that was taken. In framing QOC as an RL problem, the action is the control pulse, the environment is the quantum system, and the state is the output state of the quantum system after applying that specific control pulse.^[26] The negative reward (or penalty) is modeled as a measure (e.g., process distance) between the target environment state and the achieved environment state. In the case of a model-free RL algorithm, the policy to decide the action at every step can either be updated using various methods like the use of a neural network (NN) or via a method called Q-Learning.^[27] To increase the robustness in the feedback loop, the information of the Hessian matrix can be potentially used in combination with a closed-loop learning-based algorithm, e.g., the msMS-DE algorithm.^[21] Samples of a quantum system for training purposes can also be used to evaluate the performance within a test and evaluation phase. This is called sampling-based learning control (SLC).^[28] Finally, there are hybrid methods that use the gradient-based GRAPE algorithm in combination with reinforcement learning or other machine learning methods, such as s-GRAPE or b-GRAPE.^[21]

Open-loop methods (analytical and numerical) can further be categorized into gradient-based or gradient-free. Gradient-based methods rely on the calculation of local gradients to, in return, move toward a local optimum, while gradient-free methods usually use variants of stochastic search algorithms to reach an optimum.^[21]

An example of a gradient-based open-loop numerical method is the so-called Krotov method.^[29] This method utilizes Lagrange multipliers based on the process fidelity to find optimal pulses for gate synthesis and state-to-state transfer. If representing the control pulse in an analytical form is preferred, we may use the so-called gradient optimization of analytical controls (GOAT) method.^[30] This method uses an educated guess of the shape of the pulse (e.g., Gaussian), to form a coupled system of equations, which can then be solved numerically by forward integration methods, such as the Runge–Kutta method. Potentially the most well-known and widely used method is the gradient ascent

pulse engineering (GRAPE) method.^[31] This method uses the discretization of the control operators to iteratively solve for the optimal control pulse.

An example of gradient-free open-loop numerical methods is the chopped random basis optimization (CRAB).^[32] This method leverages the fact that optimal solutions could reside in a low-dimensional subspace of the total search space. The control sequences are represented as a linear addition of basis functions. Another gradient-free method is based on an evolutionary process called the genetic algorithm (GA).^[33] This method utilizes a set of initial random guesses and then evolves these candidates based on a fitness function. A variation of GA is the differential evolution (DE) method.^[34] **Table 1** provides an overview of the quantum optimal control technique classifications as discussed.

The methods surveyed above focus on achieving a high fidelity of control as the parameter defining optimality. In this research, we use both an open-loop and a closed-loop method to assess whether it is possible to incorporate energy-efficient control pulses that can trade-off an optimality cost accounting for both fidelity and energy. To this end, we have chosen the naive gradient ascent pulse engineering (GRAPE) as the open-loop method and deep reinforcement learning (DRL) as the closed-loop method that we extend to include energetic cost tuning. The motivation for this choice will be discussed in the corresponding sections detailing the implementations.

2.3. Energetic Cost of a Unitary Operation

The energy of a quantum state is defined as the expectation value of the Hamiltonian,

$$\langle E \rangle = \langle \psi | \hat{H} | \psi \rangle \quad (8)$$

However, the energetic cost of a unitary operation is harder to define.^[15] defines the energetic cost as the time-integrated norm of the Hamiltonian over the total duration of the unitary gate,

$$E[U] = \int_0^T dt \|\hat{H}(t)\| \quad (9)$$

For single qubit quantum unitary gates, this cost can be visually interpreted on the Bloch Sphere. State-to-state transfer between two quantum states can be accomplished via various paths in Hilbert space. The most energy-efficient way to implement a single unitary gate corresponds to the geodesic between the two quantum states on the Bloch sphere, i.e., the shorter the path length between the two states, the lower the energetic cost of the quantum unitary gate.^[16] As an example, **Figure 2** visualizes different path lengths between two quantum states on the Bloch sphere. The unitary gate along the path depicted with blue vectors, in this case, has a lower energetic cost than the unitary gate that implements the red or green vectors. Previous work has focused on minimizing the gate duration while satisfying hardware constraints on control amplitudes by introducing penalty terms in the cost function.^[35] In the context of adiabatic quantum computing, quantum annealing, and gate-based quantum optimization penalizing the amplitude of control has been explored in Ref. [36].^[37] has demonstrated the trade-off of fidelity and speed in quantum dots for a universal gate set.

Table 1. Overview of some existing methods of Quantum Optimal Control, including their category based on analytical, numerical, closed loop, open loop, gradient-based, and gradient-free methods.

Quantum Optimal Control method	Analytical	Numerical	Closed loop	Open loop	Gradient-based	Gradient-free
Pontryagin's maximum principle	✓			✓	✓	
Derivative removal by adiabatic gates	✓			✓		✓
Perturbative expansion	✓			✓	✓	
SU(2) Lie algebra	✓			✓		✓
Reinforcement learning		✓	✓			
Q-Learning		✓	✓			
msMS-DE		✓	✓			
Sampling-based learning control		✓	✓			
s-GRAPE		✓	✓			
b-GRAPE		✓	✓			
Krotov		✓		✓	✓	
Gradient optimization of analytical control		✓		✓	✓	
Gradient ascent pulse engineering		✓		✓	✓	
Chopped random basis optimization		✓		✓		✓
Genetic algorithm		✓		✓		✓
Differential evolution		✓		✓		✓

In this research, we will investigate the cost governed by Equation (9), as well as the path length representation, for computing the energetic cost of a quantum unitary gate. Although one cannot visualize multi-qubit unitary gates on the Bloch sphere, similar geometric arguments in corresponding higher dimensional manifolds hold there as well.

3. Energy-Optimized Gradient Ascent Pulse Engineering

In this section, the gradient ascent pulse engineering (GRAPE) algorithm is reviewed, which uses a gradient that is based on the

fidelity of the unitary synthesis. Thereafter, we derive the gradient of the energetic cost that is used for co-optimizing within our proposed energy-optimized GRAPE algorithm. Based on this gradient, we present our proposed EO-GRAPE quantum control algorithm. Our implementation of the algorithm is used to study the resultant control pulses, the trade-off between energy and fidelity of the synthesized target unitary, and the dependence of the algorithm on system noise using a QPU simulator.

3.1. GRAPE Algorithm Preliminaries

The total Hamiltonian of a closed quantum system consists of a constant drift Hamiltonian \hat{H}_D , and the sum of control Hamiltonians \hat{H}_k governed by parameters $u_k(t)$. Thus, the total system Hamiltonian can be written as,

$$\hat{H} = \hat{H}_D + \hat{H}_C = \hat{H}_D + \sum_{k=1}^K \hat{H}_k = \hat{H}_D + \sum_{k=1}^K u_k(t) \hat{\sigma}_k \quad (10)$$

where $\hat{\sigma}_k$ can be any Pauli operator $\hat{\sigma}_x$, $\hat{\sigma}_y$, or $\hat{\sigma}_z$. The dynamics of a state $|\psi(t)\rangle$, represented as a vector in a Hilbert space, evolves according to the time-dependent Schrödinger equation as reviewed earlier in Equation (1). The time-evolution of the quantum state over time T is discretized into N steps of Δt . During a specific time step, n , the evolution is given by the unitary operator,

$$U_n = \exp \left\{ -i\Delta t \left(\hat{H}_D + \sum_{k=1}^K u_k(n) \hat{\sigma}_k \right) \right\} \quad (11)$$

Thus, the total unitary that is implemented after time $T = N \cdot \Delta t$ is given by:

$$U(t = T) = U_N \dots U_1 = \prod_{n=1}^N \exp \left\{ -i\Delta t \left(\hat{H}_D + \sum_{k=1}^K u_k(n) \hat{\sigma}_k \right) \right\} \quad (12)$$

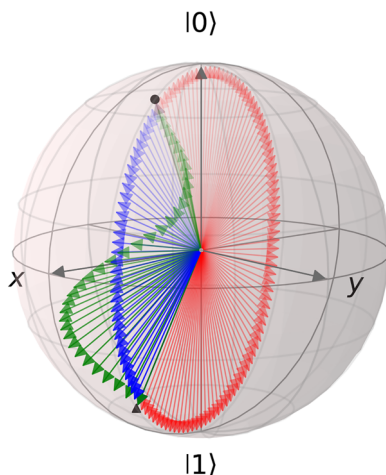


Figure 2. Visual interpretation of an energy-efficient quantum unitary gate. The three paths (blue, red, and green) accomplish the same transformation between the two states. The optimal geodesic path (in blue) requires a rotation against a single axis along an angle $\leq \pi$.

Note that we use the product notation to denote the right-to-left product ordering of the elements.

For a specific time slice, the forward and backward propagators refer to the rest of the unitary after and before this time slice, respectively, such that the total unitary is constructed from their composition. The forward propagator X_n is given by,

$$X_n \equiv U_n \dots U_1 \quad (13)$$

Accordingly, the backward propagator P_n can be defined as,

$$P_n \equiv U_{n+1}^\dagger \dots U_N^\dagger \quad (14)$$

We also need to define a performance function that is to be maximized. In the case of quantum unitary gate synthesis, this function is defined as the overlap between the target unitary U_T and the final unitary after time T ,

$$\Phi = |\langle U_T | U(T) \rangle|^2 = |\text{Tr}[U_T^\dagger U(T)]|^2 \quad (15)$$

The GRAPE algorithm iteratively updates each control parameter u_k for each time step n according to the gradient of the performance function Φ with respect to the control parameters u_k . This is repeated for a certain amount of iterations, called the GRAPE-iterations, and denoted by N_G . This update rule, for an arbitrary step size ϵ , is defined^[31] as,

$$u_k(n) \rightarrow u_k(n) + \epsilon \frac{\partial \Phi}{\partial u_k(n)} \quad (16)$$

To evaluate the gradient, the derivative of the performance function with respect to the control parameters $u_k(n)$ needs to be evaluated. To derive this, first, the performance function is expressed in terms of the forward and backward propagators,

$$\Phi = \langle U_T | U_N \dots U_1 \rangle \langle U_1 \dots U_N | U_T \rangle = \langle P_n | X_n \rangle \langle X_n | P_n \rangle \quad (17)$$

By using perturbation theory^[31] to the first order in $\partial u_k(n)$,

$$\begin{aligned} \frac{\partial \Phi}{\partial u_k(n)} &= -\langle P_n | X_n \rangle \langle i \Delta t \hat{\sigma}_k X_n | P_n \rangle - \langle P_n | i \Delta t \hat{\sigma}_k X_n \rangle \langle X_n | P_n \rangle \\ &= -2\mathcal{R}e\{\langle P_n | i \Delta t \hat{\sigma}_k X_n \rangle \langle X_n | P_n \rangle\} \end{aligned} \quad (18)$$

By using the definition of the gradient, the forward and backward operator, and by updating the control parameters according to Equation (16), we can arrive at global maxima of the performance function for a specific set of control parameters $u_k(n)$.

3.2. Energy Gradient

In this research, a novel performance function is introduced and utilized, consisting of a fidelity part and an energetic cost part.

The energetic cost of a discretized pulse based on Equation (9) is defined as,

$$\hat{C}[U(T)] = \frac{\sum_{n=1}^N \Delta t \left\| \hat{H}_D + \sum_{k=1}^K u_k(n) \hat{\sigma}_k \right\|}{T \left\| \hat{H}_D + \sum_{k=1}^K \hat{\sigma}_k \right\|} \equiv \phi_e \quad (19)$$

For brevity, the normalization constant in the denominator will be referred to as N_e .

We define the total cost function as,

$$\begin{aligned} \Phi &= w_f \left(1 - \left| \frac{\text{Tr}(U_T^\dagger U(T))}{\text{Tr}(U_T^\dagger U_T)} \right|^2 \right) \\ &+ w_e \left(\frac{1}{N_e} \sum_{n=1}^N \Delta t \left\| \hat{H}_D + \sum_{k=1}^K u_k(n) \hat{\sigma}_k \right\| \right) \end{aligned} \quad (20)$$

Thus, the gradient of Φ with respect to $u_k(n)$ can be written as the weighted linear addition of the two components of the cost function based on fidelity, ϕ_f and energy, ϕ_e ,

$$\frac{\partial \Phi}{\partial u_k(n)} = w_f \frac{\partial \phi_f}{\partial u_k(n)} + w_e \frac{\partial \phi_e}{\partial u_k(n)} \quad (21)$$

The gradient of fidelity ϕ_f based on Equation (18) is,

$$\frac{\partial \phi_f}{\partial u_k(n)} = -2\mathcal{R}e\{\langle P_n | i \Delta t \hat{\sigma}_k X_n \rangle \langle X_n | P_n \rangle\} \quad (22)$$

The energetic cost part of the cost function can be written as:

$$\phi_e = \frac{1}{N_e} \sum_{n=1}^N \Delta t \left\| \hat{H}_D + \sum_{k=1}^K u_k(n) \hat{\sigma}_k \right\| \quad (23)$$

Using the identity $\|A\| = \sqrt{\text{Tr}(A^* A)}$, the expression above can be expanded as follows:

$$\begin{aligned} \phi_e &= \frac{1}{N_e} \sum_{n=1}^N \Delta t \\ &\times \sqrt{\text{Tr} \left(\hat{H}_D^* \hat{H}_D + \sum_{k=1}^K u_k(n) (\hat{H}_D^* \hat{\sigma}_k + \hat{\sigma}_k^* \hat{H}_D) + \sum_{k=1}^K \sum_{k'=1}^K u_k(n) u_{k'}(n) \hat{\sigma}_k^* \hat{\sigma}_{k'} \right)} \end{aligned} \quad (24)$$

We define the term within the square root as $g(u_k(n)) = \text{Tr}(\dots)$ and $f(g(u_k(n)))$ as

$$f(g(u_k(n))) = \frac{1}{N_e} \sum_{n=1}^N \Delta t \sqrt{g(u_k(n))} \quad (25)$$

The gradient can then be expressed in terms of $f(g(u_k(n)))$ and $g(u_k(n))$:

$$\frac{\partial \phi_e}{\partial u_k(n)} = \frac{\partial f}{\partial g} \frac{\partial g}{\partial u_k(n)} \quad (26)$$

The first part is given by:

$$\frac{\partial f}{\partial g} = \frac{1}{N_e} \sum_{n=1}^N \Delta t \frac{1}{2\sqrt{g(u_k(n))}} \quad (27)$$

while the second part is given by:

$$\begin{aligned} \frac{\partial g}{\partial u_k(n)} = & \text{Tr} \left(\sum_{k=1}^K \hat{H}_D^* \hat{\sigma}_k + \hat{\sigma}_k^* \hat{H}_D \right) \\ & + \text{Tr} \left(\sum_{k=1}^K \sum_{k'=1}^K (u_k(n) + u_{k'}(n)) \hat{\sigma}_k^* \hat{\sigma}_{k'} \right) \end{aligned} \quad (28)$$

The total gradient of ϕ_e with respect to $u_k(j)$ can thus be written as:

$$\frac{\partial \phi_e}{\partial u_k(n)} = \frac{1}{N_e} \sum_{n=1}^N \Delta t \frac{\text{Tr} \left(\sum_{k=1}^K \hat{H}_D^* \hat{\sigma}_k + \hat{\sigma}_k^* \hat{H}_D \right) + \text{Tr} \left(\sum_{k=1}^K \sum_{k'=1}^K (u_k(n) + u_{k'}(n)) \hat{\sigma}_k^* \hat{\sigma}_{k'} \right)}{2\sqrt{\text{Tr} \left(\hat{H}_D^* \hat{H}_D + \sum_{k=1}^K u_k(n) (\hat{H}_D^* \hat{\sigma}_k + \hat{\sigma}_k^* \hat{H}_D) + \sum_{k=1}^K \sum_{k'=1}^K u_k(n) u_{k'}(n) \hat{\sigma}_k^* \hat{\sigma}_{k'} \right)}} \quad (29)$$

3.3. EO-GRAPE Algorithm

Combining Equation (22) with Equation (29), the total gradient of Φ with respect to the control parameters $u_k(n)$ is given as,

$$\frac{\partial \Phi}{\partial u_k(n)} = \begin{pmatrix} -2w_f \\ -w_e \end{pmatrix}' \left(\frac{1}{N_e} \sum_{n=1}^N \Delta t \frac{\mathcal{R}e\{\langle P_n | i\Delta t \hat{\sigma}_k X_n \rangle \langle X_n | P_n \rangle\}}{2\sqrt{\text{Tr} \left(\hat{H}_D^* \hat{H}_D + \sum_{k=1}^K u_k(n) (\hat{H}_D^* \hat{\sigma}_k + \hat{\sigma}_k^* \hat{H}_D) + \sum_{k=1}^K \sum_{k'=1}^K u_k(n) u_{k'}(n) \hat{\sigma}_k^* \hat{\sigma}_{k'} \right)}} \right) \quad (30)$$

Based on these gradients that were derived in Section 3.2 leading to Equation (30), we can now formulate our novel EO-GRAPE algorithm that employs the updating rule of,

$$u_k(n) \rightarrow \epsilon_f \frac{\partial \phi_f}{\partial u_k(n)} + \epsilon_e \frac{\partial \phi_e}{\partial u_k(n)} \quad (31)$$

The EO-GRAPE can be implemented as algorithm 1.

To test the fidelity-energy trade-off and for the other experiments presented later in this research, we consider two exemplary drift Hamiltonians:

$$\begin{aligned} \hat{H}_D^1 & \in \left\{ \hat{\sigma}_i, \frac{\hbar\omega_1}{2} \hat{\sigma}_z \right\} \\ \hat{H}_D^2 & \in \left\{ \frac{\hbar\omega_1}{2} \hat{\sigma}_z^{(1)} \otimes \hat{I}_2^{(2)} + \frac{\hbar\omega_2}{2} \hat{I}_2^{(1)} \otimes \hat{\sigma}_z^{(2)} + \hbar J \hat{\sigma}_z^{(1)} \otimes \hat{\sigma}_z^{(2)} \right\} \end{aligned} \quad (32)$$

Where ω is the eigenfrequency of the target qubit, and J is the coupling strength between the two target qubits.

Algorithm 1 Energy-Optimized Gradient Ascent Pulse Engineering

```

{u_k(n)} = Initial guess of u_k(n) for all k
for g = 1 to N_G do
  for n = 1 to N do
    for k = 1 to K do
      Calculate X_n
      Calculate P_n
      Evaluate ∂φ_f/∂u_k(n)
      Evaluate ∂φ_e/∂u_k(n)
      Update control amplitudes u_k(n) → ε_f ∂φ_f/∂u_k(n) + ε_e ∂φ_e/∂u_k(n)
    end for
  end for
end for
return {u_k(n)}

```

In **Figure 3**, the effect of increasing or decreasing the weight associated with fidelity w_f and energetic cost w_e is shown. As one can see, the higher the value of w_e , the lower the amplitude of the control pulses are, decreasing the area and thus decreasing

the energetic cost, intuitively matching our expectations. One can also see some other harmonics being introduced when increasing w_e to decrease the energetic cost of the pulses. An overview of the parameters such as U_T , \hat{H}_D , $\hat{\sigma}_k$, and others are given in the figure caption, where T_1 and T_2 are the relaxation and decoherence times of the qubit respectively, and $U_T = \text{RAND}$ is sampled over a Haar random distribution.

3.4. Trade-Off Between Fidelity and Energetic Cost

Having derived the theoretical energetic cost and an algorithmic procedure to optimize pulses based on the energy gradient, in this section, we explore the following research question: what is the relation between fidelity and the energetic cost? Naively, one would expect that if the control amplitudes are zero, i.e., no energetic cost, the drift Hamiltonian will be solely responsible for the evolution. Thus, we expect a fidelity score that depends on the relation between the target unitary and the unitary implemented by the drift Hamiltonian for the pulse duration.

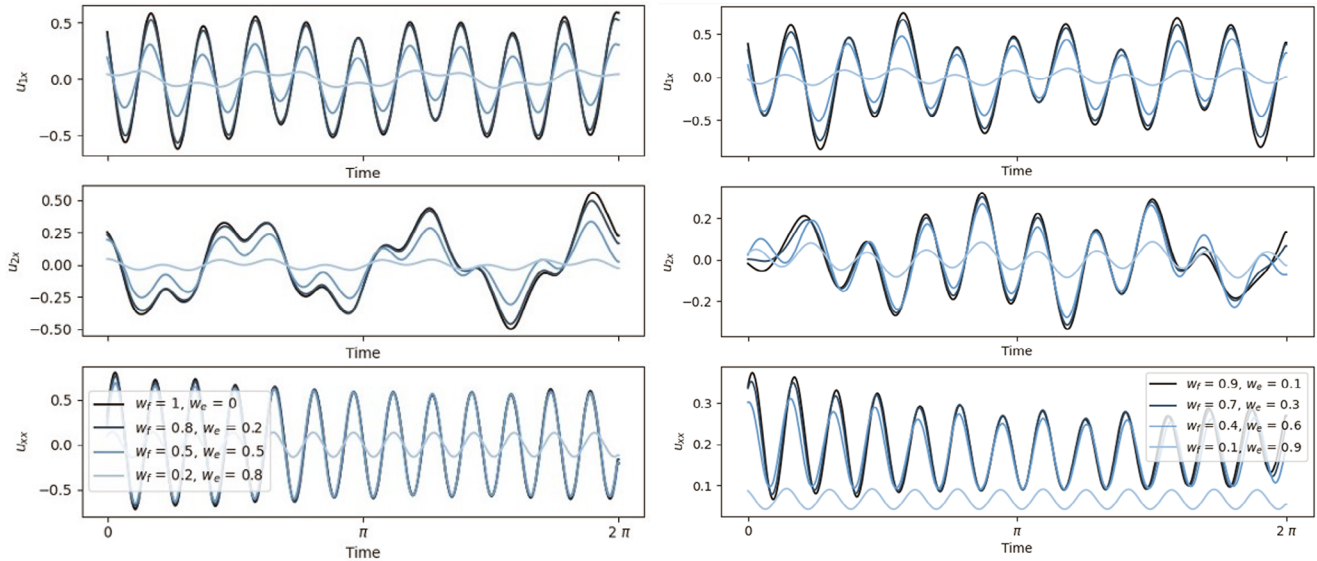


Figure 3. Example control pulses generated by the EO-GRAP algorithm for different weight settings. Parameters: $U_T = \text{RAND}$, $\hat{H}_D = \hat{H}_D^2$, $\hat{\sigma}_k \in \{\hat{\sigma}_x^1, \hat{\sigma}_x^2, \hat{\sigma}_x^1\hat{\sigma}_x^2\}$, $T_1 = \infty$, $T_2 = \infty$, $w_f = [1, 0.1]$, $w_e = [0, 0.9]$, $N = 500$, $N_G = 500$.

Note that the trade-off between energetic cost and fidelity has already been studied from the perspective of noise and dissipative dynamics.^[38] While having similar motivations, these are unrelated to the research presented in this article in that we consider all intermediate states in the state evolution to be pure without any environmental coupling.

To mitigate the dependence on the drift Hamiltonian, we sample the target unitary over a 2-qubit Harr random distribution. In **Figure 4**, the results of this experiment are depicted. The EO-GRAP algorithm was run for different sets of weights w_f and w_e . Each diamond marker on the plot represents a different weight setting between $[w_f : w_e] = [1 : 0] \dots [0.1 : 0.9]$ in steps of

0.1. These values have been plotted against each other for different values of the learning rates, ϵ_f and ϵ_e . The Pareto front depicting the trade-off between fidelity and the energetic cost is clearly visible, where a reduction in energetic cost directly leads to an increase in infidelity or a decrease in fidelity. From our learning rate optimization analysis, it can be concluded that the optimal hyperparameter is $\epsilon_f = 1$ and $\epsilon_e = 100$.

This result matches our intuition that to achieve a lower energetic cost for a quantum unitary gate, one needs to inherently decrease the area or amplitude of the control pulses, resulting in a lower process fidelity. Nevertheless, as seen on the high fidelity end of the curve in **Figure 4b**, the two factors are not completely

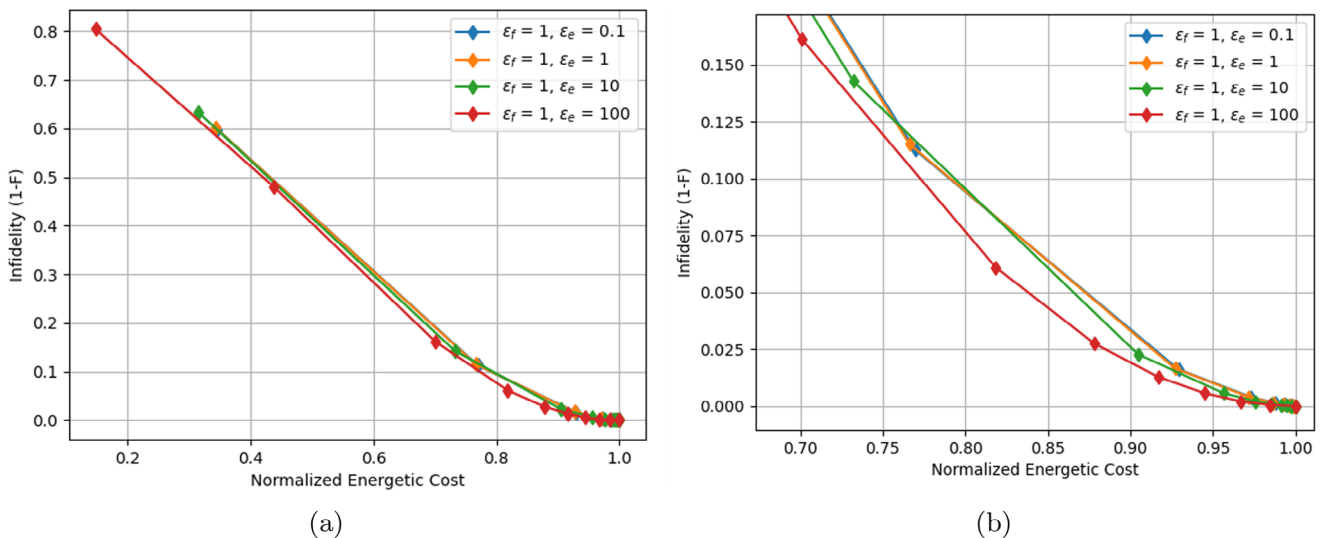


Figure 4. a) Infidelity and energetic cost values for 10 different weight settings (indicated by diamond markers for each color) and learning rates ϵ_e , showing the trade-off or Pareto front between fidelity and energetic cost. b) Zoomed-in view of the Pareto front between fidelity and energetic cost. Parameters: $U_T = \text{RAND}$, $\hat{H}_D = \hat{H}_D^2$, $\hat{\sigma}_k \in \{\hat{\sigma}_x^1, \hat{\sigma}_x^2, \hat{\sigma}_x^1\hat{\sigma}_x^2\}$, $T_1 = \infty$, $T_2 = \infty$, $w_f = [1, 0.1]$, $w_e = [0, 0.9]$, $N = 500$, $N_G = 100$.

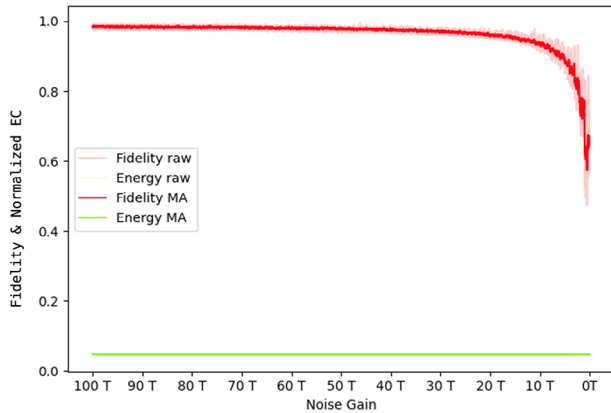


Figure 5. Fidelity (red) and Energetic Cost (green) of EO-GRAPE generated control pulses as a function of decreasing decoherence time, or increasing noise level. The moving average (MA) is shown in bold colors. Parameters: $U_T = \text{Hadamard}$, $\hat{H}_D = \hat{H}_D^1$, $\hat{\sigma}_k \in \{\hat{\sigma}_x^1\}$, $T_1 = [100T, 17]$, $T_2 = [100T, 17]$, $w_f = 0.7$, $w_e = 0.3$, $N = 100$, $N_G = 500$.

inversely proportional to each other. We can, therefore, still decrease the energetic cost of a quantum unitary gate by roughly 10% while decreasing the fidelity by roughly 1%, which can be compensated with error correction [39]. As an estimate from this example of a noise-less setting, one could decrease the energetic cost of each quantum unitary gate by 10% while maintaining a minimum 2-qubit gate fidelity of 99%. Thus, *using EO-GRAPE, we can minimize the energetic cost while maintaining the gate fidelity above the error budget of the quantum application or the threshold of the quantum error correction code.*

To test the performance of the EO-GRAPE algorithm while increasing the noise in the system, we inject an increasing noise over the EO-GRAPE iterations. From the result of the infidelity, as shown in **Figure 5**, we infer that the algorithm is able to achieve high fidelity and low energetic cost throughout most noise settings. From a noise gain setting of $20T$, where T is defined as the pulse duration, we can see that the performance decreases quite rapidly. Note that this noise setting is unrealistically high for current quantum devices.

4. Energy-Optimized Deep Reinforcement Learning Based Pulse Engineering

As introduced in Section 2.2, reinforcement learning can be used for quantum pulse engineering for unitary gate synthesis. In such a formulation, the action of the RL agent is the control pulse, the environment is the quantum system, and the observation is the output state of the quantum system. The policy of an RL agent refers to the mechanism to decide the action based on the history of past actions and observations, with additional hyperparameters like the horizon, exploration-exploitation trade-off, etc. In model-free RL, the policy is often implemented as a neural network (NN). The weights of the NN are updated by training on known environments. In this section, we introduce the agents for energy-optimized deep reinforcement learning-based pulse engineering (EO-DRLPE). Thereafter, we present the architectural design of the QOC strategy and its performance in noiseless and noisy cases.

4.1. Agent Design for EO-DRLPE

In this research, two different reinforcement learning agents are utilized. Reinforcement Learning Agent 1 (RLA-1) interacts with the quantum environment and is responsible for actually devising the pulses. Reinforcement Learning Agent 2 (RLA-2) is responsible for approximating pulses generated by the EO-GRAPE algorithm. This can potentially be used as an initial policy of RLA-1. If RLA-2 is used, we will refer to this as a warm start,[40] as a pre-training and transfer learning method to initialize the weights of RLA-1 as opposed to random initialization.

Both agents use the TensorFlow REINFORCE algorithm.[42] First, the agent observes the state returned by the environment and chooses an action based on its policy. When the action is sent to the environment, the agent will receive a corresponding reward. The actions and rewards are continuously being registered in the replay buffer. After a certain number of actions, the policy is updated based on the use of a fully connected, deep feed-forward neural network. In the remainder of this paper, we will refer to the number of nodes per layer as *layer_params* = (N_1, \dots, N_i) , with N_i being the number of nodes in layer i . **Figure 6** shows an overview of the different components and steps in the REINFORCE algorithm.[41]

For RLA-1, the Environment is a quantum simulator from QuTip,[43] called the QuTip Processor class. The environment has several input parameters, such as the number of qubits N_q , the drift Hamiltonian \hat{H}_D , the control Hamiltonians \hat{H}_k , and the decoherence times of each qubit T_1 and T_2 . The agent is allowed to implement an action, which in this case are the control pulses $u_k(n)$, with shape $(K \times N)$. The output of the environment, called the state, is, in this case, the output density matrix ρ_{out} of the quantum system after applying the action. The next decision of the agent is based on the state and the reward, which in our case is defined as,

$$r_{RLA-1} = w_f F(\rho_T, \rho_{out}) + w_e (1 - \hat{C}[U(T)]) \quad (33)$$

The reward is a linear weighted addition of the fidelity F between the target output density matrix ρ_T and the output density matrix after applying the action ρ_{out} , and the inverse of the normalized energetic cost of implementing a certain Unitary after time T , $\hat{C}[U(T)]$.

The agent makes a decision based on its policy function $\pi(A_n, S_n)$, which is constantly being updated by a neural network. For RLA-1, a deep feed-forward neural network with 1 input layer, 3 – 5 hidden layers, and 1 output layer is utilized.

The goal of RLA-2 is to mimic the pulses generated by the EO-GRAPE algorithm by minimizing the distance between the control pulses it generates and the control pulses generated by the EO-GRAPE algorithm as discussed in Section 3.3. The agent interacts with a custom Environment class, called the GRAPEApproximation class, that takes in a set of control pulses and outputs the theoretical unitary. The environment has input parameters such as the number of qubits N_q , the drift Hamiltonian \hat{H}_D , the control Hamiltonians \hat{H}_k , the number of time steps N , the total time T , and the number of EO-GRAPE iterations N_G . The agent is allowed to take an action, i.e., the control pulses $u_k(n)$, with shape $(K \times N)$. The state that the environment returns to the

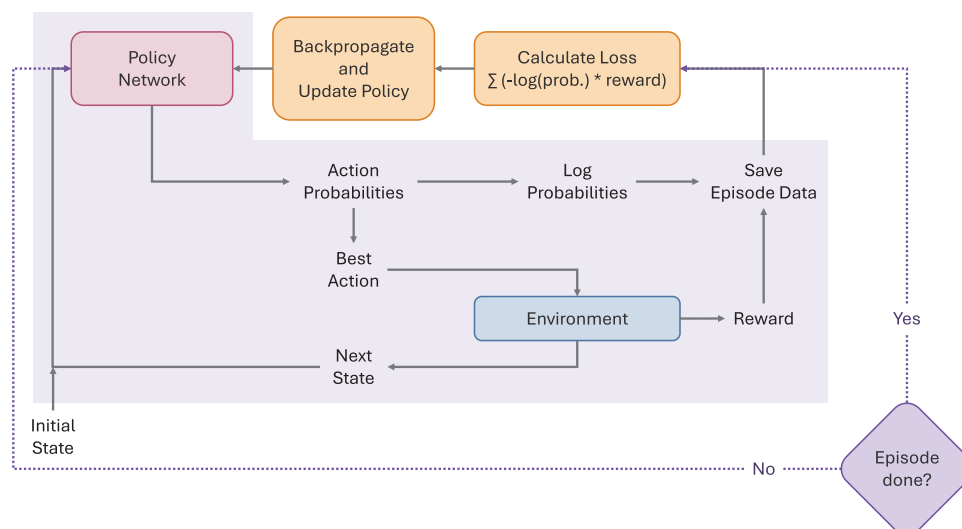


Figure 6. Schematic drawing of the basic workings of the TensorFlow REINFORCE agent.^[41]

agent is the theoretical unitary that the control pulses implement based on unitary evolution $U(u_k(n))$. RLA-2 utilizes a deep feed-forward NN with a similar configuration as RLA-1. The RLA-2 learns based on the reward that the environment returns, which is the L1 norm between the target pulse generated by the EO-GRAPE algorithm and the action that the agent takes,

$$r_{\text{RLA-2}} = - \sum_{n=1}^N \sum_{k=1}^K |u_k^{\text{EO-GRAPE}}(n) - u_k^{\text{RLA-2}}(n)|^2 \quad (34)$$

4.2. Workflow of Pulse-Level Unitary Synthesis

Our implementation of the software accompanying this research is contained in a comprehensive open-sourced Python package available on [GitHub](#). The package contains four main classes, with each having a different objective and dependencies. The four classes include two quantum environment classes and two reinforcement learning classes. The first quantum environment class (depicted by the red quantum simulator in **Figure 7**) interacts directly with the first reinforcement learning class (depicted by the RLA-1 neural network). The second quantum environment class (depicted by the graph output from EO-GRAPE) only interacts with the second reinforcement learning class (depicted by the RLA-2 neural network). **Figure 7** shows a schematic overview of the four classes working together on the quantum optimal control problem of unitary pulse synthesis.

4.3. Performance Study

To evaluate the performance of the EO-DRLPE implementation, we evaluate the fidelity and energetic cost of implementing the Hadamard gate. The noise in the system is gradually increased (i.e., the T_1 and T_2 time is decreased) to assess the robustness of the system in noisy quantum processors. In **Figure 8**, the performance of the reinforcement learning agent both with and without a warm start (by RLA-2) is shown. As we can infer, the agent

without a warm start (in red) is able to reach slightly higher fidelity than the agent with a warm start (in blue). However, the energetic cost of the agent with a warm start (in purple) is more stable and lower than a randomly initialized RLA-1 (in green).

The EO-GRAPE algorithm calculates control parameters based on the target unitary and the drift and control Hamiltonian that are provided and thus does not take into account any noise in the system. It is, therefore, quite remarkable that the control pulses generated by the EO-GRAPE algorithm are still able to achieve high fidelity and low energetic costs in a highly noisy environment. The reinforcement learning agent without a warm start seems to be able to learn the noise characteristics of the system first and could, therefore, outperform the reinforcement learning agent with a warm start on fidelity. However, when we take into account the energetic cost as well, it is clear that the reinforcement learning agent with a warm start outperforms the agent without a warm start. In **Appendix A**, the performance benchmark of EO-GRAPE and the reinforcement learning agents is presented for a universal gate set (CNOT,

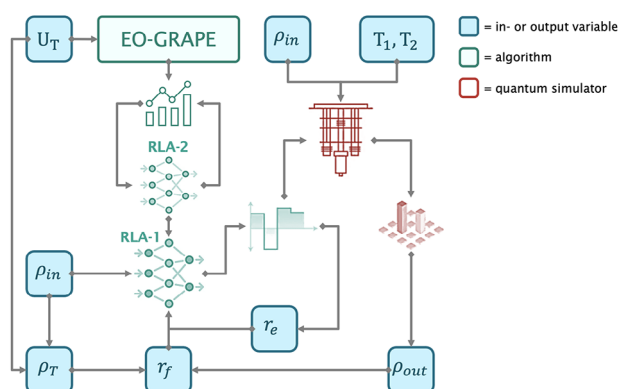


Figure 7. A schematic overview of the different classes and how they are co-dependent on each other. The red icons indicate quantum simulators, the blue icons indicate the input and output variables of the algorithms or simulators, and the green icons indicate an algorithm.

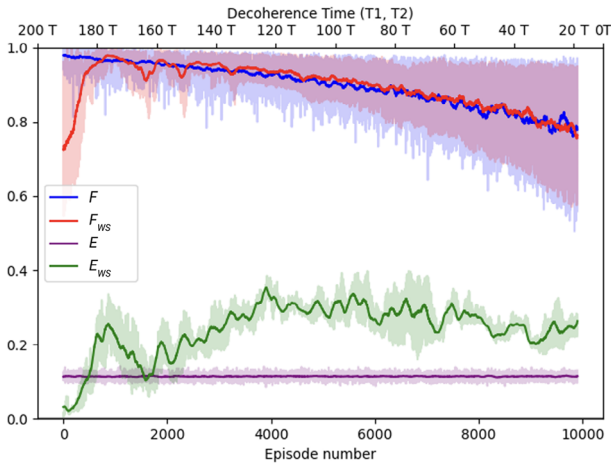


Figure 8. Fidelity (blue) and energetic cost (purple) of RL generated control pulses with warm start, and fidelity (red) and energetic cost (green) of RL generated control pulses without warm start, as a function of the training episode number and decreasing decoherence time. Parameters: $U_T = \text{Hadamard}$, $\hat{H}_D = \hat{H}_D^1$, $\hat{\sigma}_k \in \{\hat{\sigma}_x^1\}$, $T_1 = [200T, 1T]$, $T_2 = [200T, 1T]$, $w_f = 0.8$, $w_e = 0.2$, $N = 100$, $N_G = 500$, $\text{layer_params} = (200, 100, 50, 30, 10)$, $N_{QLA} = 10000$, $N_{GA} = 2000$.

Hadamard, T). In Appendix B, the effect of increasing the size of the neural network of the reinforcement learning agents on the performance is shown for a universal gate set.

5. Geodesic Analysis

In this section, we investigate the effect of the control pulses generated by the EO-GRAPE algorithm and the EO-DRLPE agents on the unitary paths on the Bloch sphere. As illustrated in figure 2, the optimal path on the Bloch sphere is directly proportional to the optimal energetic cost of the uni-

tary operation, conditioned on a fixed total evolution time. While the relation between geodesic and energetic cost has been previously investigated in its theoretical rigor,^[15] in this research, we explore the pragmatic geodesic achieved by algorithmic procedures of energy-optimized QOC, i.e., EO-GRAPE and EO-DRLPE. Co-optimizing the total evolution time guided by quantum speed limits^[16] along with fidelity and energetic cost for pragmatic quantum pulse engineering is left to future work.

Figure 9 shows the unitary path on the Bloch sphere by the control pulses generated by the EO-GRAPE algorithm (left) and the reinforcement learning agent (right). An initial state of $|\psi_i\rangle = |0\rangle$ (green vector) and a target unitary of $U_T = R_x(\pi/2)$, is used, resulting in a target state of $|\psi_T\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) \equiv |-i\rangle$ (orange vector). The control Hamiltonian operators are $\hat{\sigma}_k \in \{\hat{\sigma}_x^1, \hat{\sigma}_y^1\}$. To clearly visualize the geodesic induced by the control pulses, the drift Hamiltonian is turned off (i.e., set to identity) $\hat{H}_D = I_2$. One can see in Figure 9a that the path induced by the control pulses generated by the EO-GRAPE algorithm is much more smooth and straight than the path induced by the control pulses generated by the reinforcement learning agent depicted in Figure 9b. This is exemplary of how the reinforcement learning agent learns by reward, as seen by the random walk that the state vector travels before eventually arriving close to the target state.

In **Figure 10**, the unitary path on the Bloch sphere by the control pulses generated by the EO-GRAPE algorithm (left) and the reinforcement learning agent (right) are shown in the presence of a drift Hamiltonian $\hat{H}_D = \hbar\omega_1/2\hat{\sigma}_z$. The same initial state, target quantum unitary, and control Hamiltonian operators are used. One can see the effect of the drift Hamiltonian, causing the state vector to precess about the \hat{z} -axis at the qubit frequency ω_1 . Interestingly, one can see that the path induced by the control pulses generated by the EO-GRAPE algorithm first overshoots the target state and then rotates back up to reach the target state, while

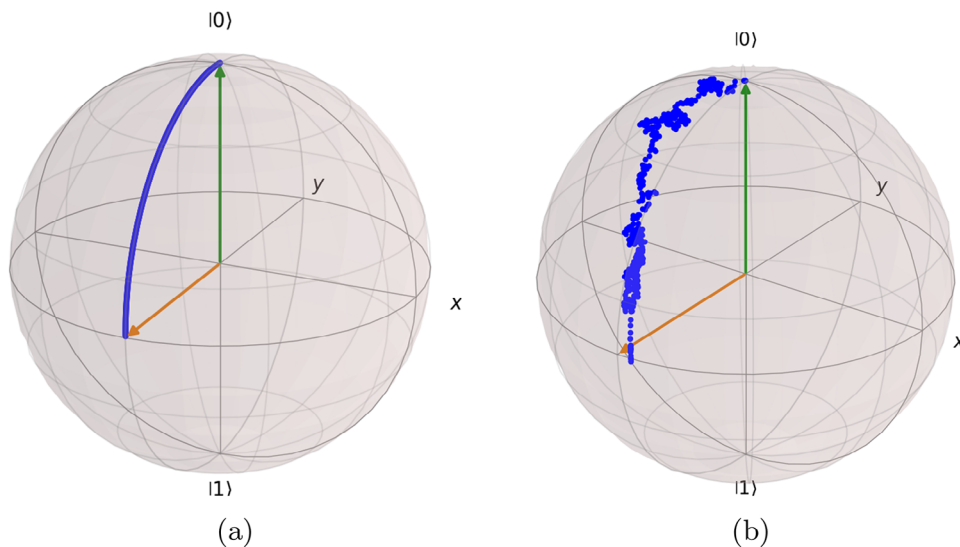


Figure 9. Initial state (green vector), target state (orange vector), and path of the quantum unitary (blue) in the absence of a drift Hamiltonian \hat{H}_D , by a) EO-GRAPE generated pulses and b) RL generated pulses on the Bloch Sphere. Parameters: $U_T = R_x(\pi/2)$, $\hat{H}_D = I_2$, $\hat{\sigma}_k \in \{\hat{\sigma}_x^1, \hat{\sigma}_y^1\}$, $T_1 = 1000T$, $T_2 = 1000T$, $w_f = 1$, $w_e = 0$, $N = 500$, $N_G = 500$, $N_{QLA} = 10000$, $|\psi_i\rangle = |0\rangle$.

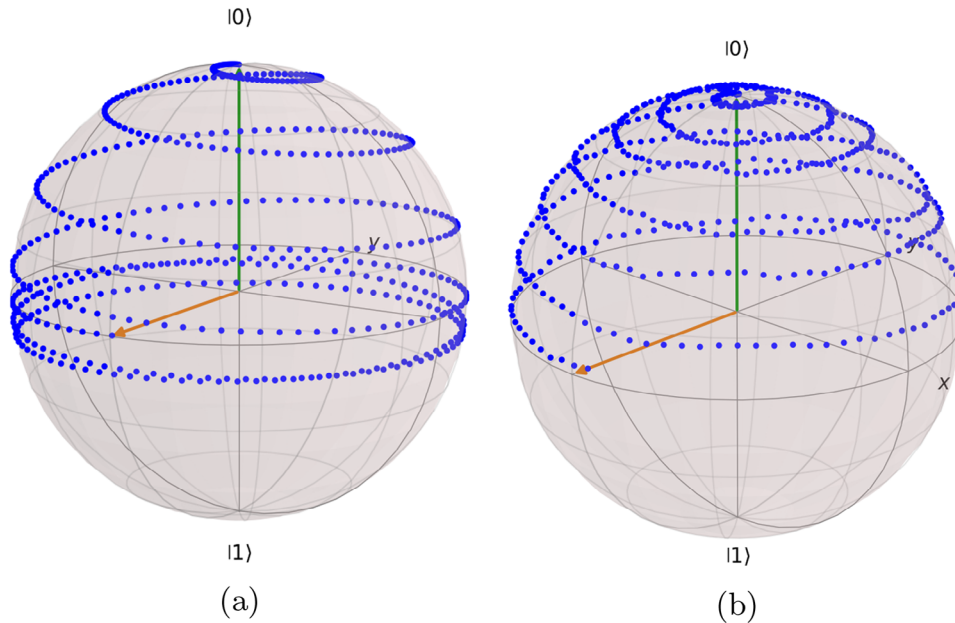


Figure 10. Initial state (green vector), target state (orange vector), and path of the quantum unitary (blue) in the presence of a drift Hamiltonian \hat{H}_D , by a) EO-GRAPE generated pulses and b) RL generated pulses on the Bloch Sphere. Parameters: $U_T = R_x(\pi/2)$, $\hat{H}_D = \hat{H}_D^1$, $\hat{\sigma}_k \in \{\hat{\sigma}_x^1, \hat{\sigma}_y^1\}$, $T_1 = 1000T$, $T_2 = 1000T$, $w_f = 1$, $w_e = 0$, $N = 500$, $N_G = 500$, $N_{QLA} = 10000$, $|\psi_i\rangle = |0\rangle$.

the path induced by the control pulses generated by the reinforcement learning agent does not overshoot and arrive at the target state in one time. We suspect that this is due to the restriction imposed by the architecture of the GRAPE algorithm on the energy difference between consecutive time slices of the pulse. This limitation can potentially be harnessed by EO-DRLPE to optimize the path length (and thus the energetic cost) further over that of EO-GRAPE. Note that the shortest accessible path is akin to the accessible geodesic^[44] based on the available control Hamiltonians and is not the shortest path on the Bloch sphere.

The theory presented in Ref. [15] suggests that the most energy-efficient quantum unitary is equivalent to a state vector traveling from the initial state to the target state via the geodesic between the two states on the Bloch sphere surface. In this scenario, the path induced by the reinforcement learning agent seems shorter than the path induced by the EO-GRAPE algorithm, suggesting that the energetic cost of the reinforcement learning agent control pulses is lower than the EO-GRAPE generated pulses in this specific scenario. To further investigate this, we explore the relation between the energetic cost and the path length of the unitary path on the Bloch sphere. In Figure 11, the correlation between the path length of the path on the Bloch sphere and the energetic cost of the control pulses of both EO-GRAPE generated pulses and RL generated pulses are shown. The path length was calculated by summing all the small line segments around the arc of the Bloch Sphere between each time step (blue dots in Figure 10).

One can see that for a weight setting of $w_e = 1$, $w_f = 0$, the pulses generated by the EO-GRAPE algorithm give a lower energetic cost due to the more structured nature of the optimizer. However, for lower weight settings of the energetic cost, the EO-

DRLPE is able to find pulses with a much shorter path length than the EO-GRAPE-generated pulses. The likely reason is that the RL agent is not restricted to keeping the pulse harmonics in shape.

As can be seen from the results presented in this subsection, the theory presented in Section 2.3 agrees with our findings. The shorter the path length on the Bloch sphere,

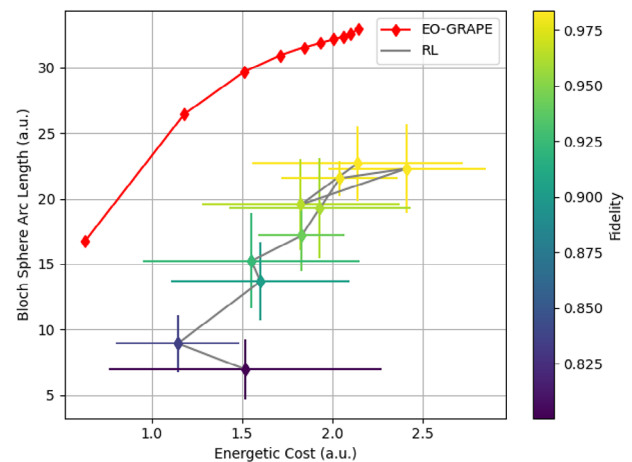


Figure 11. Combined plot of EO-GRAPE and EO-DRLPE generated pulses showing the correlation between energetic cost and path length of the unitary on the Bloch Sphere, where the color coding indicates the average fidelity of the control pulse. Each diamond marker corresponds to a different weight setting (in steps of 0.1, with lines joining them in ascending order of w_e). Parameters: $U_T = R_x(\pi/2)$, $\hat{H}_D = \hat{H}_D^1$, $\hat{\sigma}_k \in \{\hat{\sigma}_x^1, \hat{\sigma}_y^1\}$, $T_1 = 1000T$, $T_2 = 1000T$, $w_f = [1, 0.1]$, $w_e = [0.1, 0.9]$, $N = 100$, $N_{QLA} = 10000$, $|\psi_i\rangle = |0\rangle$.

the lower the energetic cost required to implement a quantum unitary gate. Furthermore, the control pulses generated by the EO-GRAPE algorithm take some form of accessible geodesics, as described by the theory in Ref. [15]. In our opinion, benchmarking quantum control techniques based on the path traversed on the Bloch sphere, or more generally, the quantum geometric tensor,^[45] is a promising research direction.

6. Conclusion

In this research, we focus on estimating the energetic cost of synthesizing a quantum unitary gate. In this regard, this research addresses the relation between fidelity and the energetic cost and presents two novel quantum optimal control strategies to co-optimize fidelity and the energetic cost for pulse engineering.

We show that the energetic cost of implementing a quantum unitary gate through discrete pulse level control can be quantified by integrating the norm of the total Hamiltonian required to implement a specific quantum unitary gate over the total gate duration. In addition to this, we have found that this energetic cost positively correlates to the path length of the quantum unitary on the Bloch sphere, supporting the theory that the most energy-efficient way to implement a quantum unitary gate is through the geodesic between two quantum states. We demonstrate an empirical Pareto frontier between the fidelity and the energetic cost required to implement a quantum unitary gate. Specifically, in our experimental setup, we show that a decrease in the energetic cost by 10% yields an increase in the infidelity of roughly 1% in the low infidelity range.

To develop a quantum optimal control strategy, we developed a novel cost function and gradient for the GRAPE method, allowing for co-optimization of both the fidelity and energetic cost of a quantum unitary gate. This novel algorithm is the energy-optimized gradient ascent pulse engineering (EO-GRAPE). Next to the gradient-based open-loop quantum optimal control method, we have also investigated a learning-based, model-free, closed-loop method. An energy-optimized deep reinforcement learning for pulse engineering (EO-DRLPE) architecture was developed to interact with a quantum environment and to learn control pulses that minimize both the energetic cost as well as the infidelity. Our results demonstrate that both optimal control methods perform relatively well in low-noise systems. When one decreases the relaxation and decoherence times of the qubits in the system, the EO-GRAPE algorithm outperforms the reinforcement learning agents for all noise settings and neural network sizes. Finally, a positive proportional relation between the path length of the quantum unitary on the Bloch sphere and the energetic cost of a control pulse sequence was observed, suggesting that the notion of energy efficiency and geodesics on the Bloch sphere is correct and could be leveraged to benchmark quantum control strategies. An open-sourced Python software package has been developed, which implements the methods discussed in this article.

More generally, this work demonstrates that one can co-optimize a quantum unitary gate on energy efficiency as well as fidelity by using quantum optimal control methods. The developed tools can be seamlessly integrated within a full-stack quan-

tum computing framework for co-design^[46,47] and automated optimization of quantum software layers.

6.1. Future Directions

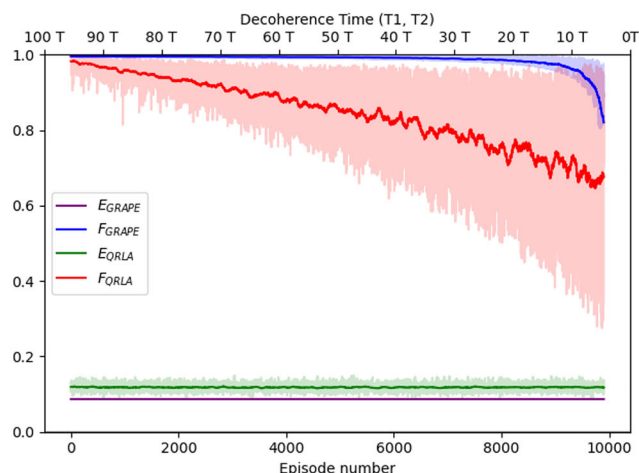
Here we outline some promising directions for extending the research presented in this article.

In this work, we have investigated co-optimizing control pulses on both fidelity and energetic cost for gates from the universal gate set of CNOT, Hadamard, and T. However, one could also train a reinforcement learning agent to learn and create other energy-optimal pulses that create new universal gate sets from a specific set of hardware restrictions, such as a specific waveform, set of frequencies, or bandwidth.^[48] This removes the restriction of specific quantum gate synthesis and allows the agent to optimize over the set of unitaries that together form a universal gate,^[49] and thereby any quantum computation, with a potentially lower energetic cost (than using CNOT, Hadamard, and T).

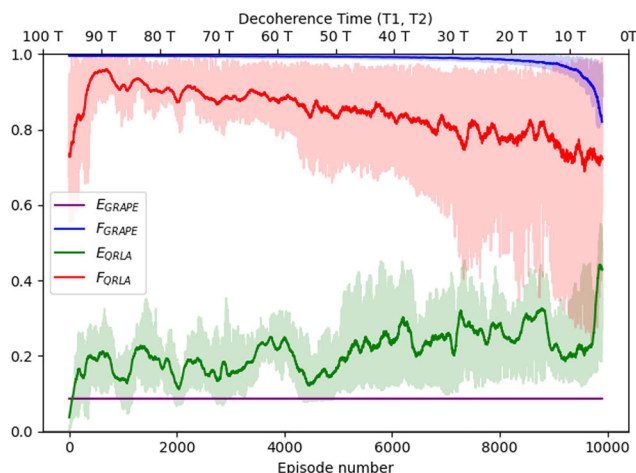
The algorithms developed in this work generate and train on control pulses that are represented as 2D matrices with dimensions (K, N) . If one has 3 control operators and 500 time steps, a control pulse already contains 1500 individual parameters that the algorithm and reinforcement learning agents need to adjust. However, as one can see from the pulses generated by the EO-GRAPE algorithm, the pulses can often be decomposed into individual sin or cos periodic functions. It is, therefore, plausible that one can transform the control pulses to the frequency domain by applying a Fourier transformation, or using alternate learning basis like Kolmogorov–Arnold networks.^[50] This would allow for representing a control pulse with just a few amplitude and frequency parameters.^[51] This would dramatically increase the computational efficiency of both algorithms and could also increase the performance of the reinforcement learning agent, as it would automatically apply harmonic pulses instead of random block pulses.

Different formulations of the drift and control Hamiltonian operators have been investigated in this research. However, the effect of having multiple redundant control operators has not been thoroughly researched. Minimizing the number of control operators per qubit to universally control the qubits is, therefore, a critical component in minimizing the energetic cost of a quantum unitary gate. We can, therefore, ask ourselves, given a multi-qubit system with drift Hamiltonian \hat{H}_D , how many individual control terms $|\hat{H}_k|$ does one need to control the quantum system universally? There exists a theoretical framework to address this problem of full controllability called the Lie rank test.^[25] One could, therefore, use this theoretical framework to minimize the number of control operators needed per qubit and per drift Hamiltonian and observe the effect that it has on the energetic cost of quantum unitary gates performed on the qubits.

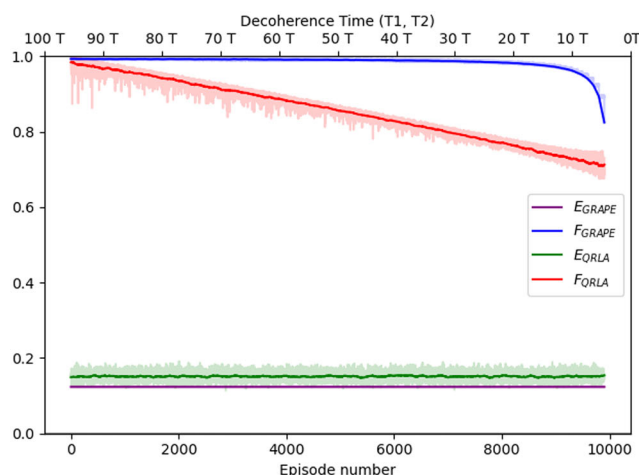
Information, entropy, and energy are closely related quantities. Landauer's principle states that the minimum energy required by a logic operation will be the temperature times the entropy.^[52] As one can see, the entropy (expressed as bits of information) and energy of a control sequence are closely related.^[53] Therefore, a similar investigation as the one performed in this research between fidelity and energy can be proposed for information and energy. One can investigate the co-optimization of both



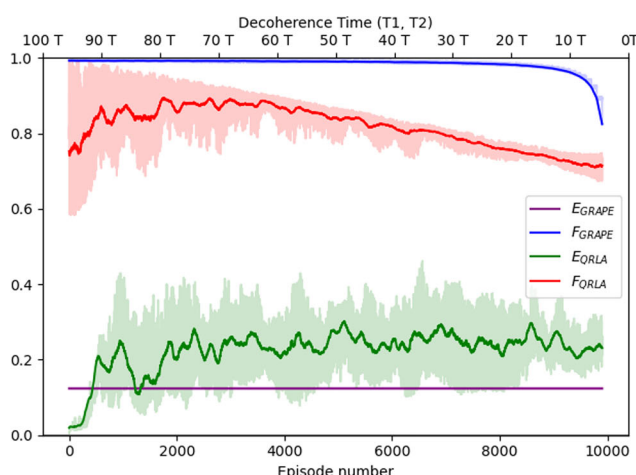
(a) Hadamard gate, with warm start



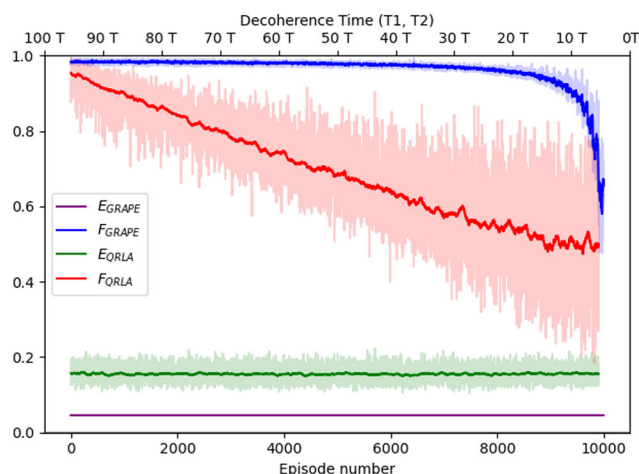
(b) Hadamard gate, without warm start



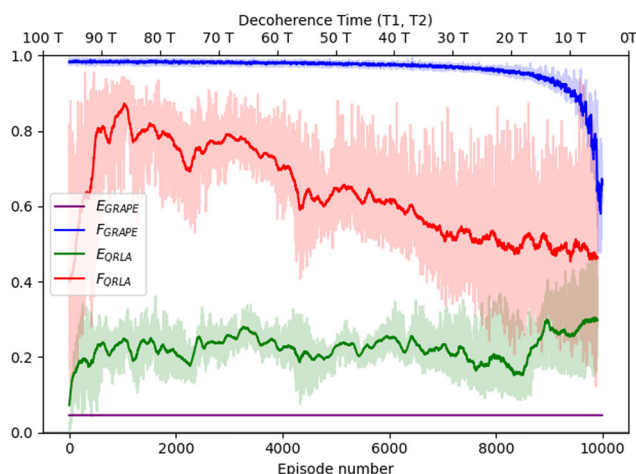
(c) T gate, with warm start



(d) T gate, without warm start

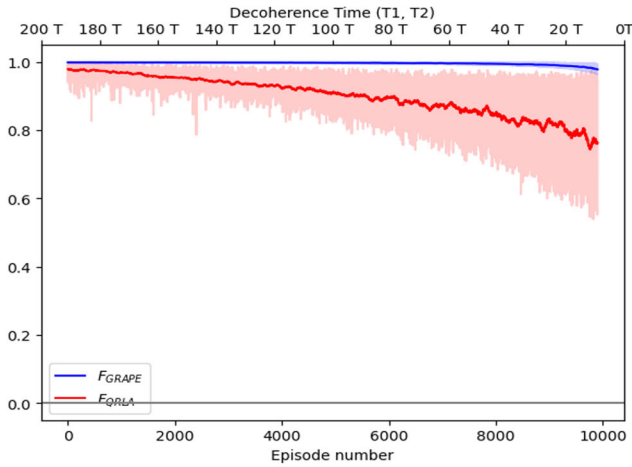


(e) CNOT gate, with warm start

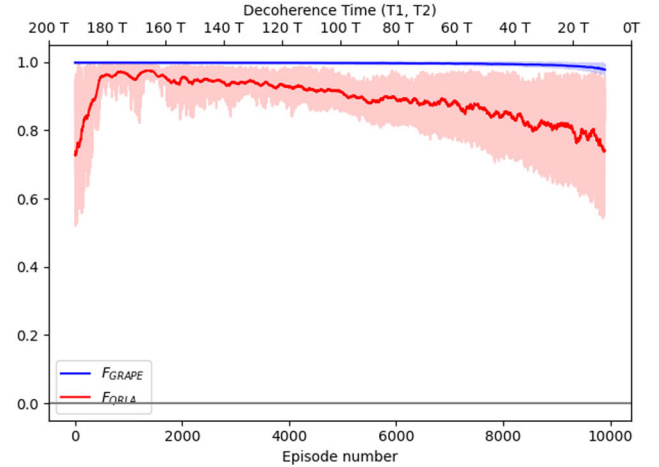


(f) CNOT gate, without warm start

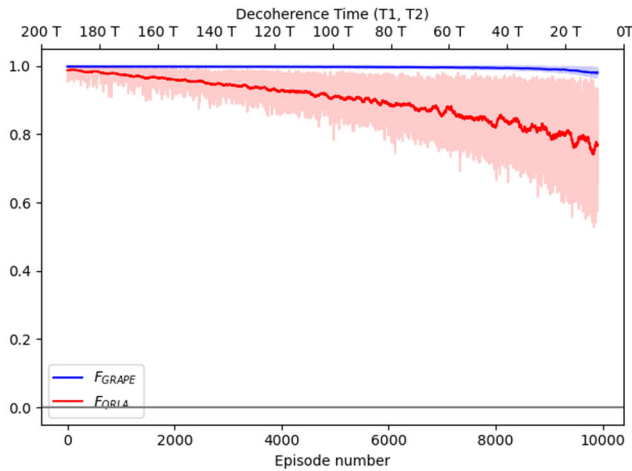
Figure A1. Fidelity (blue) and energetic cost (purple) of EO-GRAPe generated control pulses, and fidelity (red) and energetic cost (green) of RL generated control pulses, as a function of the training episode number and decreasing decoherence time, for a universal set of gates, and with and without warm start. Parameters: $U_T = \text{CNOT, Hadamard, and T}$; $\hat{H}_D = \hat{H}_D^2$ (for CNOT), \hat{H}_D^1 ; $H_k = \{\sigma_x^1, \sigma_x^2, \sigma_x^1 \sigma_x^2\}$ (for CNOT), $\{\sigma_x^1\}$; $T_1 = [100T, 1T]$; $T_2 = [100T, 1T]$; $w_f = 0.8$; $w_e = 0.2$; $N_t = 100$; $N_g = 500$; $\text{layer_params} = (200, 100, 50, 30, 10)$; $N_{QRLA} = 10,000$, $N_{GA} = 2,000$.



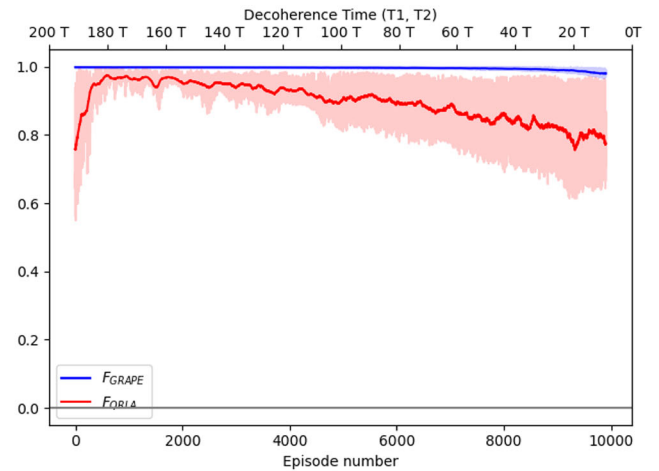
(a) NN Size 1, with warm start



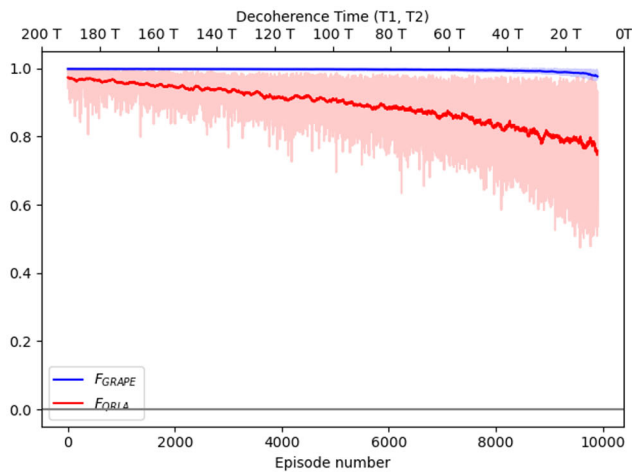
(b) NN Size 1, without warm start



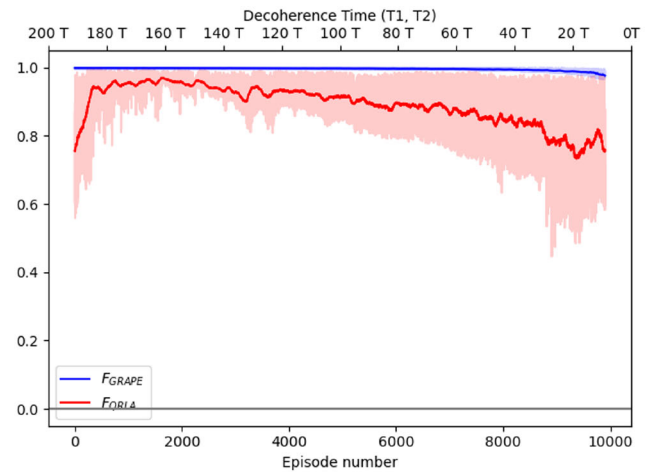
(c) NN Size 2, with warm start



(d) NN Size 2, without warm start



(e) NN Size 3, with warm start



(f) NN Size 3, without warm start

Figure B1. EO-GRAPE generated pulses fidelity (blue), and RL generated pulses fidelity (red), as a function of training episode number and decreasing decoherence time, for different neural network sizes and with and without warm start. Parameters: $U_T = \text{Hadamard}$, $\hat{H}_D = \hat{H}_D^1$, $H_k = \{\sigma_x^1\}$, $T_1 = [200T, 10T]$, $T_2 = [200T, 10T]$, $w_f = 1$, $w_e = 0$, $N_t = 100$, $N_g = 500$, $N_{QRLA} = 10,000$, $N_{GA} = 2,000$.

the information contained in a control sequence,^[54] as well as the energy of a control sequence, and see what the relation is between the two and if it is possible to co-optimize, similar to fidelity and energy. In theory, one could also change the fidelity part of the cost function presented in this work by measuring for information and repeating the same experiments that have been performed in this work.

In this work, we investigated closed quantum systems described by the Schrödinger equation. In reality, however, qubits experience decoherence and are thus actually open quantum systems. One can, therefore, propose to describe the qubit quantum states by the Lindblad master equation and investigate whether the proposed algorithms in this work still apply when noise is modeled as open-system dynamics.^[55,56]

Appendix A: Performance of EO-GRAPE and EO-DRLPE for a Universal Gate Set

To compare the EO-GRAPE algorithm performance to both the warm start and without warm start reinforcement learning agents, the fidelity and energetic cost of a universal gate set (CNOT, Hadamard, and T) using all three methods has been plotted as a function of increasing noise gain or decreasing relaxation and decoherence time.

In Figure A1, the results of these experiments are presented. One can see that the EO-GRAPE algorithm outperforms the warm start and without warm start reinforcement learning agents for all three target unitary gates and all noise settings. Interestingly, the variance in the performance of the reinforcement learning agent optimizing a T gate is much smaller than the variance with the CNOT gate or Hadamard gate as the target unitary gate. Next to this, one can see that the reinforcement learning agent with a warm start, in general, has a minor variance in performance compared to the reinforcement learning agent without a warm start and that the energetic cost of the pulses generated by the reinforcement learning agent with warm start is in general lower than the energetic cost of the pulses generated by the reinforcement learning agent without warm start.

From this figure, we can thus conclude that the reinforcement learning agent can learn a universal set of gates co-optimized on fidelity and energetic cost. Additionally, the EO-GRAPE algorithm outperforms both reinforcement learning agents regarding fidelity and energetic cost for all noise settings.

Appendix B: Effect of Neural Network Size on EO-DRLPE

To further explore the performance of the reinforcement learning agent compared to the EO-GRAPE algorithm, we have investigated the effect of the size of the neural network (NN) on the performance of the reinforcement learning agent. To make the effect more noticeable, the weight of energetic cost was set to zero $w_e = 0$, and the weights of the fidelity were set to one $w_f = 1$, i.e., the agent is only trying to optimize the pulse on fidelity, and not on energetic cost. The same experiments as in Figure A1 have been done, where the fidelity of both the EO-GRAPE algorithm and the reinforcement learning agents with and without warm start are plotted as a function of increasing noise or decreasing relaxation and decoherence time. The experiments all use the Hadamard gate as the target quantum unitary gate. They are plotted for three different neural network sizes, with and without a warm start and two different noise settings. The different neural network sizes used are: NN Size 1 = (200, 100, 50, 30, 10), NN Size 2 = (400, 200, 100, 50, 30, 10) and NN Size 3 = (600, 400, 200, 100, 50, 30, 10). The experiment results are displayed for a low noise setting, i.e., $T_1, T_2 \in [2007, 107]$.

In Figure B1, one can see that in a low noise environment, the EO-GRAPE algorithm outperforms the reinforcement learning agent, both with and without warm start, and for all three neural network sizes. One

can also see that increasing the neural network size has little effect on the performance of the reinforcement learning agent, apart from the variance getting slightly smaller.

Appendix C: Software Availability

The open-sourced code for the project, configuration files, output data, and plotting codes for the experiments presented in this article are available at: <https://github.com/QML-Group/EO-QCtrl>.

Acknowledgements

A didactic introduction and additional details of the implementation presented in this article can be found in the corresponding master thesis.^[57] The authors thank Ramon Overwater, Ilker Polat, and Yining Zhu for insightful discussion on the current state-of-the-art pulse control optimization for superconducting qubit processors in QuTech. A.S. acknowledges funding from the Dutch Research Council (NWO) through the project "QuTech Part III Application-based research" (project no. 601.QT.001 Part III-C—NISQ).

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are openly available in EO-QCtrl at <https://github.com/QML-Group/EO-QCtrl>, reference number 1.

Keywords

energy efficiency, pulse control, unitary synthesis

Received: December 16, 2024

Revised: March 25, 2025

Published online:

- [1] P. Benioff, *J. Stat. Phys.* **1980**, 22, 563.
- [2] D. Deutsch, *Proc. R. Soc. London A. Math. Phys. Sci.* **1985**, 400, 97.
- [3] E. Bernstein, U. Vazirani, in *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, **1993**, pp. 11–20.
- [4] R. P. Feynman, *Int. J. Theor. Phys.* **1982**.
- [5] P. W. Shor, *arXiv preprint arXiv:2208.09964* **2022**.
- [6] F. Leymann, J. Barzen, *Quantum Sci. Technol.* **2020**, 5, 044007.
- [7] O. Ezratty, *arXiv preprint arXiv:2305.09518* **2023**.
- [8] X. Waintal, *Proc. Natl. Acad. Sci.* **2024**, 121, e2313269120.
- [9] K. Bertels, A. Sarkar, T. Hubregtsen, M. Serrao, A. A. Mouedenne, A. Yadav, A. Krol, I. Ashraf, C. G. Almudever, *IEEE Transactions on Quantum Engineering* **2020**, 1, 1.
- [10] K. Bertels, A. Sarkar, A. Krol, R. Budhrani, J. Samadi, E. Geoffroy, J. Matos, R. Abreu, G. Gielen, I. Ashraf, *arXiv preprint arXiv:2102.02035* **2021**.
- [11] Y. Shi, P. Gokhale, P. Murali, J. M. Baker, C. Duckering, Y. Ding, N. C. Brown, C. Chamberland, A. Javadi-Abhari, A. W. Cross, D. I. Schuster, K. R. Brown, M. Martonosi, F. T. Chong, *Proc. IEEE* **2020**, 108, 1353.
- [12] QEI, *QEI* **2023**, <https://quantum-energy-initiative.org/> (accessed: April 2025).

- [13] M. P. Frank, *arXiv preprint arXiv:1803.02789* **2018**.
- [14] A. Benhari, D. Trystram, F. Dufossé, Y. Denneulin, F. Desprez, *arXiv preprint arXiv:2403.17466* **2024**.
- [15] S. Deffner, *Europhys. Lett.* **2021**, 134, 40002.
- [16] M. Aifer, S. Deffner, *New J. Phys.* **2022**, 24, 055002.
- [17] A. Auffèves, *PRX Quantum* **2022**, 3, 020101.
- [18] E. Schrödinger, *Phys. Rev.* **1926**, 28, 1049.
- [19] C. P. Koch, U. Boscain, T. Calarco, G. Dirr, S. Filipp, S. J. Glaser, R. Kosloff, S. Montangero, T. Schulte-Herbrüggen, D. Sugny, F. K. Wilhelm, *EPJ Quantum Technol.* **2022**, 9, 19.
- [20] S. J. Glaser, U. Boscain, T. Calarco, C. P. Koch, W. Köckenberger, R. Kosloff, I. Kuprov, B. Luy, S. Schirmer, T. Schulte-Herbrüggen, D. Sugny, F. K. Wilhelm, *Eur. Phys. J. D* **2015**, 69, 1.
- [21] T. S. Mahesh, P. Batra, M. H. Ram, *arXiv* **2022**.
- [22] U. Boscain, M. Sigalotti, D. Sugny, *PRX Quantum* **2021**, 2, 3.
- [23] F. Motzoi, J. M. Gambetta, P. Rebentrost, F. K. Wilhelm, *Phys. Rev. Lett.* **2009**, 103, 110501.
- [24] A. Ma, A. B. Magann, T.-S. Ho, H. Rabitz, *Phys. Rev. A* **2020**, 102, 013115.
- [25] A. Ibort, J. M. Pérez-Pardo, *J. Phys. A: Math. Theor.* **2009**, 42, 205301.
- [26] M. Y. Niu, S. Boixo, V. Smelyanskiy, H. Neven, *arXiv* **2018**.
- [27] D. Dong, J. Tang, Q. Yu, P. Girdhar, O. Shindi, *UNSW Library* **2023**.
- [28] D. Dong, M. A. Mabrok, I. R. Petersen, B. Qi, C. Chen, H. Rabitz, *IEEE Trans. Control Syst. Technol.* **2015**, 23, 2155.
- [29] M. E. Fonseca, F. F. Fanchini, E. F. de Lima, L. K. Castelano, *arXiv* **2023**.
- [30] S. Machnes, E. Assémat, D. Tannor, F. K. Wilhelm, *Phys. Rev. Lett.* **2018**, 120, 150401.
- [31] N. Khaneja, T. O. Reiss, C. Kehlet, T. Schulte-Herbrüggen, S. J. Glaser, *J. Magn. Reson.* **2005**, 172, 296.
- [32] T. Caneva, T. Calarco, S. Montangero, *Phys. Rev. A* **2011**, 84, 2.
- [33] J. Brown, M. Paternostro, A. Ferraro, *Quantum Sci. Technol.* **2023**, 8, 025004.
- [34] S. Hu, H. Ma, D. Dong, C. Chen, *J. Franklin Inst.* **2023**, 360, 13972.
- [35] S. Günther, N. A. Petersson, *arXiv* **2023**.
- [36] M. E. Fernandes, E. F. de Lima, L. K. Castelano, *arXiv preprint arXiv:2110.13852* **2021**.
- [37] L. K. Castelano, E. F. de Lima, J. R. Madureira, M. H. Degani, M. Z. Maialle, *Phys. Rev. B* **2018**, 97, 235301.
- [38] T. Van Vu, T. Kuwahara, K. Saito, *Phys. Rev. Research* **2024**, 6, 033225.
- [39] J. T. Merrill, K. R. Brown, *Quantum Information and Computation for Chemistry* **2014**, 241.
- [40] J. Ash, R. P. Adams, *Adv. Neural. Inf. Process. Syst.* **2020**, 33, 3884.
- [41] D. Jayakody, Reinforce - a quick introduction (with code), **2023**, <https://dilithjay.com/blog/reinforce-a-quick-introduction-with-code> (accessed: April 2025).
- [42] M. Abadi, TensorFlow: Large-scale machine learning on heterogeneous systems, **2015**, <https://www.tensorflow.org/>, Software available from [tensorflow.org](https://www.tensorflow.org/) (accessed: April 2025).
- [43] J. Johansson, P. Nation, F. Nori, *Comput. Phys. Commun.* **2012**, 183, 1760.
- [44] M. Bukov, D. Sels, A. Polkovnikov, *Phys. Rev. X* **2019**, 9, 011034.
- [45] C. V. Meinersen, S. Bosco, M. Rimbach-Russ, *arXiv preprint arXiv:2409.03084* **2024**.
- [46] P. Gokhale, A. Javadi-Abhari, N. Earnest, Y. Shi, F. T. Chong, in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, IEEE, **2020**, pp. 186–200.
- [47] N. Earnest, C. Tornow, D. J. Egger, *Phys. Rev. Research* **2021**, 3, 043088.
- [48] M. Abdelhafez, B. Baker, A. Gyenis, P. Mundada, A. A. Houck, D. Schuster, J. Koch, *Phys. Rev. A* **2020**, 101, 022321.
- [49] A. Sarkar, A. Kundu, M. Steinberg, S. Mishra, S. Fauquenot, T. Acharya, J. A. Miszczak, S. Feld, *arXiv preprint arXiv:2406.17610* **2024**.
- [50] A. Kundu, A. Sarkar, A. Sadhu, *EPJ Quantum Technol.* **2024**, 11, 76.
- [51] D. Dong, C.-C. Shu, J. Chen, X. Xing, H. Ma, Y. Guo, H. Rabitz, *IEEE Trans. Control Syst. Technol.* **2021**, 29, 1791.
- [52] T. Van Vu, K. Saito, *Phys. Rev. Letters* **2022**, 128, 1.
- [53] S. Haseli, H. Dolatkah, S. Salimi, A. S. Khorashad, *Laser Phys. Lett.* **2019**, 16, 045207.
- [54] S. Mishra, A. Sarkar, S. Feld, *preprint* (in preparation).
- [55] F. Nathan, M. S. Rudner, *Phys. Rev. B* **2020**, 102, 11.
- [56] P. A. Erdman, R. Czupryniak, B. Bhandari, A. N. Jordan, F. Noé, J. Eisert, G. Guarnieri, *arXiv preprint arXiv:2408.15328* **2024**.
- [57] S. Y. Fauquenot, Master's thesis, Delft University of Technology, **2024**, <https://resolver.tudelft.nl/uuid:d115c6c2-329f-47ed-9791-2ef6b1f8c359>.