Quantum Information Processing (2025) 24:175
https://doi.org/10.1007/s11128-025-04791-w

Quantum gate control pulse optimization based on the Adam algorithm

Mengdi Yang[1] · Feng Yue[1] · Bo Lu[1] · Hanshi Zhao[1] · Geyuyan Ma[1] · Lixin Wang[1]

Received: 4 February 2025 / Accepted: 19 May 2025

Abstract

The efficient implementation of quantum computing is contingent upon high-fidelity quantum operations. Nevertheless, the fidelity of these operations is constrained by the precision of quantum system evolution control. The optimization of quantum control pulses is essential for improving the manipulation accuracy of superconducting qubits. Traditional optimization methods, including gradient descent and the gradient ascent pulse engineering algorithm, frequently encounter challenges such as slow convergence and susceptibility to local optima in pulse optimization problems. This study introduces an adaptive open-loop optimization algorithm based on the adaptive moment estimation (Adam) optimizer, capitalizing on the benefits of momentum and adaptive learning rate adjustments inherent in Adam. We conduct experimental analysis on the algorithm's hyperparameters to achieve optimal solutions with a higher fidelity range at a faster convergence speed, effectively improving the fidelity and optimization efficiency of quantum gate operations. Through numerical simulations on the QuTiP platform, we validate the excellent performance of the Adam algorithm in quantum gate optimization. In optimizing the X and SWAP gates, its fidelity improved by 0.03% and 0.0016%, respectively, compared to the GRAPE algorithm. Compared to the CRAB method, the initial convergence speed of the Adam method increased fivefold, enabling it to achieve the target fidelity more rapidly. Future research will

✉ Lixin Wang
wlx715@163.com

Mengdi Yang
ymd2022@foxmail.com

Feng Yue
yf_hpc@163.com

Bo Lu
lubo@mail.bnu.edu.cn

Hanshi Zhao
zhs_IEU@163.com

Geyuyan Ma
mgyy_quantum@163.com

[1] Information Engineering University, 62 Kexue Avenue, Zhengzhou 450001, Henan, China

Published online: 12 June 2025

investigate closed-loop optimization strategies, utilizing feedback derived from the fidelity results of actual quantum computers to further augment quantum control performance

Keywords GRAPE · Quantum gate · Adam algorithm · Pulse optimization

1 Introduction

In the context of rapid technological advancements today, the importance of quantum computing is increasingly prominent, with its influence extending far beyond the field of computation itself, reaching into disciplines such as materials science [1], drug design [2], and cryptography [3]. The core advantage of quantum computing lies in its tremendous potential to tackle complex problems, primarily due to the superposition and entanglement properties of qubits. However, quantum systems are susceptible to environmental noise and interactions with the environment during their evolution, leading to decoherence of quantum states and deviations in the evolution path, thereby affecting the precision of quantum computations. This has become a critical bottleneck in achieving efficient quantum computing. Consequently, quantum optimal control theory has emerged, aiming to precisely design and manipulate the evolution paths of quantum systems to achieve specific quantum state transformations or quantum gate operations, thereby enhancing the fidelity of quantum operations [4–6].

In superconducting quantum systems, the optimization of quantum control pulses is key to improving the precision of qubit manipulation. In superconducting quantum systems, the optimization of quantum control pulses is key to improving the precision of qubit manipulation. Traditional optimization methods, such as the gradient descent algorithm, iteratively adjust control parameters by moving in the opposite direction of the gradient of the objective function to minimize it, thereby reducing the distance between the quantum state and the target state [7]. This algorithm has better convergence, and thus, most quantum optimal control algorithms are optimized based on gradient descent. Khaneja et al. proposed the Gradient Ascent Pulse Engineering (GRAPE) algorithm for designing nuclear magnetic resonance pulse sequences. This algorithm approximates control functions with piecewise constant functions and derives explicit derivatives, effectively balancing the speed and memory usage in computing fidelity, making it suitable for designing idealized pulse sequences. In the TorchQC framework proposed by Dimitris Koutromanos in 2025, researchers further integrated the GRAPE algorithm with PyTorch's automatic differentiation (AD) capability, achieving enhanced quantum gate fidelity through GPU acceleration [8]. Building on the work in reference [9], Larocca and Wisniacki proposed the K-GRAPE algorithm, which uses the Krylov subspace to estimate quantum states during time evolution [10]. Another class of quantum optimal control algorithms is based on the chopped random basis (CRAB) optimal control technique, which describes the control space through a series of basis functions, parameterizes control waveforms, and optimizes over randomly chosen basis functions, effectively avoiding local minima in high-dimensional parameter spaces, suitable for quantum systems with complex control requirements [11, 12]. The Krotov algorithm is a gradient descent method suitable for time-continuous control fields, ensuring monotonic fidelity increase by gradually adjusting control parameters at each time step [13]. The Gradient Optimization for Analytic Controls (GOAT) is a gradient optimization method based on analytic controls, particularly suitable for complex quantum systems [14]. It is noteworthy that existing studies exhibit significant methodological divergence in gradient-based optimization approaches: In 2019, Mohamed et al. addressed the high computational complexity of open quantum systems by developing a gradient optimization method combining quantum trajectories with automatic differentiation [15]. In 2022, Michael's team tackled the challenge of non-analytic objective functions by creating a semi-automatic differentiation (Semi-AD) technique that hybridizes analytical gradients with automatic differentiation [16]. These seminal works have established important references for control optimization in closed quantum systems. However, the evolution of quantum systems is often described by nonlinear equations, and the nonlinear nature of these equations makes the relationship between quantum state evolution and quantum gate fidelity control parameters complex, leading to non-convexity in the objective function for optimization. Non-convex optimization can have multiple local minima, and its solution space is complex, resulting in a slow optimization process. Therefore, the aforementioned methods often face challenges such as slow convergence and susceptibility to local optima when dealing with non-convex optimization problems. The complexity and noise of quantum systems also increase the difficulty of optimization, making traditional methods seem inadequate in addressing these challenges, thus necessitating the integration of optimization algorithms to improve the efficiency of pulse optimization. In recent years, the Adaptive Moment Estimation (Adam) optimization algorithm has gained widespread attention for its efficiency and stability in handling large-scale and complex optimization problems [17]. Although initially designed for optimizing deep neural networks, the Adam algorithm is also applicable to quantum pulse control. The evolution of quantum systems is typically described by nonlinear Schrödinger equations, making the relationship between quantum state evolution and quantum gate fidelity control parameters extremely complex. In this context, the Adam algorithm, by combining momentum and adaptive learning rate adjustments, can accelerate convergence without significantly increasing computational costs. This paper proposes an adaptive open-loop optimization algorithm for quantum control, leveraging the advantages of Adam to enhance the efficiency and accuracy of waveform optimization. Unlike traditional optimization algorithms, the Adam algorithm utilizes reverse evolution from the target state to the initial state and decomposes the optimization problem into low-dimensional subsystems, quickly achieving the target fidelity of quantum gates. Our numerical simulations demonstrate that the application of the Adam algorithm in quantum systems significantly improves optimization speed and quantum gate fidelity.

The structure of this paper is as follows: Chapter 2 introduces the theoretical foundation of quantum systems, including Hamiltonian descriptions and quantum control objectives. Chapter 3 provides a detailed description of the Adam algorithm and its implementation. Chapter 4 presents numerical simulations and experimental validations on quantum systems. Finally, Chapter 5 discusses potential improvements and future applications of the Adam algorithm.

2 Background

2.1 Mathematical model of quantum control

In quantum computing, the dynamics of a quantum system's evolution are determined by its Hamiltonian and initial state. The Hamiltonian is a self-adjoint operator that describes the system's energy. According to the Schrödinger equation $i\hbar\frac{d}{dt}|\psi(t)\rangle = H|\psi(t)\rangle$ [18], the time evolution of a quantum state can be expressed as follows. For the density matrix, the system's evolution follows the Liouville–von Neumann equation $i\hbar\frac{d}{dt}\rho(t) = [H, \rho(t)]$ [19], where $[H, \rho(t)]$ denotes the commutator of the Hamiltonian and the density matrix. The state of a quantum system can be described by a state vector or a density matrix. The state vector is typically used to describe pure states and is a unit vector in Hilbert space, usually denoted as $|\psi\rangle$. For an n-qubit system, the state vector can be represented as a complex vector of dimension. However, in practical quantum computing, due to environmental noise and interactions between the system and the environment, quantum systems often exist in mixed states, in which case the density matrix is used ρ to describe the system state. The density matrix is a positive-definite, self-adjoint matrix that satisfies $Tr(\rho) = 1$. For pure states, the density matrix can be expressed as $\rho = |\psi\rangle\langle\psi|$. Noise can be represented using non-unitary Kraus operators [20]. For example, in a transmon qubit, we can divide the Hamiltonian into a time-dependent drift Hamiltonian and a controllable Hamiltonian, setting the single-qubit Hamiltonian as $H = \frac{\omega}{2}\sigma_z + \Omega(t)\sigma_x + \Delta(t)\sigma_y$, where ω is the intrinsic frequency of the qubit, usually related to the energy level difference of the qubit; $\sigma_z, \sigma_x, \sigma_y$ is the Pauli matrix; $\Omega(t)$ is the amplitude of the driving field, typically a time-dependent function used to control the state of the qubit; and $\Delta(t)$ is the phase of the driving field, also usually a time-dependent function. For multi-qubit systems, the Hamiltonian becomes more complex as it includes not only single-qubit terms but also terms describing interactions between qubits. In a two-qubit system, the Hamiltonian can be expressed as $H = H_1 \otimes I + I \otimes H_2 + H_{\text{int}}$, where H_1 and H_2 describe the single-qubit Hamiltonians of the two qubits, and H_{int} describes the interaction between the two qubits [21]. Precise control of the Hamiltonian allows for accurate manipulation of quantum states, which is the core task of pulse optimization in quantum computing.

The specific optimization steps are as follows. First, we need to define a physical model of system dynamics $H(t) = H_d + \sum(u_1(t)H_{c_1} + u_2(t)H_{c_2} + \ldots)$, which can be described by a composite Hamiltonian . $H(t)$ is the time-dependent Hamiltonian, H_{c_1} is the constant part known as the drift Hamiltonian, and H_d is the time-varying part known as the control Hamiltonian, scaled by control amplitude functions $u_j(t)$. Next, by adjusting control parameters such as the amplitude $\Omega(t)$ and phase $\Delta(t)$ of the driving field, the target transformation of the quantum state is achieved. Optimization of control parameters typically involves defining an objective function that quantifies the difference between the current state of the system and the target state. Common objective functions include state fidelity $F(\rho, \rho_{\text{target}}) = \text{Tr}\left(\sqrt{\sqrt{\rho}\rho_{\text{target}}\sqrt{\rho}}\right)$, energy minimization $F(\rho, \rho_{\text{target}}) = \text{Tr}\left(\sqrt{\sqrt{\rho}\rho_{\text{target}}\sqrt{\rho}}\right)$, etc. The waveform optimization problem can be defined as minimizing the objective function while satisfying the system's dynamic constraints. Mathematically, this can be expressed as $\min_{u(t)} \||\psi_f\rangle - U(T, 0, u(t))|\psi_0\rangle\|^2$, where $|\psi_f\rangle$ is the target state, $|\psi_0\rangle$is the initial state, and $U(T, 0, u(t))$ is the evolution operator determined by the control parameters $u(t)$. Constraints usually include physical limits on control parameters, such as bounds on amplitude and phase [22, 23]. The numerical simulations in this work are based on a closed quantum system framework, neglecting environmental noise and non-unitary evolution effects (e.g., decoherence or relaxation). This idealized assumption enables us to focus on comparative performance evaluation of optimization algorithms under controlled conditions, while future studies will extend this framework to open quantum systems.

2.2 GRAPE pulse optimization method

In quantum control waveform optimization, gradient-based optimization methods are key technologies for achieving high-fidelity quantum operations. The GRAPE algorithm is one of the most representative methods. This algorithm discretizes the pulse sequence and optimizes the control parameters at each discrete time point to maximize the fidelity function. The core idea of the GRAPE algorithm is to compute the gradient of the objective metric with respect to the control parameters to determine the direction of adjustment for each parameter. In each iteration, parameter values are updated based on gradient information, gradually improving the target fidelity. This process is repeated until the desired convergence condition is met. Figure 1 illustrates the optimization process of the GRAPE algorithm. For a single-qubit system, its Hamiltonian can be expressed as $H = H_0 + \sum_k H_k$, where H_k is the internal coupling Hamiltonian of the system, and H_k is the external control Hamiltonian. By discretizing the time function of the control Hamiltonian into multiple control vectors and updating the parameter values in these vectors, optimization of the target fidelity function can be achieved. On the basis of the GRAPE algorithm, second-order optimization methods such as the BFGS (Broyden–Fletcher–Goldfarb–Shanno) algorithm have been introduced to accelerate convergence and improve optimization stability [23, 24]. The BFGS algorithm uses second-order derivative information (Hessian matrix) of the objective function to update control parameters. Specific steps include: initializing control parameters and control waveforms, calculating the objective function under the current waveform, using the GRAPE method to compute the gradient, estimating or calculating the Hessian matrix, and finally updating parameters using the BFGS algorithm. These steps are repeated until convergence conditions or a predetermined number of optimizations are reached.

2.3 Adam optimizer

Adam is an adaptive learning rate optimization algorithm initially proposed by Diederik P. Kingma and Jimmy Ba in 2014. It was designed to achieve fast and stable convergence in complex optimization problems, making it particularly suitable for scenarios requiring efficient optimization [17]. Adam combines the advantages of the

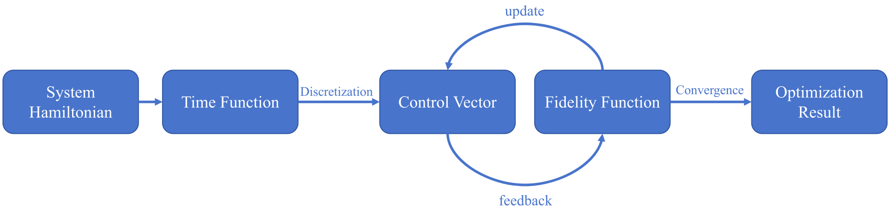

Fig. 1 Optimization process of the GRAPE

momentum method and the RMSProp algorithm, effectively accelerating the convergence process through adaptive learning rate adjustments for each parameter. Its core mechanism involves using estimates of the first and second moments to dynamically adjust the learning rate. The first moment (i.e., the exponentially weighted average of the gradient) accumulates historical information about the gradient, similar to the concept of momentum in physics, allowing the optimization process to consider the direction and magnitude of previous gradients when updating parameters, thereby accelerating convergence and reducing oscillations. The second moment (i.e., the exponentially weighted average of the squared gradient) accumulates historical information about the squared gradient, measuring the degree of change in the gradient. By adaptively adjusting the learning rate, it helps the optimizer better adapt to the gradient changes of different parameters, thereby improving the stability and efficiency of optimization. In deep learning, the Adam algorithm is widely used to train various neural network models, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and variational autoencoders (VAEs) [25–27]. It performs excellently when handling large-scale datasets and complex models. For example, in image classification tasks, Adam can quickly adjust network weights, enabling the model to achieve high accuracy in fewer iterations [28, 29]. In natural language processing tasks, Adam can efficiently optimize the parameters of language models, enhancing the performance of text generation and understanding [30, 31].

3 Pulse optimization algorithm based on Adam

In quantum control pulse optimization, the Adam algorithm iteratively optimizes control parameters, gradually reducing the gap between the quantum system's evolution and the target state, thereby finding better control strategies in complex quantum systems. This process first involves selecting the time evolution, determining the total evolution time, and dividing the time axis of the control amplitude into multiple time slices. Next, it requires initializing the waveform sequence, including setting initial waveform parameters, configuring algorithm hyperparameters, and generating the initial waveform sequence. The hyperparameters that affect algorithm performance mainly include step size, momentum factors, and the number of iterations. Since these hyperparameters fall under the category of experimental design rather than the core part of the algorithm logic, they will be discussed in detail in Chapter 4 and not in Algorithm 1. Then, starting from an initial value α of a waveform, a gradient-driven search is initiated. In each iteration, the Adam algorithm updates the first and second moments and calculates the infidelity error between the target gate and the actual gate. Infidelity is a metric that measures the difference between the target unitary transformation and the actual evolution, usually defined as Infidelity = 1 - f. Finally, the optimization process is terminated by setting stopping conditions, such as reaching a specified number of iterations or meeting convergence conditions. For different quantum gates, we set different control Hamiltonians and drift Hamiltonians according to the specific needs of the system to achieve optimal control effects. The overall algorithm flow is shown in Algorithm 1.

Algorithm 1 Pulse Optimization Algorithm Based on Adam

```
Require: Initial pulse sequence parameters $u$; time slices $t_n$; number of iterations $R$; evolution operator
    $U(t)$ of the quantum system
Ensure: Optimized control parameters $u$; final fidelity $F$
Evolution:
Initialize control parameters $u$
Initialize first moments $m$ and second moments $v$ to 0
Pulse amplitude constraints: $u_{\min} \leq u \leq u_{\max}$
Set iteration counter $r = 0$
while $r \leq R$ and stopping condition not met do
    Initialize the list of backward propagation operators $\mathbf{U}_b$ and forward propagation operators $U_f$
    for each time slice do
        Compute backward propagation operator: $U_{b\_list}[m] \rightarrow U_{b\_list}[0]$
        Compute the backward state of time: $P = U_{b\_list}[m] \cdot U$
        Compute the current Hamiltonian: $H_{ops}[n]$
        Compute the forward state of time: $Q = 1j \cdot dt \cdot H_{ops}[j] \cdot U_{f\_list}[m]$
        Compute gradient: $du = -2 \cdot \text{overlap}(P, Q) \cdot \text{overlap}(U_{f\_list}[m], P)$
        Update $m$ and $v$
    end for
    Update control parameters $u$ using bias-corrected momentum and variance
    Compute the fidelity of the current iteration: $F_r$
    Check if stopping conditions are met
    $r = r + 1$
end while
return Optimized control parameters $u$ and final fidelity $F$
```

When performing pulse gradient evolution, it is first necessary to determine the total evolution time and divide the time axis of the control amplitude into N slices. Choosing an appropriate number of time slots and total evolution time is crucial because it is necessary to ensure that the duration of each time slot is sufficiently small relative to the system's dynamics to guarantee the accuracy of the evolution approximation. Under this division, the control amplitude is considered constant within each time slot, so the evolution within each time slot can be calculated using $U(t_k) = \exp(-iH(t_k)dt)$, where dt is the duration of the time slot. By combining the evolutions of all time slots, the evolution at the final time $U(T)$ can be approximately obtained from the identity transformation $U(0)$ at the initial moment $t = 0$. This method ensures that the evolution approximation within segmented time slots is accurate, thereby effectively describing the overall evolution process of the system.

3.1 Initialization of waveform sequence

In the Adam algorithm, we need to prepare two related parameters for the waveform: one is the iterative parameter of the waveform, and the other is the initial waveform sequence. The iteration of the waveform is controlled by three parameters: R, J, and M, where R is the number of iterations, J is the number of pulse divisions, and M is the number of controllable Hamiltonians. For the waveform sequence, we can set different initial waveforms u_0 on the control Hamiltonians according to the actual situation of the quantum system and impose constraints on the waveform amplitude. The amplitude constraints of the waveform are mainly related to the physical system and need to be referenced to the actual situation. Random waveforms or periodic waveforms, such as periodic sine waves, periodic cosine waves, and frequency-doubled sine waves, can also be generated. We first initialize the iterative parameters of the waveform and perform calculations with the generated waveform sequence to produce the waveform form that needs algorithm optimization.

3.2 Iteration and update

In each iteration, the Adam optimizer uses momentum and variance matrices to update the control pulses. The momentum matrix m_t is used to accumulate historical information about the gradient, similar to the concept of momentum in physics. It allows the optimization process to consider the direction and magnitude of previous gradients when updating parameters, thereby accelerating convergence and reducing oscillations. In each iteration, the momentum matrix is updated using the formula $m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$, where β_1 is the decay rate of the first moment, and g_t is the current gradient. The update of m_t is an exponentially weighted average process, allowing earlier gradient information to gradually decay over time. The variance matrix v_t is used to accumulate historical information about the squared gradient, measuring the degree of change in the gradient. In contrast, the GRAPE method typically uses a fixed learning rate that needs to be manually adjusted to suit different optimization problems. The Adam optimizer can adaptively adjust the learning rate, and the variance matrix v_t helps the optimizer better adapt to the gradient changes of different parameters, thereby improving the stability and efficiency of optimization. In each iteration, the variance matrix v_t is updated using the formula $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$, where β_2 is the decay rate of the second moment, and g_t^2 is the square of the real part of the current gradient. The update of v_t is also an exponentially weighted average process. Specifically, the update process is as follows:

First, compute the bias-corrected momentum and variance. To make the updates of the momentum and variance matrices more stable in the initial stage, they need to be bias-corrected. The bias-corrected momentum and variance are calculated as follows: $\hat{m}_t = \frac{m_t}{1-\beta_1^t}$ and $\hat{v}_t = \frac{v_t}{1-\beta_2^t}$. Next, update the control pulses. Using the bias-corrected momentum $\hat{m}_t$ and variance $\hat{v}_t$, along with the learning rate α and a small constant ϵ to prevent division by zero, update the control pulses: $u_t = u_{t-1} - \frac{\alpha \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$.

3.3 Setting termination conditions

We calculate the fidelity of quantum gate operations through quantum process tomography, where fidelity can be viewed as the similarity between the ideal quantum process and the actual quantum process [32, 33]. Calculating fidelity involves computing the ideal Choi matrix and the actual Choi matrix. In a unitary system, the value of fidelity ranges between 0 and 1, with fidelity closer to 1 indicating greater similarity between the two operations. In experiments, we first construct the Choi matrix of the ideal quantum gate and then compute it with the Choi matrix of the actual quantum process to obtain the Frobenius distance between the two matrices, thereby calculating the infidelity. The goal of the optimization process is to minimize infidelity. When the fidelity results after iterations tend to stabilize, the optimization can be considered complete.

3.4 Factors affecting performance

In the Adam algorithm, there are three important factors: the step size (learning rate), momentum factors β_1 and β_2, and the number of iterations. During the optimization process, the step size functions similarly to the learning rate α, determining the magnitude of parameter updates in each iteration. A larger step size may lead to faster convergence but also increases the risk of oscillations near local minima. Conversely, a smaller step size may result in slower convergence but allows for a more precise approach to the optimal solution. By adjusting the step size, a balance can be achieved between the convergence speed and stability of the algorithm. An appropriate step size ensures that the algorithm maintains a fast convergence speed while avoiding oscillations near local minima. Although the Adam optimizer can adaptively adjust the step size based on gradient changes, the initial step size remains a crucial factor in determining optimization efficiency.

4 Simulation verification

To verify the advantages of the Adam algorithm, we designed and implemented a series of experiments focusing on the three hyperparameters. Each experiment was run under the same initial conditions, including the same initial waveform and target state. We conducted the experiments using the Quantum Toolbox in Python platform (QuTiP) platform without considering noise. QuTiP is an open-source Python library for simulating quantum systems, offering various pulse optimization algorithms and solvers capable of simulating the evolution of quantum systems [34, 35].

For this experiment, we selected two single-qubit gates and one two-qubit gate as targets: the single-qubit X gate, the single-qubit H gate, and the two-qubit SWAP gate. These gates were chosen for their diversity and representativeness. The X and H gates are the most basic and commonly used single-qubit gates in quantum computing, used for qubit flipping and creating superposition states, respectively. The SWAP gate is a typical two-qubit gate used to exchange the states of two qubits. By testing

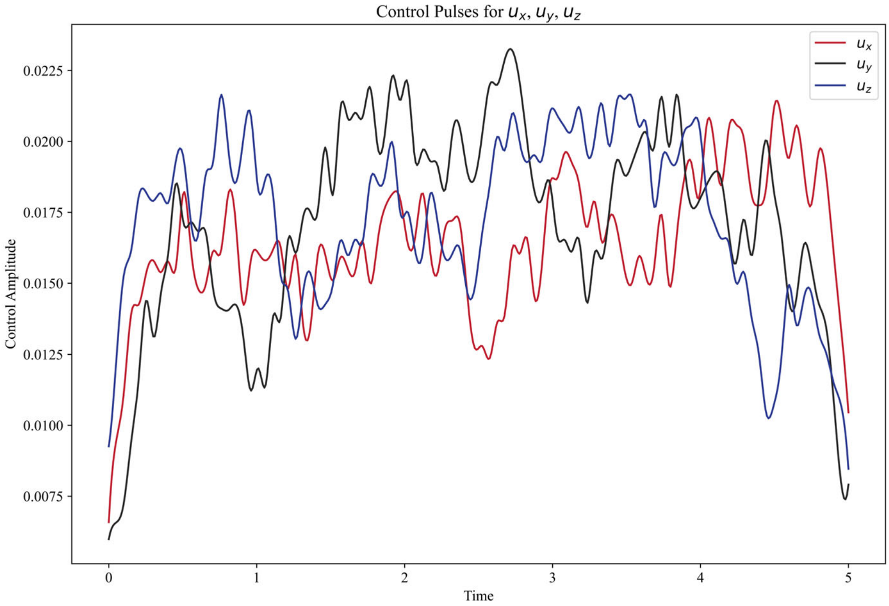

Fig. 2 Control pulse sequence for the X gate generated by QuTiP's pulse optimization function

these fundamental gates, we can evaluate the performance of the Adam algorithm on quantum gates of varying complexity.

During the experiments, we recorded the fidelity changes throughout the optimization process, running each algorithm 1024 times to ensure the reliability of the results. In the experiments, we assumed that the dynamics of the simulated system are described by a drift Hamiltonian H_0 and a set of control Hamiltonians H_k. The system's drift Hamiltonian is σ_Z, and the control Hamiltonians are σ_X, σ_Y, and σ_Z. We set the evolution time T to 5 microseconds, divided into 300 time slices. We then defined the target optimization gate U. Taking the X gate as an example, we generated a random control pulse as shown in Figure 2, where U_x, U_y, and U_z represent the evolution of the control qubit in the x, y, and z directions, respectively.

4.1 Step size optimization

First, we tested the H gate. The experiment revealed that due to the different update mechanisms and objective function characteristics of various algorithms, there are different applicable step sizes. Therefore, in the optimization of different gates, the two algorithms chose different step size ranges, all compared within the range where the gate error rate is minimized. Figure 3 shows the experimental results of the H gate under the Adam and GRAPE algorithms, with the applicable range of EPS (step size) being from 0 to 0.2. It can be seen that as EPS increases, the Adam algorithm can quickly reach a minimum error of 1×10^{-3} and tends to stabilize, compared to the GRAPE algorithm.

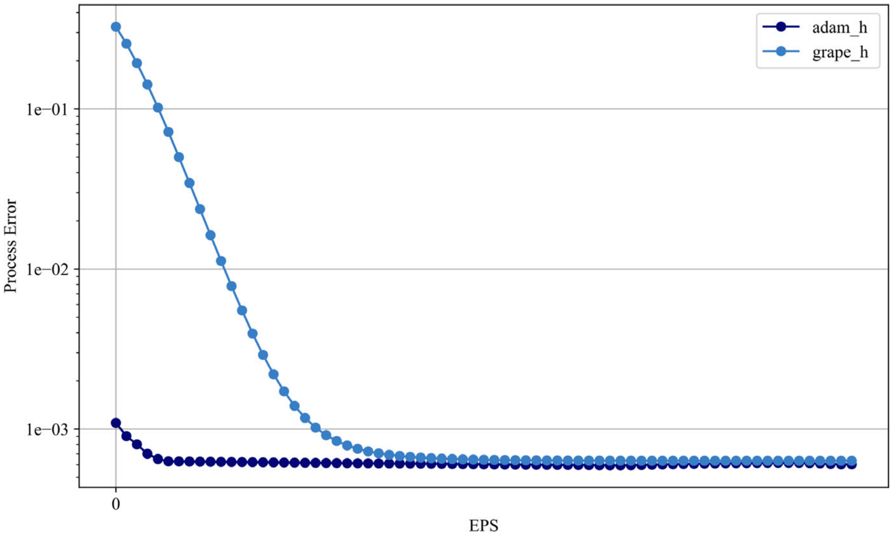

Fig. 3 H gate step size optimization

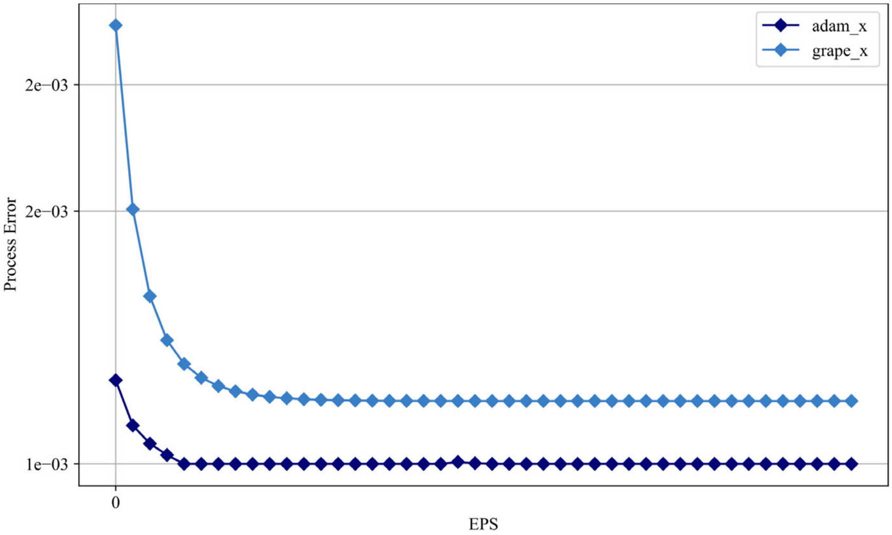

Fig. 4 X gate step size optimization

Next, we tested the X gate, with the applicable range of EPS being from 0 to 0.2. Figure 4 shows the experimental results for the X gate. It can be observed that the convergence speed of both algorithms is essentially the same. As EPS increases, the error rates of both algorithms tend to stabilize. When EPS reaches 0.008, the final error rate of the Adam algorithm stabilizes around 1×10^{-3}. When EPS reaches 0.036, the final error rate of the GRAPE algorithm stabilizes around 1.3×10^{-3}. The error rate of the Adam algorithm is significantly lower than that of the GRAPE algorithm.

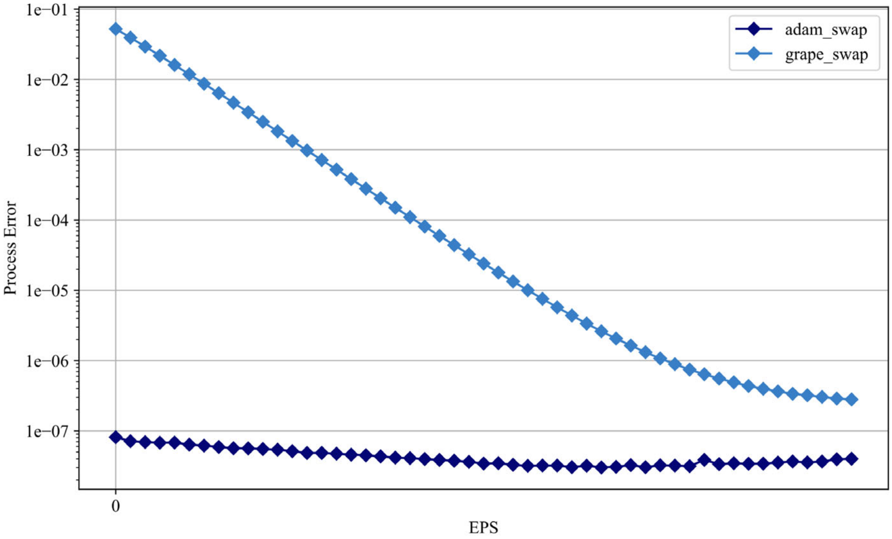

Fig. 5 SWAP Gate Step Size Optimization

Finally, we tested the SWAP gate, with the applicable range of EPS being from 0 to 0.4. Figure 5 shows the experimental results for the SWAP gate. It can be seen that compared to the GRAPE algorithm, the Adam algorithm can achieve a stable error rate more quickly. When the step size for the Adam algorithm is 0.01 and for the GRAPE algorithm is 0.032, the error rates of both algorithms tend to stabilize. The error rate of the Adam algorithm remains around 4×10^{-8}, while the error rate of the GRAPE algorithm remains around 2×10^{-7}.

The experimental results indicate that during pulse optimization, the Adam algorithm can quickly demonstrate a significant advantage in convergence speed within a fixed step size range. Compared to the GRAPE algorithm, it can achieve high fidelity in a faster step size search process. Moreover, the final fidelity optimized by the Adam algorithm is higher than that of the GRAPE algorithm. Specifically, in the waveform optimization of the H gate, X gate, and SWAP gate, the Adam algorithm can achieve fidelity of over 99.94%, 99.88%, and 99.9%, respectively. The H gate and SWAP gate show improvements of 0.03% and 0.0016% compared to the GRAPE algorithm. Since the Adam algorithm can achieve the target fidelity within a smaller step size range and in a shorter time, this may lead to savings in computational resources and time, especially in large-scale quantum computing tasks. Therefore, the efficiency and performance advantages demonstrated by the Adam algorithm in quantum gate optimization problems support its use as a preferred optimization tool.

4.2 Sensitivity analysis of momentum factors

Next, we conducted a sensitivity analysis on the momentum factors β_1 and β_2 in Adam's hyperparameters to evaluate the impact of different settings on algorithm

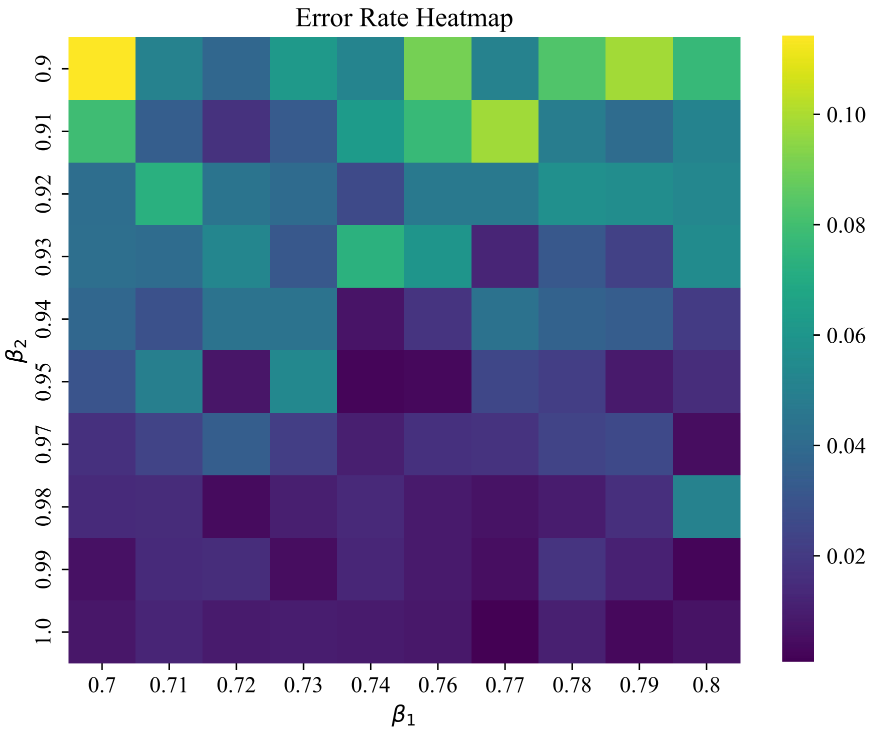

Fig. 6 Hyperparameter optimization results

performance. Theoretically, the values of β_1 and β_2 typically range from 0 to 1. Using the optimization of the H gate as an example, we employed a combination of grid search and random search methods to perform fine-grained parameter scanning within the ranges of β_1 and β_2. Specifically, we first conducted a broad scan of both parameters within the 0–1 range. After determining a more suitable parameter interval, we performed a more detailed narrow-range scan, where β_1 was uniformly distributed between 0.7 and 0.8, and β_2 was uniformly distributed between 0.899 and 0.999. For each combination of momentum factors, we recorded the error rate during the quantum gate training process and verified the reliability of the results through multiple runs. The experimental results are shown in Figure 6.

The experimental data revealed that, under a broad parameter scan, the error rate increased as β_2 decreased while keeping β_1 constant. This may be due to the stability of the second moment estimation, as β_2 is the momentum factor used for the exponentially weighted moving average of the squared gradients. A larger β_2 value implies a smoother and more stable estimation of past squared gradients. This smoothness helps reduce the impact of noise during updates, making parameter updates more stable. Within the finer parameter scan range, the gate error rate ranged from 10^{-4} to 10^{-1}, indicating significant performance differences in quantum gates under different momentum factor combinations. As shown in Table 1, specifically, when the momentum factors were (0.73, 0.94), the quantum gate had the lowest error rate of

Table 1 Optimal momentum factors and final fidelity

β_1	β_2	Computation time (s)	Convergence iterations	Final fidelity
0.74	0.94	25	300	6.77×10^{-4}
0.31	0.91	30	300	1.3×10^{-2}

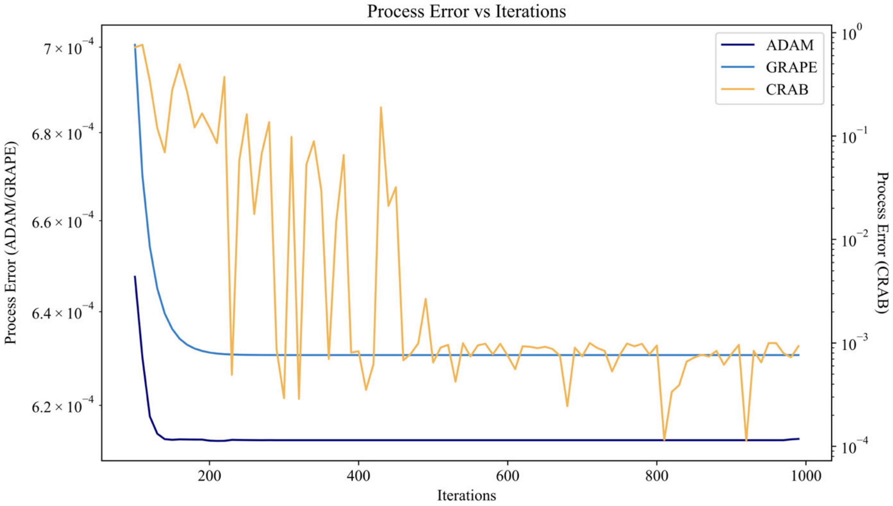

Fig. 7 Iteration count and fidelity

6.77×10^{-4}. Conversely, when the momentum factors were (0.79, 0.91), the error rate was highest at 1.3×10^{-2}. The experimental results indicate that selecting the optimal combination of momentum factors within the studied range can significantly enhance the accuracy and efficiency of quantum gate implementation, effectively reducing the error rate.

4.3 Iteration count comparison

Finally, we conducted a comparative experiment on error rates and iteration counts using the CRAB, GRAPE, and Adam methods. Similarly, we set the same initial control waveform for the H gate and defined a target fidelity of 99.99%, with a maximum iteration count of 1000 for each algorithm. During the experiment, we performed multiple independent experiments to eliminate the impact of randomness on the results and recorded the fidelity changes at each iteration to analyze the convergence speed and final optimization accuracy of each algorithm. The experimental results are shown in Fig. 7.

Figure 8 displays the optimized pulse sequences for the H gate using three different methods. The CRAB-optimized pulse sequence exhibits significant high-frequency oscillations due to its random basis approach. While these wide fluctuations demonstrate the method's global search capability in parameter space, they compromise

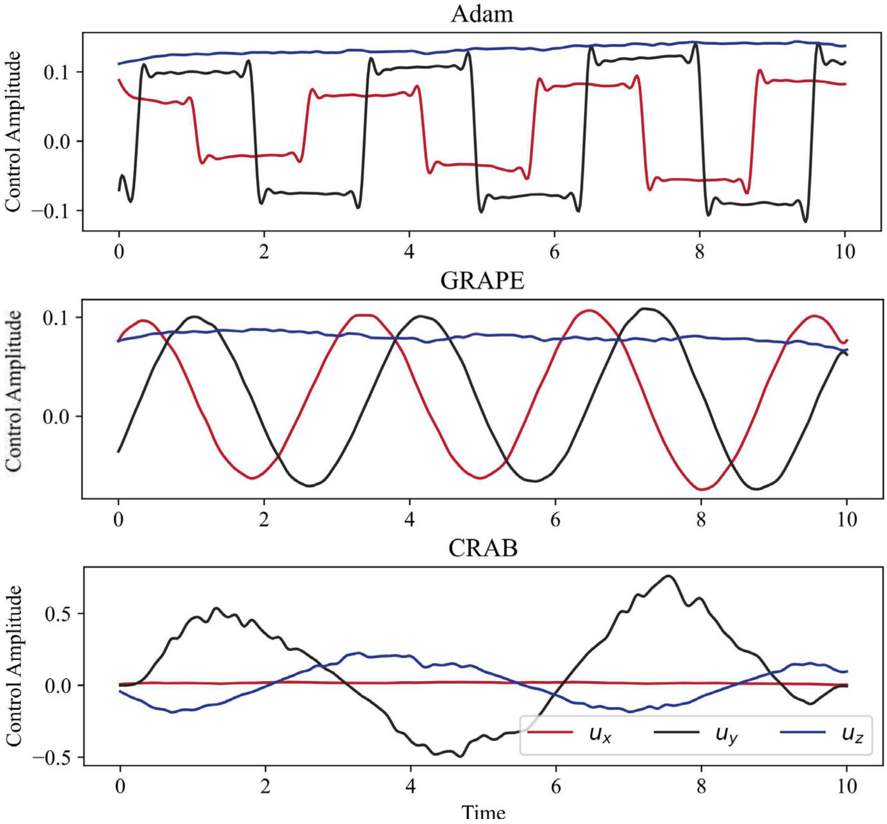

Fig. 8 Comparison chart of H gate control pulse sequences optimized by three methods

experimental feasibility. In contrast, the Adam-optimized pulses show substantially lower peak amplitudes than CRAB, with no high-frequency oscillations, thereby reducing risks of hardware nonlinear distortion. Compared to GRAPE-optimized waveforms, Adam maintains comparable pulse smoothness while achieving higher H gate fidelity.

The experiment shows that the Adam method exhibits a faster convergence speed in the initial stage, quickly approaching the target fidelity around the 100th iteration. This may be attributed to its effective combination of adaptive learning rates and momentum factors, allowing it to find the optimization path more efficiently in the parameter space. In contrast, while the GRAPE method shows a stable convergence trend in the mid-to-late stages, its initial convergence speed is relatively slow. The CRAB method exhibits significant volatility throughout the iteration process, eventually reaching fidelity around the 500th iteration, but its convergence path is less stable than the Adam and GRAPE methods. Compared to the GRAPE and CRAB methods, the fidelity for the H gate also improved by 3.125%. Considering convergence speed, final fidelity, and computational efficiency, the Adam algorithm demonstrates superior overall performance in various application scenarios, providing an important reference for selecting optimization algorithms in quantum control tasks.

5 Conclusion and future work

In this study, we conducted an in-depth analysis of the application of the Adam waveform optimization algorithm in quantum control tasks. The algorithm features an adaptive learning rate, which provides advantages in gradient computation. By maintaining first-order and second-order moment estimates of the gradients, the Adam algorithm adjusts the learning rate for each parameter. The adaptive learning rate characteristic of the Adam algorithm can mitigate the problem of gradient explosion to some extent. Additionally, the introduction of momentum factors allows the Adam algorithm to consider historical information in gradient updates, thereby alleviating the issue of local minima and increasing the likelihood of global convergence.

The experimental results demonstrate that the Adam algorithm exhibits excellent convergence speed in quantum control tasks. Compared to other optimization algorithms, Adam can quickly approach the target fidelity in the initial stages, which is crucial for quantum computing tasks that require rapid response. Its efficient parameter adjustment capability not only enhances optimization efficiency but also significantly reduces the consumption of computational resources. This advantage makes the Adam algorithm highly stable and reliable when dealing with complex quantum systems, providing technical support for achieving high-fidelity quantum operations.

In practical applications, the fast convergence and high-precision characteristics of the Adam algorithm enable it to play a significant role in various quantum operation scenarios, especially in complex quantum systems that require efficient optimization. In summary, this study demonstrates the significant advantages of the Adam algorithm in optimizing quantum control pulses, achieving higher fidelity and faster convergence compared to traditional methods. Future research will focus on closed-loop optimization strategies, utilizing feedback from actual quantum computers to further enhance quantum control performance. Additionally, we plan to extend this framework to open systems by incorporating dissipative terms and environmental noise models.

Author Contributions The authors confirm contribution to the paper as follows: Mengdi Yang and Feng Yue did study conception and design; Mengdi Yang and Bo Lu collected the data; Lixin Wang, Geyuyan Ma and Hanshi Zhao done analysis and interpretation of results; Mengdi Yang was involved in draft manuscript preparation. All authors reviewed the results and approved the final version of the manuscript.

Funding No funding was received to assist with the preparation of this manuscript.

Declarations

Data Availability No datasets were generated or analysed during the current study.

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

1. Alexeev, Y., Amsler, M., Barroca, M.A., et al.: Quantum-centric supercomputing for materials science: a perspective on challenges and future directions. Futur. Gener. Comput. Syst. **160**, 666–710 (2024). https://doi.org/10.1016/j.future.2024.04.060
2. Santagati, R., Aspuru-Guzik, A., Babbush, R., et al.: Drug design on quantum computers. Nat. Phys. (2024). https://doi.org/10.1038/s41567-024-02411-5
3. Portmann, C., Renner, R.: Security in quantum cryptography. Rev. Mod. Phys. **94**, 025008 (2022). https://doi.org/10.1103/revmodphys.94.025008
4. Lloyd, S., Montangero, S.: Information theoretical analysis of quantum optimal control. Phys. Rev. Lett. **113**, 010502 (2014). https://doi.org/10.1103/physrevlett.113.010502
5. De Keijzer, R., Tse, O., Kokkelmans, S.: Pulse based variational quantum optimal control for hybrid quantum computing. Quantum **7**, 908 (2023). https://doi.org/10.26226/m.646636392a24130012d2f5d7
6. Castro, A., García Carrizo, A., Roca, S., et al.: Optimal control of molecular spin qudits. Phys. Rev. Appl. **17**, 064028 (2022). https://doi.org/10.1103/physrevapplied.17.064028
7. Nocedal, J., Wright, S.: Numerical Optimization. Springer Science & Business Media, Cham (2006). https://doi.org/10.1007/b98874
8. Koutromanos, D., Stefanatos, D., Paspalakis, E.: TorchQC-A framework for efficiently integrating machine and deep learning methods in quantum dynamics and control. Comput. Phys. Commun. (2025). https://doi.org/10.1016/j.cpc.2025.109505
9. Khaneja, N., Reiss, T., Kehlet, C., Schulte-Herbrüggen, T., Glaser, S.J.: Optimal control of coupled spin dynamics: design of NMR pulse sequences by gradient ascent algorithms. J. Magn. Reson. **172**, 296–305 (2005). https://doi.org/10.1016/j.jmr.2004.11.004
10. Larocca, M., Wisniacki, D.: Krylov-subspace approach for the efficient control of quantum many-body dynamics. Phys. Rev. A **103**, 023107 (2021). https://doi.org/10.1103/physreva.103.023107
11. Doria, P., Calarco, T., Montangero, S.: Optimal control technique for many-body quantum dynamics. Phys. Rev. Lett. **106**, 190501 (2011). https://doi.org/10.1103/physrevlett.106.190501
12. Müller, M.M., Said, R.S., Jelezko, F., et al.: One decade of quantum optimal control in the chopped random basis. Rep. Prog. Phys. **85**, 076001 (2022). https://doi.org/10.1088/1361-6633/ac723c
13. Goerz, M., Basilewitsch, D., Gago-Encinas, F., et al.: Krotov: a python implementation of Krotov's method for quantum optimal control. SciPost Phys. **7**, 080 (2019). https://doi.org/10.21468/scipostphys.7.6.080
14. Machnes, S., Assémat, E., Tannor, D., et al.: Tunable, flexible, and efficient optimization of control pulses for practical qubits. Phys. Rev. Lett. **120**, 150401 (2018). https://doi.org/10.1103/physrevlett.120.150401
15. Abdelhafez, M., Schuster, D.I., Koch, J.: Gradient-based optimal control of open quantum systems using quantum trajectories and automatic differentiation. Phys. Rev. A **99**(5), 052327 (2019). https://doi.org/10.1103/PhysRevA.99.052327
16. Goerz, M.H., Carrasco, S.C., Malinovsky, V.S.: Quantum optimal control via semi-automatic differentiation. Quantum **6**, 871 (2022). https://doi.org/10.22331/q-2022-12-07-871
17. Kingma, D.P., Ba, J.A.: A method for Stochastic optimization. arXiv preprint arXiv:1412.6980 (2014). https://doi.org/10.48550/arXiv.1412.6980

18. Berezin, F.A., Shubin, M.: The Schrödinger Equation. Springer, Cham (2012). https://doi.org/10.1007/978-94-011-3154-4_3
19. Berman, M., Kosloff, R.: Time-dependent solution of the Liouville-von Neumann equation: non-dissipative evolution. Comput. Phys. Commun. **63**, 1–20 (1991). https://doi.org/10.1016/0010-4655(91)90233-B
20. Breuer, H.P., Petruccione, F.: The Theory of Open Quantum Systems. Oxford University Press, Oxford (2002). https://doi.org/10.1093/acprof:oso/9780199213900.001.0001
21. Krantz, P., Kjaergaard, M., Yan, F., et al.: A quantum engineer's guide to superconducting qubits. Appl. Phys. Rev. **6**, 021318 (2019). https://doi.org/10.1063/1.5089550
22. Ajagekar, A., Humble, T., You, F.: Quantum computing based hybrid solution strategies for large-scale discrete-continuous optimization problems. Comput. & Chem. Eng. **132**, 106630 (2020). https://doi.org/10.1016/j.compchemeng.2019.106630
23. Ajagekar, A., You, F.: Quantum computing for energy systems optimization: challenges and opportunities. Energy **179**, 76–89 (2019). https://doi.org/10.1016/j.energy.2019.04.186
24. Dalgaard, M., Motzoi, F., Jensen, J.H.M., Sherson, J.: Hessian-based optimization of constrained quantum control. Phys. Rev. A **102**, 042612 (2020). https://doi.org/10.1103/physreva.102.042612
25. Kang, J., Zhu, X., Shen, L., et al.: Fault diagnosis of a wave energy converter gearbox based on an Adam optimized CNN-LSTM algorithm. Renew. Energy **231**, 121022 (2024). https://doi.org/10.1016/j.renene.2024.121022
26. Chang, Z., Zhang, Y., Chen, W.: Electricity price prediction based on hybrid model of Adam optimized LSTM neural network and wavelet transform. Energy **187**, 115804 (2019). https://doi.org/10.1016/j.energy.2019.07.134
27. Im, D., Ahn, S., Memisevic, R., et al.: Denoising criterion for variational auto-encoding framework. Proc. AAAI Conf. Artif. Intell. **31**, 1 (2017). https://doi.org/10.1609/aaai.v31i1.10777
28. Reyad, M., Sarhan, A.M., Arafa, M.: A modified Adam algorithm for deep neural network optimization. Neural Comput. Appl. **35**, 17095–17112 (2023). https://doi.org/10.1007/s00521-023-08568-z
29. Zhang, Z.: Improved Adam optimizer for deep neural networks. In 2018 IEEE ACM 26th International Symposium on Quality of Service (IWQoS), 1-2 (2018). https://doi.org/10.1109/IWQoS.2018.8624183
30. Jiang, Z., Xu, F.F., Araki, J., et al.: How can we know what language models know? Trans. Assoc. Comput. Linguist. **8**, 423–438 (2020). https://doi.org/10.1162/tacl_a_00324
31. Beltagy, I., Lo, K., Cohan, A.: SciBERT: A pretrained language model for scientific text. arXiv preprint arXiv:1903.10676 (2019). https://doi.org/10.18653/v1/D19-1371
32. Chuang, I.L., Nielsen, M.A.: Prescription for experimental determination of the dynamics of a quantum black box. J. Mod. Opt. **44**, 2455–2467 (1997). https://doi.org/10.1080/09500349708231894
33. D'Ariano, G.M., Lo Presti, P.: Quantum tomography for measuring experimentally the matrix elements of an arbitrary quantum operation. Phys. Rev. Lett. **86**, 4195–4198 (2001). https://doi.org/10.1103/PhysRevLett.86.4195
34. Johansson, J.R., Nation, P.D., Nori, F.: QuTiP: an open-source Python framework for the dynamics of open quantum systems. Comput. Phys. Commun. **183**, 1760–1772 (2012). https://doi.org/10.1016/j.cpc.2012.02.021
35. Johansson, J.R., Nation, P.D., Nori, F.: QuTiP 2: a Python framework for the dynamics of open quantum systems. Comput. Phys. Commun. **184**, 1234–1245 (2013). https://doi.org/10.1016/j.cpc.2012.11.019