# Hamiltonian quantum generative adversarial networks

Leeseok Kim,[1] Seth Lloyd,[2] and Milad Marvian[1]

[1]*Department of Electrical & Computer Engineering and Center for Quantum Information and Control, University of New Mexico, Albuquerque, New Mexico 87131, USA*

[2]*Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA*

(Received 8 December 2022; accepted 11 June 2024; published 3 July 2024)

We propose Hamiltonian quantum generative adversarial networks (HQuGANs) to learn to generate unknown input quantum states using two competing quantum optimal controls. The game-theoretic framework of the algorithm is inspired by the success of classical generative adversarial networks in learning high-dimensional distributions. The quantum optimal control approach not only makes the algorithm naturally adaptable to the experimental constraints of near-term hardware, but also offers a more natural characterization of overparameterization compared to the circuit model. We numerically demonstrate the capabilities of the proposed framework to learn various highly entangled many-body quantum states, using simple two-body Hamiltonians and under experimentally relevant constraints such as low-bandwidth controls. We analyze the computational cost of implementing HQuGANs on quantum computers and show how the framework can be extended to learn quantum dynamics. Furthermore, we introduce a cost function that circumvents the problem of mode collapse that prevents convergence of HQuGANs and demonstrate how to accelerate the convergence of them when generating a pure state.

DOI: 10.1103/PhysRevResearch.6.033019

## I. INTRODUCTION

Generative adversarial networks (GANs) [1] are one of the most powerful tools of unsupervised learning algorithms in classical machine learning to generate complex and high-dimensional distributions. The learning process of GANs is based on an adversarial game between two players, a generator and a discriminator. The generator's goal is to produce fake data similar to real data, and the discriminator's goal is to discriminate between the data generated from the generator and the real data. Such an adversarial game can be seen as a minimax game that converges to a Nash equilibrium in which the generator efficiently simulates the real data under plausible assumptions [1]. GANs have worked successfully on several realistic tasks including photorealistic image generations [2], image super-resolution [3], video generation [4], molecular synthesis [5], etc.

Inspired by the success of classical GANs, a quantum mechanical counterpart of GANs, a quantum GAN (QuGAN) [6,7], has recently been proposed. Unlike classical GANs, both input and output data in the QuGAN are quantum mechanical, such as an ensemble of quantum states (which could themselves be generated from classical data). In this framework, the generator can be viewed as a quantum circuit that aims to reproduce the ensemble, and the discriminator makes quantum measurements to distinguish the real ensemble from the generated (fake) ensemble. For convex cost functions, such as 1-norm [6] or quantum Wasserstein distance of order 1 [8], the generator and the discriminator perform convex optimization within compact and convex sets: consequently, there always exists a Nash equilibrium point in the generator-discriminator strategy space [6]. In fact, such an equilibrium point is unique and is achieved when the discriminator is unable to tell the difference between the true ensemble and the generated ensemble [6]. Similar to classical GANs, QuGANs have been used to learn random distributions [7], discrete distributions [9], and quantum states [10] and to generate images [11]. Such applications make use of variational quantum algorithms (VQAs) to train QuGANs: the generator and the discriminator are parameterized quantum circuits, where the parameters are optimized via classical optimizers. This approach makes QuGANs feasible to be implemented on near-term noisy intermediate-scale quantum computers (NISQ) [12]. In fact, implementations of QuGANs have already been explored in quantum devices such as superconducting quantum processors to learn quantum states of small systems [13,14].

Rather than performing the computational task using parameterized quantum circuits, one can directly control the parameters of the system Hamiltonian. How to control such time-dependent Hamiltonians is a well-studied field: *quantum optimal control* (QOC) [15]. The goal of QOC is to find optimal sets of control parameters, or pulses, to achieve a predefined goal by steering the dynamics of a given quantum system. Examples of such objectives include optimizing the fidelity between two quantum states, average gate fidelity, and expectation values of an observable [16,17]. In fact, some applications of QOC give promising results in the field of quantum computation, such as designing high-fidelity quantum entangling gates [18,19]. In addition, QOC can effectively reduce the latency of groups of quantum gates, which current gate-based compilations for quantum systems suffer from [20]. There have been extensive studies on developments and applications of popular methods of QOC including gradient-based methods such as GRAPE [21] and Krotov [22] and gradient-free methods such as CRAB [23], to many different quantum systems.

In this work, we introduce a Hamiltonian QuGAN (HQuGAN), a framework to generate quantum resources, such as quantum states or unitary transformations, by directly controlling the native parameters of system Hamiltonians using two competing quantum optimal controls, one for the generator and one for the discriminator.

The proposed HQuGAN has several favorable properties compared to circuit-model variational algorithms. First, unlike parameterized quantum circuit models, HQuGANs perform the learning task by changing the native parameters of the Hamiltonian itself. In the circuit model, each quantum gate must be translated into control pulses, for example, electrical signals, that implement the specified operations on the underlying quantum device. But it is not necessarily the case that the variational parameters specified by the algorithm, can be directly translated to the control pulses implementing the gate. This creates a barrier between the expressibility of the logical gates and the set of operational instructions that can be efficiently implemented on real experimental systems [24]. Besides, even if the approximate translation is possible, the algorithm suffers from possible gate errors accumulated by each translated gate, causing a mismatch between the ideal gates and the implemented pulses [25]. Controlling the Hamiltonian itself, however, avoids both barriers.

For these reasons, recent studies have indicated that replacing variational quantum circuits with QOC methods can be advantageous for NISQ devices, which have a limited gate depth due to a short coherent time and gate errors. For instance, Ref. [26] introduces an algorithm for variational quantum eigensolver (VQE) simulations at the device level using QOC, which significantly reduces the coherence time required for the state preparation by several orders of magnitude compared to using variational quantum circuits in superconducting transmon platforms. Furthermore, Ref. [27] has extended the work and demonstrated that one can prepare target molecular ground states on the transmon processors within the optimal time by directly controlling a device Hamiltonian that describes coupled transmon qubits. Similarly, Ref. [28] conducts a comparison between two methods to approximate molecular ground states of various molecules. The authors show that directly controlling Hamiltonians using QOC generally has better convergence and requires fewer quantum resources compared to the gate-based approaches. In addition, when considering short evolution times, it outperforms the gate-based approaches.

Moreover, HQuGANs can benefit from *overparameterization* due to the continuous nature of the control parameters, leading to a better convergence on the minimax game. For classical GANs, it has recently been shown that overparameterization appears to be a key factor in the successful training of GANs to global saddle points [29]. Furthermore, overaparameterization appears to provide substantial advantages in training deep neural networks [30]. In the (circuit model) quantum setting, it has been shown that while underparameterized quantum neural networks (QNNs) have spurious local minima in the loss landscape, overparameterized QNNs make the landscape more favorable and thus substantially improve a trainability of QNNs [31–33]. Given the fact that Hamiltonian quantum computing includes the circuit model as a specific subcase, the advantage of the overparameterization phenomenon also applies to QOC models [34]. Importantly, a key observation we make in this work is that, for NISQ devices, directly controlling the parameters of the experimentally available Hamiltonian provides a more natural route to achieve overparameterization. We quantify this observation using optimal control bounds and verify the performance using numerical simulations. We discuss methods to incorporate experimentally relevant constraints on control fields such as low-bandwidth controls and their effect on overparameterization.

The paper is organized as follows. We begin by introducing the concepts of GANs and QuGANs in Sec. II, followed by an introduction to quantum optimal control (QOC) in Sec. III with a specific focus on the GRAPE method. We then describe in Sec. IV methods to incorporate bandwidth limitations of the control fields and the effect on the parametrization of the control problem. In Sec. V we introduce our Hamiltonian QuGANs (HQuGANs), followed by numerical simulations on generating different quantum states using the proposed HQuGAN in Sec. VI. We then highlight in Sec. VII how different cost functions can affect the convergence rate of the HQuGAN. Specifically, we introduce a form of cost function for QuGANs to circumvent the issue of mode collapse that was first raised in Ref. [10]. Executing the QOC for large systems can be computationally infeasible using classical computers. Hence, we propose methods to use quantum computers as subroutines of the HQuGAN to avoid such intractability in Sec. VIII and analyze the required resources.

## II. QUANTUM GAN

In classical GANs, to learn a distribution $p_g$ over data $x$, we consider a parameterized generative neural network map $G(\theta_g, z)$ where $\theta_g$ represents the parameters of the network and $p_z(z)$ is a prior on the input noise variables. We also define another parameterized map $D(\theta_d, x)$, corresponding to the discriminative neural network, that outputs the probability that a given $x$ is sampled from the data set rather than the generator's distribution $p_g$. The goal of the generator is to fool the discriminator by generating $G(\theta_g, z)$ that is indistinguishable from $D(x)$. The discriminator then tries to distinguish between the true data distribution and the generator's distribution, the best she can. Hence, GANs alternate between the discriminator maximizing the probability of assigning the correct label to both training examples and samples from $G(\theta_g, z)$ and the generator minimizing the same loss that $\theta_d$ is maximizing. Formally, the two players play the following minimax game by solving

$$\min_{\theta_g} \max_{\theta_d} V(\theta_g, \theta_d) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(\theta_d, x)] + \mathbb{E}_{z\sim p_z(z)}[1 - D(\theta_d, G(\theta_g, z))], \quad (1)$$

where a global Nash equilibrium point exists at $p_g = p_{\text{data}}$ [1]. In practice, however, training GANs to reach the desired equilibrium point can be challenging for several reasons, such as vanishing gradients [35] and mode collapse [36]. While none of these issues have been completely solved, there are several attempts to remedy the issues including by using Wasserstein GANs [37] and modifying the minimax cost function [38]. Since similar problems have observed in quantum GANs [10,39], we later discuss how to remedy them using quantum Wasserstein GANs [8,40] and modifying the minimax cost function in Sec. VII.

In a quantum GAN (QuGAN), the goal is to learn an unknown quantum state $\sigma$, representing the true data. This goal is achieved by an iterative game played by two quantum agents: a generator and a discriminator. In each iteration, after the generator updates his parameters $\theta_g$ to produce a density matrix $\rho(\theta_g)$, the discriminator takes as input the quantum state from the generator or the true data and performs a discriminating measurement. In other words, the discriminator attempts to find a Hermitian operator $D$ that maximally separates the expected values with respect to the two quantum states, i.e., maximizing $\text{Tr}[D(\sigma - \rho)]$. As a consequence, the objective of QuGANs can be expressed as solving [6]

$$\min_{\theta_g} \max_{D} \text{Tr}[D(\sigma - \rho(\theta_g))]. \tag{2}$$

The core idea of QuGANs, analogous to classical GANs, is based on an *indirect* learning process of the minimax game suggested above, where it aims to generate the true quantum state $\sigma$ without using classical descriptions of $\sigma$. While including the optimal discriminative measurements the constraint $\|D\|_\infty \leqslant 1$ was considered in the original QuGAN proposal [6], recently a QuGAN based on the quantum Wasserstein distance of order 1 (or quantum $W_1$ distance) has been proposed [40].

The quantum $W_1$ distance is based on the notion of neighboring quantum states. Two states are called neighbors if they differ only by one qubit. The quantum $W_1$ distance is then the maximum norm induced by assigning distance at most one to every couple of neighboring states. Using the quantum $W_1$ distance dual formulation [40], minimizing the quantum $W_1$ distance can be expressed as the following minimax game:

$$\min_{\theta_g} \max_{D} \{\text{Tr}[D(\sigma - \rho(\theta_g))], \|D\|_L \leqslant 1\}, \tag{3}$$

where the quantum Lipschitz constant of an observable $H$ is defined as

$$\|H\|_L = 2 \max_{i=1,\dots,n} \min\{\|H - H^{(i)}\|_\infty : H^{(i)} \text{ does not act on the } i\text{th qubit}\}. \tag{4}$$

In fact, a recent work [8] shows that the loss landscape induced by using the quantum $W_1$ distance as the cost function for QuGANs can potentially provide an advantage in learning certain structured states like the GHZ states compared to other metrics such as fidelity. In addition to the optimization landscape, the choice of cost functions also can affect the rate of convergence and the attainability of the equilibrium point. The main obstacle preventing proper convergence of QuGANs is mode collapse [10]. This happens because the generator in Eq. (2) focuses on producing a state that aligns with $D$ without considering the target state $\sigma$. We introduce a novel form of cost function that prevents this issue, as it will be discussed in Sec. VII.

## III. QUANTUM OPTIMAL CONTROL

The goal of quantum optimal control (QOC) is to find control parameters, or control pulses $\{\epsilon_i(t)\}$, that achieve a predefined task, for example, generating a desired (known) quantum state, using a given Hamiltonian such as

$$H(t) = H_0 + \sum_i \epsilon_i(t) H_i, \tag{5}$$

where $H_0$ is the drift Hamiltonian and $\{H_i\}$ is the set of control Hamiltonians. A standard approach in QOC is to optimize an objective functional that depends on the control fields $J[\{\epsilon_i(t)\}]$,

$$\min_{\{\epsilon_i\}} J[\{\epsilon_i(t)\}] \tag{6}$$

subject to the Schrödinger equation of the time-dependent Hamiltonian. A common choice for the main functional is the infidelity between fully time-evolved quantum states and a known target state, e.g.,

$$F = 1 - |\langle\psi_{\text{targ}}|\psi(T)\rangle|^2 = 1 - |\langle\psi_{\text{targ}}|U(T)|\psi_0\rangle|^2, \tag{7}$$

where $U(T) = \mathcal{T}\exp[-i\int_0^T H(t)\,dt]$ is the total evolution propagator of $H(t)$ from $t \in [0, T]$ and $|\psi_0\rangle$ is an initial state. Additional penalty terms can be added to the cost function to achieve specific tasks such as realizing smooth, low-bandwidth controls by suppressing rapid variations of the control fields [41], which we will discuss in more detail in Sec. IV. It is worth noting that quantum optimal control, similar to a gate-based model, is subject to a phenomenon known as barren plateaus in the control landscape associated with variational optimizations [39,42]. This refers to a situation where the gradient of the objective functional $J[\epsilon_i(t)]$ [such as Eq. (7)] vanishes exponentially in terms of the size of the quantum system. To address the issue at hand, which is particularly relevant when learning certain structured quantum states such as GHZ states, it could be beneficial to consider alternative cost functions such as the quantum $W_1$ distance [8,40]. Currently, it is an open problem to determine the extent to which the quantum $W_1$ distance can improve the landscape of cost functions in quantum optimal control.

A popular QOC technique is Gradient Ascent Pulse Engineering (GRAPE) [21]. Since $U(T)$ is difficult to obtain analytically, the GRAPE algorithm first discretizes the time domain into $N$ equal pieces of intervals ($\Delta t = T/N$) and approximates the Hamiltonian with a piecewise constant Hamiltonian within each interval $[t_j, t_j + \Delta t)$. The total time evolution operator can then be expressed as a product of $N$ unitary matrices,

$$U(T) = \prod_{j=1}^{N} U(t_j) = \prod_{j=1}^{N} \exp\left[-i\Delta t\left(H_0 + \sum_i \epsilon_i(t_j) H_i\right)\right]. \tag{8}$$

Then the control fields at all the time steps are updated concurrently using their gradients with respect to the objective function $J$,

$$\epsilon_i(t_j) \longleftarrow \epsilon_i(t_j) + \alpha \frac{\partial J}{\partial \epsilon_i(t_j)}, \tag{9}$$

where the gradient can be obtained via approximating the propagator derivatives to the first order of $\Delta t$ [21],

$$\frac{\partial U(t_j)}{\partial \epsilon_i(t_j)} \approx -i\Delta t H_i U(t_j), \tag{10}$$

which makes the computation very affordable. To achieve faster and more stable convergence of the optimization process, one can incorporate a quasi-Newton method, particularly the Broyden-Fletcher-Goldfarb-Shanno algorithm (BFGS) [43] or limited-memory BFGS (L-BFGS) [44] in the GRAPE algorithm, which requires calculating the Hessian matrix of the cost function.

Another quantum optimal control protocol we consider in this paper is Krotov's method [22]. The method is based on a rigorous examination of conditions for calculating the updated control fields such that it always guarantees a monotonic convergence of the objective functional $J[\{\epsilon_i(t)\}]$ by construction. An appealing feature of Krotov's method is that it mathematically guarantees that control fields are continuous in time [45]. However, it is computationally more expensive than GRAPE since a single optimization step requires solving the Schrödinger equations $2N$ times, where $N$ is the number of time steps. We discuss the details of Krotov's method and its applications to HQuGANs in Appendix D.

## IV. CONTROL BANDWIDTH AND OVERPARAMETERIZATION

In practice, it is often desirable to generate bandwidth-limited control fields, as high-frequency control pulses are hard to implement with high accuracy in many experiments. There exist various ways to constrain the bandwidth of control fields in different quantum optimal control techniques [41,46–48]. One of the most common methods is to penalize rapid variations of control fields by adding the following penalty term to the cost function [41]:

$$J_p = \alpha \sum_{i,j} |\epsilon_i(t_j) - \epsilon_i(t_{j-1})|^2. \tag{11}$$

Minimizing $J_p$ reduces the variations of every pair of adjacent control pulses and thus serves as a soft penalty term to limit the control bandwidth. This penalty term has been successfully used to find low-bandwidth control in many quantum optimal control settings [41,49].

In the circuit model, the number of independent parameters that can be varied to implement an algorithm is directly determined by the number of parameterized quantum gates. In Hamiltonian quantum computing (HQC), determining the number of independent parameters can be more involved. Intuitively, the number of free parameters in the HQC setting should increase linearly with the total evolution time $T$. Also, a smaller cost function $J_p$, or equivalently, a smaller control bandwidth should decrease the corresponding number of independent parameters. Such intuition has been formally proven via an information-theoretic argument on the information content of a classical field controlling a quantum system [50].

Such a bound can be derived by first defining a minimum number of $\epsilon$ balls to cover the whole space of reachable states of a given quantum system so that one of the balls identifies a generic target state within a radius $\epsilon$. To uniquely specify which ball the target state is in, the control fields need to be able to express at least as many configurations as the number of balls. As a consequence, one can derive the following fundamental quantum speed limit in terms of the bandwidth of the control field:

$$T \geqslant \frac{D}{\Delta\Omega \kappa_s} \log_2(1/\epsilon), \tag{12}$$

where $D$ is the dimension of a set of reachable states of a given quantum system ($D = 2^{2n}$ in general, $2^n$ for pure states), $\Delta\Omega$ is the bandwidth of the control field, $\kappa_s = \log_2(1 + \Delta\gamma/\delta\gamma)$ ($\Delta\gamma$ and $\delta\gamma$ are the maximal and minimal allowed variations of the control field), and $\epsilon$ is a maximum (any) norm difference between the target state and a state generated by the control field [50]. This time-bandwidth quantum speed limit thus tells us that control fields with higher bandwidth require less evolution time $T$ to steer a quantum system to achieve a target state, compared to control fields with lower bandwidth. The bound of Eq. (12) has been numerically verified in various settings [51,52]. Since the number of independent parameters is proportional to $T\Delta\Omega$, the time-bandwidth quantum speed limit in Eq. (12) provides the dimension of a set of reachable states $D$ as the lower bound on the number of parameters to reach any state in the set. This bound matches the result of Ref. [31], where it has been shown that having as many parameters as the dimension of the dynamical lie algebra of a given system is enough to achieve overparameterization in parametrized quantum circuits.

This relationship between the number of independent variables and the control bandwidth provides us with a tool to study the trade-off between the limited bandwidth to implement control fields in experimental settings and the advantage of overparameterization in the performance of classical GANs [29] and also quantum neural networks [31,32]. On the other hand, overparameterization in the circuit model is solely determined by the number of parameterized quantum circuits, which fails to account for experimental constraints. We numerically verify this relationship by proposing the penalty term in the cost function of HQuGANs in Appendix E.

## V. HAMILTONIAN QUANTUM GAN

We now introduce the HQuGAN algorithm to learn an arbitrary unknown quantum state $\sigma$. As illustrated in Fig. 1, the learning process is based on a minimax game consisting of two players, a generator and a discriminator, where each player has access to a Hamiltonian in the form of Eq. (5). At each round of the HQuGAN, each player uses quantum optimal control techniques to update the control parameters of their Hamiltonians to optimize the cost function $C$ while fixing the other player's parameters. More specifically, in each round, the generator finds optimal control parameters $\{g\}$ such that the generated quantum state $\rho(\{g\})$ minimizes the cost

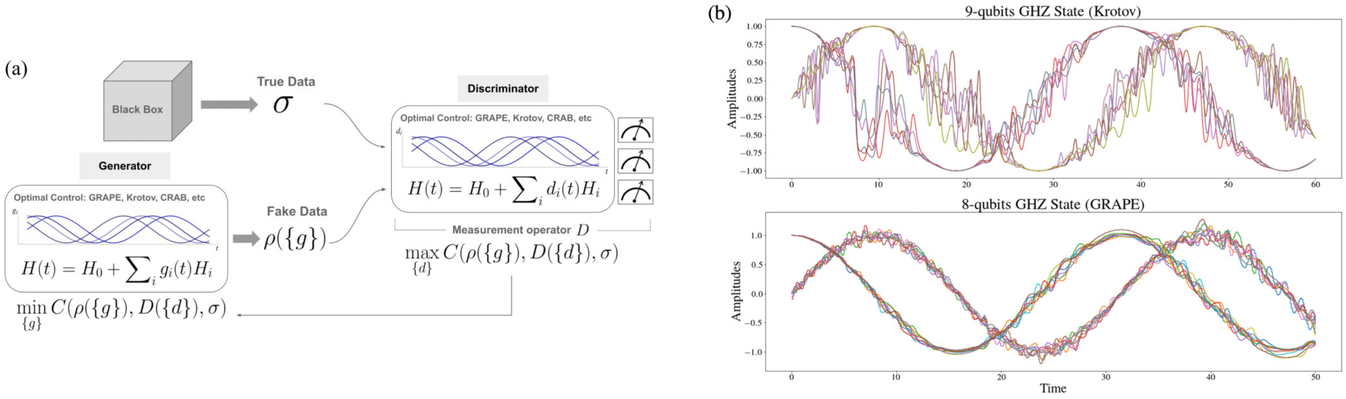

FIG. 1. (a) Schematic of a general Hamiltonian quantum generative adversarial network (HQuGAN) protocol. Given an unknown quantum state $\sigma$ generated from a black box (e.g., an unknown quantum process or experiment), the ultimate goal of the HQuGAN is to find control fields that generate a fake quantum state $\rho(g)$ close to $\sigma$ as much as possible. To achieve this task, the HQuGAN employs an iterative game with the objective function $C$ between two players, a generator and a discriminator, who update control fields ($\{g\}$, $\{d\}$) in every round while the other player's fields are kept fixed. The generator aims to produce $\rho(\{g\})$ to fool the discriminator by minimizing $C$, but the discriminator tries to discriminate the two quantum states by maximizing $C$. Such protocol can be phrased as a minimax game $\min_{\{g\}} \max_{\{d\}} C(\rho(\{g\}), D(\{d\}), \sigma)$, where the objective function $C$ can be in various forms. (b) Examples of optimized control fields to generate generalized GHZ states. Optimized control fields that successfully generate nine- and eight-qubit GHZ states using Krotov's method and GRAPE in the HQuGAN setting are presented respectively. Different colors represent the time-dependent control fields for the local Pauli $X$ and $Z$ terms in the Ising chain in Eq. (15).

function $C$. Once the generator's turn is finished, the discriminator finds her optimal control parameters $\{d\}$ that produce a measurement operator $D$ that maximally discriminates the two quantum states $\sigma$ and $\rho(\{g\})$. As illustrated in Fig. 1, in this work we restrict the measurements to the ones that can be decomposed as a parameterized quantum dynamics [generated by Hamiltonian in the form of Eq. (5)] followed by a fixed quantum measurement $D_0$. Hence, the measurement operator $D = U^\dagger(\{d\})D_0U(\{d\})$. Therefore, the HQuGAN solves the following game:

$$\min_{\{g\}} \max_{\{d\}} C(\rho(\{g\}), D(\{d\}), \sigma). \tag{13}$$

Such an iterative game between the two players continues until the fixed point is approximately reached, or other desired criteria, such as the Uhlmann fidelity between the generator's state and the target state, are achieved. The algorithm is also described in Algorithm 1.

ALGORITHM 1. Hamiltonian Quantum Generative Adversarial Networks for Learning Arbitrary Quantum State.

**Input:** Time-dependent Hamiltonian $H(t)$, unknown target state $\sigma$, initial control fields $\{\epsilon_i(0)\}$, initial state $\rho_0$, evolution time $T$, Trotter steps $N$, fixed measurement operator $D_0$
**Output:** Control fields $\{\epsilon_i\}$ (which can be used to generate a quantum state $\rho(\{\epsilon_i\})$ close to the target state $\sigma$)

```
procedure Hamiltonian Quantum Generative Adversarial Networks
    while $F(\rho(\{g_i\}), \sigma) \leqslant 0.999$ do                ▷ Terminates when $F(\rho(\{g_i\}), \sigma) > 0.999$
        procedure Generator
            if first round of QuGAN
                Initialize initial control pulse $\{g_i(0)\} = \{\epsilon_i(0)\}$ and fixed measurement operator $D = D_0$
            Minimize $C(\rho(\{g_i\}), D(\{d_i\}), \sigma)$ using QOC
            $\{g_i(0)\} \leftarrow \{g_i\}$                             ▷ Updates optimize control fields
          return $\rho(\{g_i\})$                    ▷ Returns time-evolved quantum state with optimized control fields
        Procedure Discriminator
            Initialize $\{d_i(0)\} = \{\epsilon_i(0)\}$
            Maximize $C(\rho(\{g_i\}), D(\{d_i\}), \sigma)$ using QOC
            $D \leftarrow U^\dagger(\{d_i\})D_0U(\{d_i\})$                  ▷ Updates new observable $D$
            return $D$
    $\{\epsilon_i\} \leftarrow \{g_i\}$                                 ▷ Obtain final control fields $\{g_i\}$
    return $\{\epsilon_i\}$
```

As discussed in Sec. II, we will study various forms of the cost function $C$. Choosing $C = \mathrm{Tr}[D(\{d\})(\rho(\{g\}) - \sigma)]$ recovers the trace distance, used in [6], when $\|D\|_\infty \leqslant 1$ and the quantum $W_1$ distance when $\|D\|_L \leqslant 1$ [the quantum Lipschitz constant is described in Eq. (4)]. The choice of the cost function not only changes the optimization landscape, but can also affect the reachability of the fixed point, which will be discussed in Sec. VII. To study this issue, an additional cost function that we consider is

$$C = |\mathrm{Tr}[D(\{d\})(\sigma - \rho(\{g\}))]|^2. \tag{14}$$

In fact, the minimax game using the cost function above has a Nash equilibrium point at the desired location. The Nash equilibrium is a stationary point where no player can benefit by changing their strategy while the other player keeps their strategy unchanged. In other words, the Nash equilibrium is a point $(\{g\}^*, \{d\}^*)$ where $\{g\}^*$ gives a global minimum of $f(\cdot, \{d\}^*)$ and $\{d\}^*$ gives a global maximum of $f(\{g\}^*, \cdot)$. Therefore, the above minimax game has the Nash equilibrium at the desired location of $\rho(\{g\}^*) = \sigma$.

## VI. NUMERICAL EXPERIMENTS

In this section, we present numerical experiments on the performance of the proposed HQuGANs in learning various many-body quantum states and quantum dynamics.

### A. Setup

Motivated by current experimental capabilities [53–55], to test the performance of the proposed algorithm we consider an $n$-qubit 1D time-dependent longitudinal and transverse field Ising model (LTFIM) Hamiltonian [56] with open boundary conditions,

$$H(t) = \sum_{i=1}^{n} \epsilon_i(t) X_i + \sum_{i=1}^{n} \epsilon_{i+n}(t) Z_i - J \sum_{i=1}^{n-1} Z_i Z_{i+1}, \tag{15}$$

for both the generator and the discriminator. (We set $\hbar = 1$, and therefore the coupling parameters are expressed in hertz. For example, if the total evolution time $T$ is in nanoseconds, then $\epsilon_i(t)$ is in gigahertz.) Note that the strength of all the $ZZ$ couplings is set to a fixed value (i.e. $J = 1$), and we assume only the stringent condition of having control over the local fields. (Of course, having more control, especially over the entangling interactions, will introduce more degrees of freedom and therefore will reduce the required time to generate arbitrary quantum states.) Hence, for an $n$-qubit system, each player optimizes over $2n$ control pulses of the local Pauli terms in the Hamiltonians.

The initial control fields are chosen as simple sinusoidal shapes that can be easily generated by both players,

$$\epsilon_i(t=0) = \begin{cases} \sin(10t/T), & \text{for } i = 1, \dots, n, \\ \cos(10t/T), & \text{for } i = n+1, \dots, 2n. \end{cases} \tag{16}$$

[We also consider a constant initial control, $\epsilon_i(t=0) = 1$, for the bandwidth analysis in Appendix E.] We set the initial state to be the easily preparable state $|1\rangle^{\otimes n}$, which is the ground state of $H(t)$ at $t = 0$. We keep control pulses at $t = 0$ unchanged for the generator by setting the gradients of control fields at $t = 0$ to zero so that the generator always begins with $|1\rangle^{\otimes n}$. In addition, we set the observable $D_0$ to be a 1-local computational basis measurement, i.e., $D_0 = Z \otimes I^{\otimes n-1}$.

We consider the cost function of the form (14), and for the discriminator, we consider the constraints $\|D\|_\infty \leqslant 1$ and $\|D\|_L \leqslant 1$, corresponding to the trace distance and the quantum $W_1$ distance. (We discuss the effect of the cost function in more detail in Sec. VII.) Gradients of cost functions are approximated to the first order of $\Delta t = T/N$ for both the generator and the discriminator, and all experiments are optimized via the L-BFGS method. The termination criterion we consider is achieving at least 0.999 fidelity with the target state. We provide a comprehensive description of how the generator and the discriminator are trained using the quantum optimal control method in Appendix A. All the simulations are performed using the optimal control module in QuTiP [57], with the appropriate modifications for the various cost functions studied in this work.

### B. Learning three-qubit states

We attempt to learn 50 different three-qubit superposition states,

$$|\psi_{\mathrm{targ}}\rangle = \cos\theta_k |000\rangle + \sin\theta_k |111\rangle, \tag{17}$$

where $\theta_k = 2\pi k/50$ for $k = 0, 1, \dots, 49$, using the proposed HQuGAN protocol. The GRAPE algorithm is used for the quantum optimal control for both the generator and the discriminator, using a total evolution time $T = 5$ with $N = 50$ Trotter steps. Each player performs full optimization at each round. (The optimization terminating criteria is if the cost function is within $10^{-5}$ of the extreme point, or if the norm of the gradients is smaller than $10^{-5}$, or if the maximum iteration of 50 is reached.)

In Fig. 2 the number of iterations of the HQuGAN algorithm to successfully learn to generate rotated GHZ states with at least 0.999 fidelity is presented. Clearly, the HQuGAN successfully generates all 50 states (red lines). To evaluate the performance, we compare the result to an *optimal discriminator* that always chooses the Helstrom measurement (blue lines), which we choose as a sum of two projectors onto positive and negative eigenspaces of $\rho - \sigma$ (see in Appendix B). For almost all instances, the blue line performs comparability well, indicating that the discriminative models using the GRAPE method are expressive enough for the HQuGAN to successfully learn the target states. Indeed, we have numerically observed that optimizing using the GRAPE algorithm always produces a discriminator very close to the Helstrom measurement that fully maximizes $\mathrm{Tr}[D(\rho - \sigma)]$. Another interesting point to note is that the optimal discriminator result can be achieved using GRAPE (instead of calculating it analytically) by initializing the discriminator operator $D_0$ as a rank-2 operator, as the rank-2 Helstrom measurement operator is unique. To demonstrate this, we attempt to generate the same target states using the discriminator with GRAPE, with the initial operator $D_0 = |0\rangle\langle 0|^{\otimes n} - |1\rangle\langle 1|^{\otimes n}$, which is a rank-2 operator. The performance is presented as the green lines in the figure. As illustrated in the plot, we observe that the performance of the blue and green lines match perfectly. In

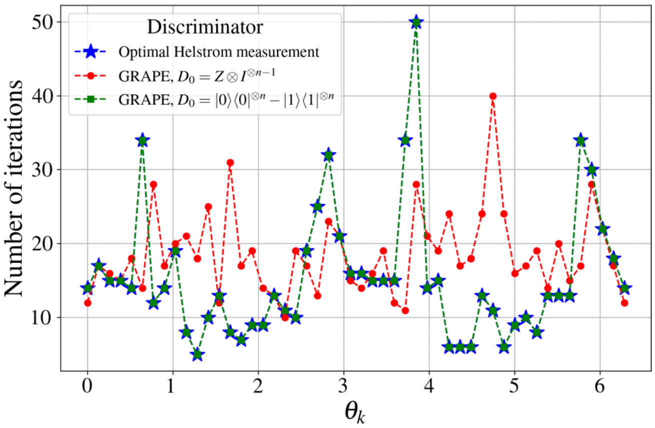

FIG. 2. HQuGAN experiments for learning three-qubit states. The number of iterations required to generate 50 different three-qubit generalized GHZ states, where $\theta_k$ are angles of the GHZ state in Eq. (17), is presented for three different discriminators: a discriminator that maximizes Eq. (14) using GRAPE with $D_0 = Z \otimes I^{\otimes n-1}$ (red) and $D_0 = |0\rangle\langle 0|^{\otimes n} - |1\rangle\langle 1|^{\otimes n}$ (green), and the optimal discriminator performing the Helstrom measurement analytically (blue). While the optimal discriminator performs overall the best, the discriminator using GRAPE with $D_0 = Z \otimes I^{\otimes n-1}$ gives comparable numbers of iterations. By setting the initial discriminator (with GRAPE) operator $D_0$ as a rank-2 operator, we can obtain the analytic Helstrom measurement operator, given the uniqueness of the rank-2 Helstrom measurement operator. This is demonstrated by the perfect match between the green and blue lines.

Sec. VII C and Appendix B, we discuss how using the optimal discriminator can accelerate the convergence of the minimax game. By fixing $D_0$ to be a rank-2 operator, therefore, we can achieve the speedup when the discriminator is using optimal control protocols.

### C. Learning generalized GHZ states

After the successful learning of various three-qubit superposition states, we now shift gears to the challenging task of generating generalized Greenberger-Horne-Zeilinger (GHZ) states, which are extremely useful resource states in quantum information and quantum metrology. We hence focus on generating $n$-qubit GHZ states,

$$|\psi_{\text{targ}}\rangle = \frac{1}{\sqrt{2}}(|0\rangle^{\otimes n} + |1\rangle^{\otimes n}), \tag{18}$$

with keeping the HQuGAN settings unchanged from the previous experiment. We keep $T$ proportional to the system sizes and the number of time grids to $N = 10T$ for all instances. As before, we use both GRAPE-equipped and optimal discriminators to evaluate the performance of the HQuGAN. We set $D_0 = Z \otimes I^{\otimes n-1}$.

The numerical experiments are summarized in Table I. The HQuGAN successfully generates up to the six-qubit GHZ state using the optimal control (GRAPE) discriminators with a number of iterations similar to the iterations required for the optimal discriminator.

TABLE I. HQuGAN for learning generalized GHZ states. The number of iterations required for the HQuGAN to learn generalized $n$-qubit GHZ states with a discriminator that uses GRAPE and the optimal discriminator. We observe that the numbers of iterations of the HQuGAN for both cases are comparable.

| $n$ | Iter. (GRAPE) | Iter. (Helstrom) | $T$ | $N$ |
|---|---|---|---|---|
| 1 | 3 | 3 | 5 | 50 |
| 2 | 6 | 3 | 5 | 50 |
| 3 | 21 | 8 | 5 | 50 |
| 4 | 38 | 35 | 10 | 100 |
| 5 | 56 | 62 | 20 | 200 |
| 6 | 88 | 111 | 30 | 300 |

### D. Learning Haar random states

Finally, we attempt to learn 50 Haar random quantum states, i.e., states drawn from the Haar measure, up to six qubits. Similar to the previous experiments, the GRAPE algorithm is used for the quantum optimal control for both the generator and the discriminator. Table II shows both the mean and standard deviation values for the number of iterations of the HQuGAN to successfully learn to generate all 50 Haar random states with at least 0.999 fidelity. We find that the mean number of iterations required by the HQuGAN algorithm increases exponentially in terms of the system size, which is not surprising considering the fact that learning generic quantum states demands exponentially many resources [58].

### E. Krotov's method

As discussed earlier, Krotov's method is another popular gradient-based QOC technique. In contrast to GRAPE, Krotov's method mathematically guarantees that the control pulse sequences remain time-continuous throughout the optimization process [59]. We conduct the same tasks of generating 50 entangled three-qubit states in Eq. (17) and the generalized GHZ states using the HQuGAN with Krotov's method. The HQuGAN generates all instances well as presented in Fig. 3 and Table III. We further produce up to the nine-qubit GHZ state using the optimal discriminator, and also using experimentally realizable parameters. More details on the de-

TABLE II. HQuGAN experiments for learning Haar random states. The table shows both the mean and standard deviation values for the number of iterations needed for the HQuGAN algorithm (using GRAPE for both players) to reach convergence across 50 Haar random states. The HQuGAN successfully produces all states with a fidelity of 0.999 or higher.

| $n$ | No. of iterations | $T$ | $N$ |
|---|---|---|---|
| 1 | $2.57 \pm 0.70$ | 5 | 50 |
| 2 | $6.9 \pm 2.14$ | 5 | 50 |
| 3 | $12.4 \pm 3.50$ | 5 | 50 |
| 4 | $29.79 \pm 10.18$ | 10 | 100 |
| 5 | $49.95 \pm 26.59$ | 20 | 200 |
| 6 | $89.78 \pm 55.99$ | 30 | 300 |

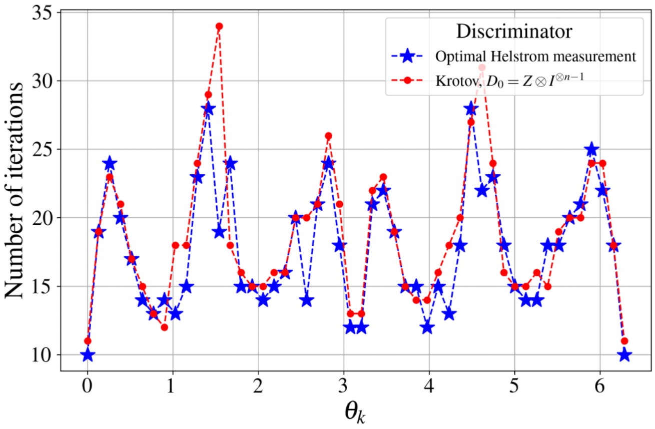

FIG. 3. HQuGANs for learning three-qubit states using Krotov's method. The number of iterations required to generate 50 different three-qubit states described in Eq. (17) for two different discriminators: a discriminator that maximizes $\mathrm{Tr}[D(\rho - \sigma)]$ using Krotov's method (blue) and the optimal discriminator (red). The two cases give similar behaviors.

scriptions of Krotov's method and the numerical results are summarized in Appendix D 2.

### F. Bandwidth limitation

HQuGANs are capable of producing low-bandwidth control fields by introducing the penalty term of Eq. (11) into the generator's cost function. In Appendix E we show that HQuGANs with the penalty terms lead to low-bandwidth optimal control fields, and demonstrate that increasing the evolution time $T$ allows lower bandwidth of the control fields to accomplish the same learning task, numerically verifying the time-bandwidth quantum speed limit in Eq. (12). These results provide concrete tools to estimate the required time for HQuGANs to learn a quantum state using bandwidth-limited control fields, which shape HQuGANs into more experimental-friendly algorithms for current devices.

### G. Learning unitary transformation

In this section we extend the HQuGAN (with GRAPE) to learn an unknown unitary transformation. This problem has been widely studied across a range of settings and techniques [60–65]. We focus on learning a desired unitary operation in two different settings: given a Choi matrix and then pairs of input-output quantum states for the unitary. Both settings are illustrated in Fig. 4.

TABLE III. HQuGANs for learning generalized $n$-qubit GHZ states using Krotov's method. The number of iterations required by the HQuGAN to learn the generalized $n$-qubit GHZ states using Krotov's method. We compare the case of a discriminator using Krotov's method to the case of the optimal discriminator. The two cases require a comparable number of iterations for all instances.

| $n$ | Iter. (Krotov) | Iter. (Helstrom) | $T$ | $N$ |
|---|---|---|---|---|
| 1 | 3 | 3 | 5 | 50 |
| 2 | 6 | 8 | 5 | 50 |
| 3 | 15 | 13 | 5 | 50 |
| 4 | 36 | 25 | 10 | 100 |
| 5 | 62 | 55 | 20 | 200 |
| 6 | 245 | 361 | 30 | 300 |

#### 1. Learning using a given Choi matrix

First, we aim to generate an arbitrary unitary operation $U_{\mathrm{targ}}$ given the Choi matrix for the operation,

$$\mathcal{C}(U_{\mathrm{targ}}) = (I \otimes U_{\mathrm{targ}})|\Omega\rangle\langle\Omega|(I \otimes U_{\mathrm{targ}}^{\dagger}), \tag{19}$$

where $|\Omega\rangle = \frac{1}{\sqrt{d}}\sum_{i=0}^{d-1}|ii\rangle$ is a maximally entangled state. Hence, the HQuGAN sets $\mathcal{C}(U_{\mathrm{targ}})$ as a target state,

$$\min_{\{g\}}\max_{\{d\}} |\mathrm{Tr}[D(\rho(\{g\}) - \mathcal{C}(U_{\mathrm{targ}}))]|^2, \tag{20}$$

where $\rho(\{g\}) = [I \otimes U(\{g\})|\Omega\rangle\langle\Omega|(I \otimes U^{\dagger}(\{g\})]$ is generated by a unitary operator $U(\{g\})$ that the generator creates. The scheme is illustrated in Fig. 4(a). Note that the fidelity between the Choi matrix $\mathcal{C}(U_{\mathrm{targ}})$ and the generator's state $\rho(\{g\})$ is

$$\begin{aligned} F(\mathcal{C}(U_{\mathrm{targ}}), \rho(\{g\}) &= |\langle\Omega|(I \otimes U^{\dagger})(I \otimes U_{\mathrm{targ}})|\Omega\rangle|^2 \\ &= \frac{1}{d^2}\left|\sum_{i,j}\langle i|j\rangle\langle i|U^{\dagger}U_{\mathrm{targ}}|j\rangle\right|^2 \\ &= \frac{1}{d^2}\left|\sum_{i}\langle i|U^{\dagger}U_{\mathrm{targ}}|i\rangle\right|^2 \\ &= \frac{1}{d^2}|\mathrm{Tr}(U^{\dagger}U_{\mathrm{targ}})|^2, \end{aligned} \tag{21}$$

which is 1 if and only if $U$ and $U_{\mathrm{targ}}$ differ only by a global phase, $U = e^{i\phi}U_{\mathrm{targ}}$. Therefore, as the generator learns to generate the Choi matrix, it also learns the target unitary $U_{\mathrm{targ}}$ up a global phase. It is worth noting that a similar approach has been explored in the quantum-assisted quantum compiling algorithm [62], which utilizes a hybrid quantum-classical variational technique to maximize the Hilbert-Schmidt inner product between $U$ and $U_{\mathrm{targ}}$, Eq. (21). The algorithm consists of three main steps: first, it prepares the maximally entangled state on $2n$-qubits starting from $|0\rangle^{\otimes 2n}$; second, it performs both $U$ and $U_{\mathrm{targ}}^{T}$ in parallel; and finally, it measures the state in the Bell basis, where the probability of measuring $|0\rangle^{\otimes 2n}$ corresponds exactly to Eq. (21). While our approach may appear similar to this algorithm, it is inherently distinct as we incorporate a minimax game.

We use the HQuGAN with the cost function of Eq. (20) to generate various unitary operations. We employ the GRAPE algorithm for both players while maintaining the same setups as described in Sec. VI A. The HQuGAN terminates when the fidelity between the Choi matrix and the generator's state exceeds 0.999. We first focus on generating simple one-qubit gates ($X, H, I$, and $T$) and two-qubit gates (CNOT, SWAP, and CZ). The number of iterations required by the HQuGAN to successfully generate each target unitary operation is illustrated in Fig. 5(a). We find that the HQuGAN can generate all unitary operations within 12 iterations. Next, we attempt a

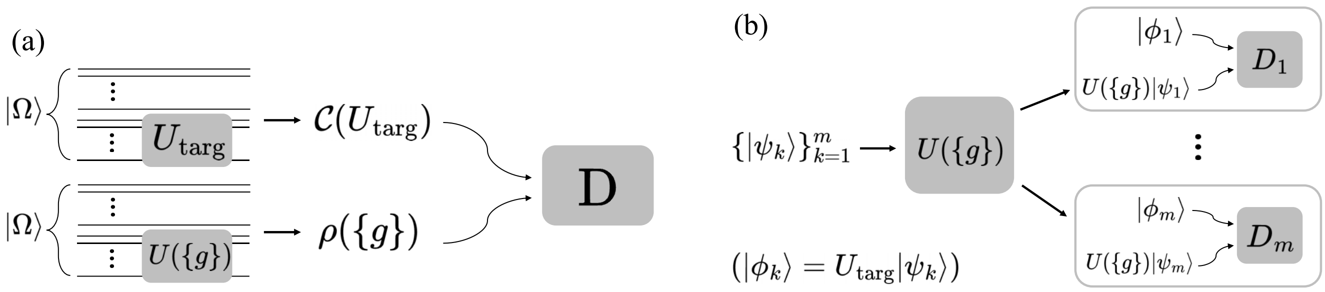

FIG. 4. Two different settings for learning an unknown unitary transformation using HQuGANs. (a) Learning using a Choi matrix. A target unitary transformation $U_{\text{targ}}$ can be represented by a quantum state (or a Choi matrix) $\mathcal{C}(U_{\text{targ}})$ as described in Eq. (19). Hence, the HQuGAN sets the Choi matrix as a target state and plays the minimax game described in Eq. (20). The HQuGAN terminates if the fidelity between the generator's state $\rho(\{g\})$ and the Choi matrix exceeds 0.999, which ensures that the two unitary operations are close up to a global phase, as explained in Eq. (21). (b) Learning using input-output pairs. We are now given $m$ input-output pairs of quantum states for the target unitary $U_{\text{targ}}$: $\{|\psi_k\rangle, |\phi_k\rangle\}$ where $|\phi_k\rangle = U_{\text{targ}}|\psi_k\rangle$. The generator aims to find a unitary $U(\{g\})$ that maps the $m$ input states to their corresponding output states, while $m$ discriminators (denoted as $D_1, \dots, D_m$) each seek to discriminate between the corresponding pairs of quantum states. Therefore, the HQuGAN optimizes a cost function described in Eq. (22), which is a linear combination of $m$ cost functions associated with each input-output pair. If the fidelity between the generator's state and the output state for every exceeds 0.999, the HQuGAN terminates. This guarantees that the minimum fidelity between the two operations also exceeds 0.999.

more challenging task of generating 50 Haar random unitary operations. The optimal Helstrom measurement operator is used for the discriminator. As shown in Fig. 5(b), HQuGANs successfully generate all Haar random unitary operation up to a gate fidelity of 0.999.

### 2. *Learning using pairs of input-output quantum states*

Next, we are given $m$ input-ouput pairs of quantum states for the target unitary $U_{\text{targ}}$: $\{|\psi_k\rangle, |\phi_k\rangle\}_{k=1}^m$ where $|\phi_k\rangle = U_{\text{targ}}|\psi_k\rangle$. Given such pairs, the HQuGAN now optimizes the following cost function:

$$\min_{\{g\}} \max_{\{d\}} \sum_{k=1}^{m} |\text{Tr}[D_k(\{d\})(U(\{g\})\rho_k U^\dagger(\{g\}) - \sigma_k)]|^2, \quad (22)$$

where $\rho_k = |\psi_k\rangle\langle\psi_k|$ and $\sigma_k = |\phi_k\rangle\langle\phi_k|$. Hence, the cost function above is a linear combination of $m$ cost functions associated with each pair of $(\rho_k, \sigma_k)$. The generator tries to find a unitary that maps the $m$ input states to the final states respectively, and $m$ discriminators find each $D_k$ that separates the corresponding pair of quantum states. The scheme is illustrated in Fig. 4(b). The HQuGAN terminates when the fidelity between every pair of the generator's state and the output state exceeds 0.999, i.e.,

$$F_k = F(U(\{g\})|\psi_k\rangle, |\phi_k\rangle) > 0.999 \forall k \in [m]. \quad (23)$$

This will guarantee that the *minimum gate fidelity* [66] between the generator's unitary operation $U(\{g\})$ and the target unitary operation $U_{\text{targ}}$,

$$\min_{|\psi\rangle} F(U(\{g\})|\psi\rangle, U_{\text{targ}}|\psi\rangle), \quad (24)$$

is also greater than 0.999. We use the HQuGAN with the cost function of Eq. (22) to learn various unitary operations. we have kept all setups unchanged from Sec. VI A, and try to learn the same set of unitary operations as those in the previous section: one- and two-qubit gates, as well as 50 Haar random unitary operations up to three qubits. To determine a target unitary up to a global phase, we begin by preparing input-output pairs that can uniquely identify it. The input states are chosen as $\{|0\rangle, \dots, |2^n - 1\rangle, \frac{|0\rangle+|1\rangle}{\sqrt{2}}, \dots, \frac{|2^n-2\rangle+|2^n-1\rangle}{\sqrt{2}}\}$, where $|k\rangle$ is defined as the binary representation of integer $k$, with $|b_i\rangle$ representing a computational basis of qubit $i$. The first $2^n$ input states, $\{|0\rangle, \dots, |2^n - 1\rangle\}$, provide elements to every row of the target unitary matrix up to a phase, while the the rest, $\{\frac{|0\rangle+|1\rangle}{\sqrt{2}}, \dots, \frac{|2^n-2\rangle+|2^n-1\rangle}{\sqrt{2}}\}$, remove the relative phases between each row of the matrix. Therefore, by using these input states and their corresponding output states, we can uniquely identify the target unitary operation up to a global phase.

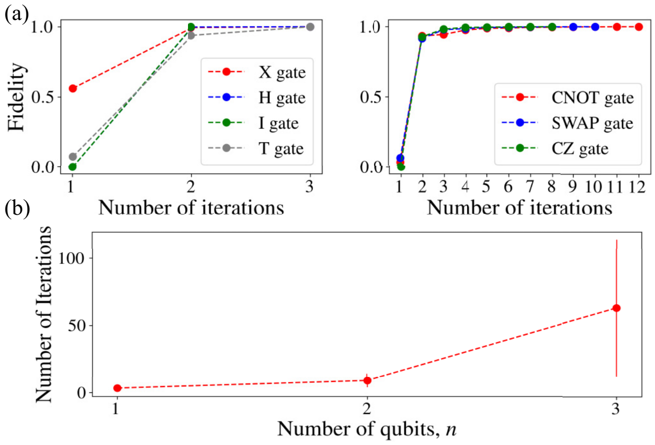

FIG. 5. Learning various unknown unitary operations using Choi matrices. (a) The changes in the fidelity of Eq. (21) during the minimax game described in Eq. (20) are presented. Within 12 iterations, the HQuGAN generate all unitary operations properly. (b) the mean and standard deviation of the number of iterations taken by the HQuGAN to generate 50 Haar random unitary operations are presented. HQuGANs can generate every Haar random unitary operations up to three qubits.

Figure 6(a) shows how the minimum fidelity between the generator's state and the output state for every such pair [i.e., $\min_k F_k$, where $F_k$ is defined in Eq. (23)] changes during the minimax game. Clearly, the HQuGAN is able to generate

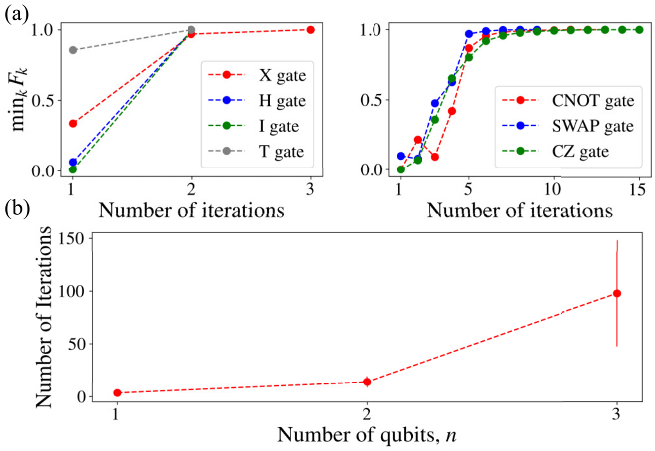

FIG. 6. Learning various unknown unitary operations using input-output pairs of quantum states. (a) The changes in the minimum fidelity between each pair of the generator's state and the output state, considered over all possible pairs [i.e., $\min_k F_k$, where $F_k$ is defined in Eq. (23)] during the minimax game described in Eq. (22) are presented. Within 15 iterations, the HQuGAN generate all unitary operations properly. (b) The mean and standard deviation of the number of iterations taken by the HQuGAN to generate 50 Haar random unitary operations are presented. The HQuGAN successfully generates every target unitary operation up to three qubits.

every gate within 15 iterations. Moreover, Fig. 6(b) indicates that the HQuGAN can successfully produce every Haar random unitary operation up to three qubits. We observe that the mean number of iterations increases exponentially, similar to the previous scenario.

We remark that there exists a trade-off between the number of qubits and the number of distinct discriminators in the two different settings. The first setting requires a $2n$-qubit system to learn an $n$-qubit unitary operation, as well as the ability to prepare a maximally entangled state every time the generator's state and the Choi matrix are prepared. The second setting does not require any additional qubits, but it demands an exponentially large number of input-output pairs of quantum states, implying that exponentially many distinct discriminators are required.

## VII. COST FUNCTION

In this section we provide additional numerical simulations to understand the role of cost functions in the convergence of HQuGANs.

### A. Mode collapse

Recently, it has been observed that the loss in the minmax game

$$\min_{\theta_g}\max_{D}\mathrm{Tr}[D(\sigma-\rho(\theta_g))] \tag{25}$$

can oscillate between a few values and thus the game may never converge to the desired Nash equilibrium point, a phenomenon called *mode collapse* [10]. In the case of Eq. (25) the fundamental reason for the mode collapse can be understood from the form of the cost function, where the generator's optimization is independent of the target state $\sigma$. When the generator minimizes $-\mathrm{Tr}[D\rho(\theta_g)]$ or equivalently maximizes $\mathrm{Tr}[D\rho(\theta_g)]$, independent of $\sigma$, there is a possibility of overshooting by selecting a generator $\rho(\theta_g)$ that aligns with $D$ [10]. If $D$ is chosen to be a previous generator's state, then the generator's minimization will output the same quantum state, falling into a loop, which prevents the game from converging. This is in agreement with the results in classical machine learning, where the generators of classical GANs tend to characterize only a few modes of the true distribution, but can miss other important modes [36]. In fact, we observe that mode collapse occurs in almost all instances of the HQuGAN simulations using the cost function in Eq. (25).

To address this issue, we also use the following cost function:

$$\min_{\theta_g}\max_{D}|\mathrm{Tr}[D(\sigma-\rho(\theta_g))]|^2. \tag{26}$$

As was discussed in Sec. V, this cost function still guarantees the existence of Nash equilibrium at the same location as before, i.e., at $\rho=\sigma$. In addition, this choice of cost function guarantees that the generator's quantum state minimizing the cost function is underdetermined. Therefore there are typically infinitely many states reaching the maximum of the cost function, which makes the mode collapse measure zero. In Appendix C we provide a detailed explanation for a one-qubit example in addition to numerical experiments.

We have numerically verified that when using the original cost function shown in Eq. (25), the global Nash equilibrium cannot be reached for any instance of our numerical experiments. However, when we use the modified cost function presented in Eq. (26), the equilibrium point is always properly reached. This result is summarized in Fig. 7, where we present the change in fidelity between the generator's state and the corresponding target state as the HQuGAN proceeds the minimax game. Here we aim to generate 50 Haar random qubits states up to five qubits and use the GRAPE algorithm for both the generator and discriminator (the system setup remains unchanged compared to the Sec. VI). The blue lines indicate the results produced by using the modified cost function, which successfully generates each target state with a fidelity of 0.999. However, the original cost function represented by the red lines fails to generate any states within the desired fidelity. When considering the one-qubit result, all instances fall into mode collapse within the first 10 iterations, as shown in the graph on the far left. In other cases, the fidelity fluctuates and fails to converge to the desired value. It is evident that for four- and five-qubit cases, it is uncommon to achieve even a relatively high fidelity when using the original cost function. Moreover, even if it does achieve relatively high fidelity, it eventually falls into mode collapse, resulting in the generator repeatedly producing only two quantum states. These states have an overlap of approximately 0.75 with the target state, as indicated in the figure. Therefore, we have used the modified cost function for all numerical experiments presented in this work.

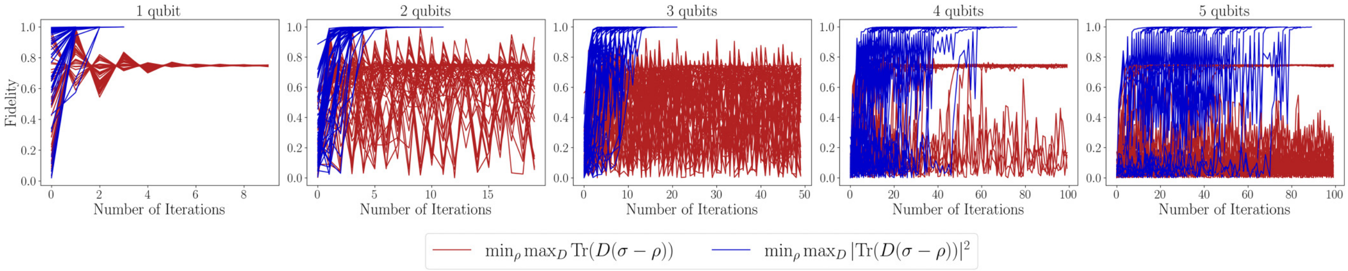

FIG. 7. A comparison between two different cost functions of Eq. (25) and Eq. (26) for learning Haar random states. We compare the performance of HQuGANs using GRAPE (for both players) with two different cost functions to generate 50 Haar random quantum states up to five qubits. By using the modified cost function (blue lines), each state can be generated with a fidelity of 0.999, whereas the original cost function (red lines) fails to produce any state correctly. Moreover, we notice that using the original cost function eventually falls into mode collapse, resulting in the generator oscillating between two quantum states indefinitely.

### B. Quantum Wasserstein distance of order 1

As discussed earlier, the dual form of the quantum $W_1$ distance makes it possible to express the learning task in terms of the minimax game described in Eq. (3). It has been shown that such quantum Wasserstein GAN (qWGAN) exhibits more favorable loss landscapes compared to other conventional metrics such as fidelity in learning specific quantum states [8]. An intuition behind such advantage lies in the fact that while common (unitary invariant) metrics such as fidelity capture only the global properties of quantum states (which can cause barren plateaus [39]), the quantum $W_1$ distance is sensitive to local operations. In fact, the cost function described in Eq. (14) with $\|D\|_\infty \leqslant 1$ is precisely the trace distance squared, which is unitary invariant. Hence, we expect that the quantum $W_1$ distance can give a faster convergence rate compared to Eq. (14), similar to what was observed in [8].

We thus explore the performance of HQuGANs using the quantum $W_1$ distance to learn up to the six-qubit GHZ state. Figure 8 compares the number of iterations of HQuGANs using the Lipschitz discriminator that calculates the quantum $W_1$ distance (blue lines) to the optimal discriminator (red lines) and the quantum optimal control discriminators (green lines) that exploit both GRAPE and Krotov's method. For generating five- and six-qubit GHZ states, we observe that the HQuGAN using the quantum $W_1$ distance converges faster.

### C. Hybrid cost functions

Depending on the nature of the learning task, using multiple cost functions can be more advantageous than using one cost function. Here we discuss one such scenario.

Although the cost function (25) can lead to mode collapse in the long run, in Appendix B we show analytically that after using it only for the first two iterations, the generator generates a state that is relatively close to the target state. To avoid the mode collapse we can then switch the cost function to Eq. (14), which robustly improves the fidelity to the desired value. Our numerical experiments show that using such a combined method can generate up to the eight-qubit GHZ state using extremely smaller numbers of iterations compared to previous results in Table I. For example, while using the single cost function of Eq. (C1) takes ∼120 iterations to generate the six-qubit GHZ state as illustrated in Table I, the combined method takes only four iterations to generate the eight-qubit GHZ state. Furthermore, while generating six-qubit Haar random states on average requires ∼90 iterations, as demonstrated in Table II, the hybrid approach requires an average of only ∼7 iterations to generate (up to) eight-qubit Haar random quantum states. We discuss more details on the analytical descriptions and numerical results in Appendix B.

## VIII. IMPLEMENTATION ON A QUANTUM COMPUTER

To perform gradient-based quantum optimal control techniques such as GRAPE, in addition to estimating the value of the cost function, we need to estimate the gradients of the pulses. Calculating the gradients can quickly become intractable as the system size grows due to *the curse of dimensionality*. To remedy this bottleneck, one can use quantum computers to directly estimate not only the cost function but also the gradients of control pulses required [67]. A similar method called *parameter-shift rules* [68,69] has been widely

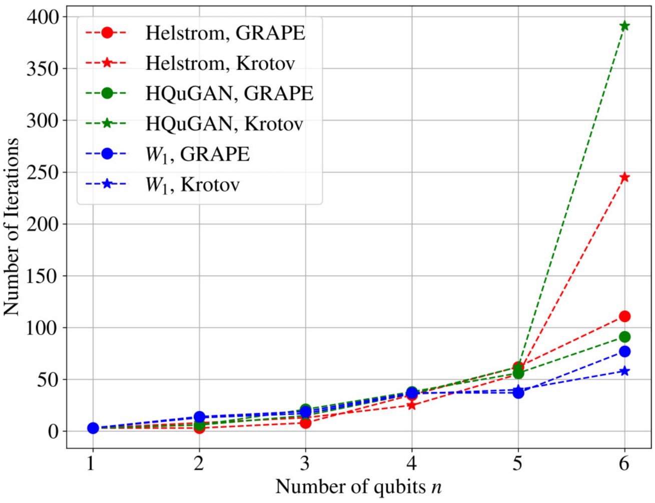

FIG. 8. HQuGANs for learning ***n***-qubit GHZ states using the quantum $\boldsymbol{W_1}$ distance. Comparing the number of iterations required by the HQuGAN to learn generalized GHZ states using the quantum $W_1$ distance discriminator to the previously discussed discriminators. The utilization of the quantum $W_1$ distance leads to a smaller number of iterations for learning $n = 5$ and six-qubit GHZ states compared to other cases.

used in the circuit model variational quantum algorithms to evaluate the gradients of cost functions. Following the same approach, in this section, we show how one can directly incorporate the GRAPE algorithm into the implementation of the HQuGAN and then analyze the computational costs of the quantum algorithm, such as sample complexity and other classical/quantum resources.

### A. Estimation of gradients

The gradients for the generator's cost function in Eq. (14), to the first order of $\Delta t$, is [67]

$$\begin{aligned}\frac{\partial}{\partial\epsilon_i(t_j)}&|\mathrm{Tr}[D(\sigma-[\sigma-\rho(\{g\}])]|^2\\ &=2\Delta t(\mathrm{Tr}[D(\rho(\{g\})-\sigma)])\big[\mathrm{Tr}\big(D\rho_{i_+}^{kj}\big)-\mathrm{Tr}\big(D\rho_{i_-}^{kj}\big)\big],\end{aligned} \tag{27}$$

where $\rho_{i_\pm}^{kj}=U(t_N)\cdots U(t_{j+1})R_\alpha^k(\pm\pi/2)U(t_j)\cdots U(t_1)\rho_0$ $[U(t_N)\cdots U(t_{j+1})R_\alpha^k(\pm\pi/2)U(t_j)\cdots U(t_1)]^\dagger$ and $R_\alpha$ is a (single-qubit) rotation around $\alpha$ axis, corresponding to the Hamiltonian term in front of $\epsilon_i$. Hence, the gradient at time $t_j$ can be calculated by estimating expectation values of operator $D$ with respect to two quantum states $\rho_{i_+}^{kj}$ and $\rho_{i_-}^{kj}$. These quantum states can be prepared by implementing three unitary transformations on a quantum annealer: (1) $U(t_N,t_{j+1})$, (2) $R_\alpha^k(\pm\pi/2)$, and (3) $U(t_j,t_1)$, where $U(t_m,t_n)$ is a unitary from time $t_n$ to $t_m$. Likewise, calculating cost functions require only use of the annealer once. Note that penalty terms ($J_P$) can be efficiently calculated via classical computers.

### B. Complexity analysis

HQuGANs find parameters of a given time-dependent Hamiltonian to generate an unknown quantum state. The closest, but not necessarily directly comparable, approach to accomplish the same task is to simply perform quantum state tomography (QST) on the unknown state to obtain the full classical descriptions of the state and then perform an optimal control method to find the parameters of the Hamiltonian that generates the state. Each step of this approach requires exponential, in the number of qubits, resources to learn general quantum states. The proposed HQuGAN framework provides an alternative method to generate the unknown quantum state directly, without using the classical description of the state. To examine this more rigorously, we analyze the computational cost of the HQuGAN using GRAPE.

It is important to note that both the sample complexity and classical postprocessing time for the HQuGAN are proportional to the total number of iterations of the algorithm which is unknown in general even for classical GANs. (While the computational complexity of solving approximate local solutions in GANs has been studied [70], a precise bound for global Nash equilibria is not known.) This sets a barrier to comparing the complexity of the HQuGAN to other existing algorithms such as QST in adaptive measurement settings [71,72]. Characterizing the set of quantum states that can provably be learned more efficiently using the direct approach of the HQuGAN framework compared to the QST approach is an open question.

#### 1. *Sample complexity*

In terms of sample complexity, recall that we need to estimate four distinct expectation terms in calculating Eq. (27), where each estimation takes $O(\|D\|^2/\epsilon^2)$ copies of $\sigma$, $\rho$, $\rho_{i+}^{kj}$ or $\rho_{i-}^{kj}$ with precision $\epsilon$. Similarly, the discriminator takes $O(\|D\|^2/\epsilon^2)$ copies of all four states. In our setting, a single optimization step of the generator then requires $O(N/\epsilon^2)$ copies of all four states in order to estimate the gradients for all the $N$ time grids/Trotter number ($\because \|D\|^2=1$). For Trotter error $\delta$ and a fixed evolution time $T$, we need the Trotter number $N=O(nT^2/\delta)$ in the first-order product formula [73]. Although a rigorous (global) convergence rate for the optimization required for the optimal control of the generator and discriminator is unknown in general, a favorable scaling is expected for generic problems with a high number of control parameters [74,75].

#### 2. *Classical storage*

The maximum storage amount that the HQuGAN requires is proportional to the size of the gradient vector (or Hessian matrix if we utilize quasi-Newton optimization methods) of the cost function at each time for all the control pulses. Since the HQuGAN can forget about past gradient values, the required classical storage is $O(\mathrm{poly}(n))$, independent of the number of iterations. This shows a substantial improvement compared to the classical storage required by quantum state tomography (QST) or self-guided QST [76], which is (at least) exponential in $n$. Hence, the HQuGAN could prove valuable when we need to generate an unknown quantum state without the need to store an extensive classical memory for its classical representation.

## IX. CONCLUSION

We have introduced a framework to learn arbitrary (unknown) quantum states using two competing optimal control techniques. This framework leverages techniques from QuGANs and quantum optimal controls (QOC), leading to insights and methodology for learning unknown quantum states under time-continuous dynamics. Applying optimal control directly to the time-dependent Hamiltonian explores a larger set of unitaries than the gate-based approach and is applicable to a wide variety of quantum information processing platforms such as superconducting processors, ion-trap quantum computers, and diabatic quantum annealers. We demonstrated the capabilities of the proposed framework by performing numerical experiments to generate various many-body quantum states using the two popular gradient-based QOC methods, GRAPE and Krotov's method, under experimentally realistic constraints on pulse amplitudes and bandwidths. Also, we extended the HQuGAN to learn quantum processes.

We discussed the role of the cost function in reaching the equilibrium point, by avoiding mode collapse and also the convergence rate. We provided numerical experiments that show that the quantum $W_1$ distance gives faster convergence of the minimax game when generating generalized GHZ states for higher system sizes. Moreover, we observe that exploiting multiple forms of cost functions properly could give a large advantage in terms of the rate of convergence of the algorithm. Since computations for QOC methods become intractable as system size increases, we remark that estimating the required cost functions and their gradients can be directly incorporated into the HQuGAN framework.

A promising direction to extend this work would be considering the effect of noise and control errors in preparing pure or more generally mixed quantum states. Also, it is natural to extend our unitary learning procedures to learn quantum channels. Given the iterative nature of the algorithm, we expect the framework to provide a natural robustness to noise, similar to what has been observed in variational quantum circuits [77]. From an algorithmic perspective, finding rigorous bounds on the total iterations of the minimax game is a major open problem.

## ACKNOWLEDGMENTS

This work has been supported by DARPA's Reversible Quantum Machine Learning and Simulation (RQMLS) program under Agreement No. HR00112109969. M.M. is supported by the NSF Grant No. CCF1954960. The authors would like to thank Nicholas Ezzell, Bobak Kiani, and Daniel Lidar for helpful discussions, Benjamin Anker for assisting with the codes, and the UNM Center for Advanced Research Computing, supported in part by the NSF, for providing parallel computing used in this work.

## APPENDIX A: TRAINING A GENERATOR AND A DISCRIMINATOR USING GRAPE

We give a detailed explanation of how the generator and the discriminator are trained with respect to the GRAPE method, i.e., lines 6 and 11 in Algorithm 1. Here we focus on the minimax cost function we used to obtain our main numerical results in Sec. VI:

$$\min_{\{g\}} \max_{\{d\}} |\mathrm{Tr}[D(\{d\})(\sigma - \rho(\{g\})]|^2, \tag{A1}$$

where $\{g\}$ and $\{d\}$ indicate control fields for the generator and the discriminator, respectively. Let $N$, $T$, $\Delta t$, and $H(t) = H_0 + \sum_k \epsilon_k(t) H_k$ denote the Trotter step, the evolution time, $T/N$, and the time-dependent Hamiltonian given to either the generator or the discriminator with control fields $\epsilon_k(t)$. The GRAPE algorithm discretizes the time domain into small pieces and approximates the Hamiltonian to be time-independent. We denote $H(t_j) = H_0 + \sum_k \epsilon_k(t_j) H_k$ by the time-independent Hamiltonian within interval $[t_j, t_j + \Delta t]$.

### 1. Training a generator

The GRAPE method requires a gradient of the cost function with respect to the control field $g_i$ at a time grid $t_j$ $\forall i \in [1, m]$, $\forall j \in [1, N]$, which can be expressed as

$$\frac{\partial}{\partial g_i(t_j)} |\mathrm{Tr}[D(\sigma - \rho(\{g\}))]|^2$$
$$= 2|\mathrm{Tr}[D(\sigma - \rho(\{g\}))]| \frac{\partial}{\partial g_i(t_j)}$$
$$\times\ (\mathrm{Tr}[D(\sigma - \rho(\{g\}))])$$
$$= 2|\mathrm{Tr}[D(\sigma - \rho(\{g\}))]| \frac{\partial}{\partial g_i(t_j)} \mathrm{Tr}[D\rho(\{g\})], \tag{A2}$$

where the last equality is because the target state $\sigma$ is independent of the generator's control fields $g_i(t_j)$ $\forall i, j$. In the GRAPE method, we usually approximate the gradient to the first order of $\Delta t$ [21]:

$$\approx 2|\mathrm{Tr}[D(\sigma - \rho)](-i\Delta t \mathrm{Tr}[D_j[H_i, \rho_j]]), \tag{A3}$$

where $D_j = U^\dagger(t_{j+1}) \dots U^\dagger(t_N) D U(t_N) \dots U(t_{j+1})$, $\rho_j = U(t_j) \dots U(t_1)\rho_0 U^\dagger(t_1) \dots U^\dagger(t_N)$, and $U(t_j) = \exp[-i\Delta t H(t_j)]$. Hence, training the generator consists of two steps. First, we calculate and store $\rho_j$ $\forall j \in [1, N]$, and similarly for $D_j$ $\forall j \in [1, N]$. Then we update control fields for all time grids by calculating Eq. (A3) $\forall i, j$. The update procedure is repeated until termination criterion is achieved, which is when either the norm of the gradient or the objective function $|\mathrm{Tr}[D(\rho - \sigma)]|^2$ is less than $10^{-5}$. Although these calculations are performed classically, it is possible to reformulate the gradient equation Eq. (A3) in a manner that allows for the use of quantum computers to compute gradients, as described in Eq. (27) [67].

### 2. Training a discriminator

Recall that the discriminator aims to find a sequence of control fields $\{d\}$ that generate a unitary transformation $U(\{d\})$ before a fixed measurement operator $D_0$, which maximizes Eq. (A1). This is equivalent to

$$\max_{\{d\}} \mathrm{Tr}[U^\dagger(\{d\}) D_0 U(\{d\})(\rho - \sigma)]. \tag{A4}$$

By cyclic property of trace, the objective function is identical to

$$\mathrm{Tr}[D_0 U(\{d\})(\rho - \sigma)U^\dagger(\{d\})], \tag{A5}$$

which can be viewed as an expectation value of the fixed measurement operator $D_0$ with respect to a time-evolved state from an initial state of $\rho - \sigma$. Hence, similar to the previous section, we can approximate the gradient to the first order of $\Delta t$:

$$\frac{\partial}{\partial d_i(t_j)} \mathrm{Tr}[D_0 U(\{d\})(\rho - \sigma)U^\dagger(\{d\})] \tag{A6}$$
$$\approx -i\Delta t \mathrm{Tr}[D_j[H_i, (\rho - \sigma)_j]], \tag{A7}$$

where $D_j = U^\dagger(t_{j+1}) \dots U^\dagger(t_N) D U(t_N) \dots U(t_{j+1})$, $\rho_j = U(t_j) \dots U(t_1)(\rho - \sigma)U^\dagger(t_1) \dots U^\dagger(t_N)$, and $U(t_j) = \exp[-i\Delta t H(t_j)]$. Hence, similar to the generator's turn, training the discriminator consists of two steps. First we calculate and store $(\rho - \sigma)_j$ $\forall j \in [1, N]$ and similarly for $D_j$ $\forall j \in [1, N]$. We then update control fields for all time grids by calculating Eq. (A6) $\forall i, j$. The update procedure is repeated until termination criteria is achieved, which is the norm of the gradient is less than $10^{-5}$. It is worth noting that the convergence criteria for the discriminator should be based solely on the norm of the gradient, rather than the value of the cost function. This is because, in practice, it is not feasible to obtain knowledge of the extreme value of the cost function. We also remark that, like the generator, the gradient of the discriminator can also be computed using quantum computers. This can be accomplished by substituting $\rho - \sigma$ for $\rho_0$ in Eq. (27).

### 3. Evaluating fidelity between generated and target states

After each round of the minimax game between the generator and the discriminator is completed, it is important to calculate the fidelity between the generated state and the target state to determine the convergence of the HQuGAN algorithm. The discriminator's cost function can be used to obtain this fidelity: if the discriminator successfully maximizes its cost function, it will ultimately become equivalent to the trace distance between the two quantum states. However, it should be noted that there is no guarantee that the discriminator will converge to the optimal Helstrom measurement operator that fully maximizes its cost function. Nonetheless, as mentioned in Sec. VI, numerical evidence suggests that the discriminator always approaches this optimal measurement operator. This enables it to converge to the trace distance between the two quantum states, which can be utilized to calculate the fidelity (for pure states). When working with mixed states, the trace distance can be used as the figure of merit instead of fidelity.

## APPENDIX B: USING HYBRID COST FUNCTIONS

In this Appendix we suggest a method to speed up the convergence of the proposed HQuGAN by using two different cost functions. We first observe that the measurement operator $D$ that maximizes $\mathrm{Tr}[D(\rho - \sigma)]$ can be chosen to be proportional to $\rho - \sigma$, if $\rho$ and $\sigma$ are pure quantum states.

The constant of proportionality depends on the Schatten $p$-norm constraint on $D$ (i.e., $||D||_p \leqslant 1$) and a positive eigenvalue of $\rho - \sigma$.

*Lemma 1.* For any $p$ and any two pure states $\rho$ and $\sigma$, the maximum of the cost function

$$\max_{||D||_p \leqslant 1} \mathrm{Tr}[D(\rho - \sigma)] \tag{B1}$$

can be achieved by the following operator:

$$D^* = 2^{-\frac{1}{p}}(|P+\rangle\langle P+| - |P-\rangle\langle P-|) = \frac{2^{-\frac{1}{p}}}{\lambda}(\rho - \sigma), \tag{B2}$$

where $|P\pm\rangle\langle P\pm|$ are projection operators onto positive and negative eigenspaces of $\rho - \sigma$ respectively, and $\pm\lambda$ are eigenvalues of $\rho - \sigma$.

*Proof.* It is straightforward to check that the proposed $D^*$ saturates the upperbound posed by Hölder's inequality,

$$\mathrm{Tr}(AB) = ||A||_p ||B||_q, \tag{B3}$$

for any $p$ and $q$ satisfying $1 - \frac{1}{p} = \frac{1}{q}$. First using the definitions we have $||\rho - \sigma||_q = 2^{\frac{1}{q}}\lambda$ and

$$||D^*||_p = (|2^{-\frac{1}{p}}|^p + |2^{-\frac{1}{p}}|^p)^{1/p} = 1. \tag{B4}$$

The proof is then completed by noting that

$$\begin{aligned} \mathrm{Tr}[D^*(\rho - \sigma)] &= \mathrm{Tr}[2^{-\frac{1}{p}}(|P+\rangle\langle P+| - |P-\rangle\langle P-|) \\ &\quad \times \lambda(|P+\rangle\langle P+| - |P-\rangle\langle P-|)] \\ &= 2^{-\frac{1}{p}} 2\lambda \\ &= 1 \times 2^{1-\frac{1}{p}}\lambda \\ &= ||D^*||_p ||\rho - \sigma||_{\frac{1}{1-\frac{1}{p}}}. \end{aligned} \tag{B5}$$

TABLE IV. Learning Haar random and generalized GHZ states using HQuGANs with a hybrid cost function. The table summarizes the number of iterations required to learn generalized $n$-qubit Haar random and GHZ states using the HQuGAN with a hybrid cost function. We generate 50 Haar random states and report the mean and the standard deviation of number of iterations. Using the hybrid cost function requires substantially fewer iterations than using a single cost function (see Tables I and II). The generator is optimized using the GRAPE algorithm. $T$ and $N$ refer to the evolution time and the Trotter number, respectively.

| $n$ | Iter. (GHZ) | Iter. (Haar) | $T$ | $N$ |
|---|---|---|---|---|
| 5 | 4 | $7.44 \pm 3.11$ | 20 | 200 |
| 6 | 5 | $6.24 \pm 4.32$ | 30 | 300 |
| 7 | 4 | $6.26 \pm 3.67$ | 40 | 400 |
| 8 | 4 | $5.55 \pm 10.29$ | 50 | 500 |

Now consider the HQuGAN optimizing a minimax game described in Eq. (2),

$$\min_{\rho} \max_{D} \mathrm{Tr}[D(\rho - \sigma)], \tag{B6}$$

where a target state $\sigma$ and an initial choice for $D$ are arbitrarily chosen. We consider the optimal discriminator that analytically calculates her operator via Eq. (B2). Here we argue that, after two rounds of interactions between two players, the generator will output a quantum state with high fidelity to the target state.

In the first round, the generator minimizes $\mathrm{Tr}(D\rho_1)$, which will output a random quantum state $\rho_1$ as $D$ is initialized randomly. In the next round, the optimal discriminator chooses $D_1 \propto (\rho_1 - \sigma)$ as shown in Lemma 1. The generator then tries to find $\rho_2$ such that $\mathrm{Tr}(D_1\rho_2) \propto \mathrm{Tr}[(\rho_1 - \sigma)\rho_2]$ is minimized. Since $\rho_1$ and $\sigma$ are random quantum states, the fidelity between the two states would generically be exponentially small. Therefore, assuming that the generator always outputs a pure state, $\rho_2 \approx \sigma$. As a consequence, the generator at the second round of the algorithm already gives a quantum state that achieves a high fidelity with the target state. To avoid the described mode collapse in future rounds, and improve the fidelity to any desired accuracy, we then switch our cost function to Eq. (14). Subsequently, we can combine two different cost functions to make the algorithm converge faster.

We numerically confirm that using such a protocol, the HQuGAN successfully generates up to eight-qubit Haar random and GHZ states with substantially fewer iterations. As shown in Table IV, it took only four iterations to generate the eight-qubit GHZ state whereas using a single cost function of Eq. (14) described in the main part (see Table 1) took ∼90 iterations to generate the six-qubit GHZ state. Furthermore, it took on average only ∼7 iterations to generate up to seven-qubit Haar random quantum states, whereas using a single cost function took ∼90 iterations for generating six-qubit Haar random states. In fact, Fig. 9 shows the mean and the standard deviation of the fidelity between the generator's state

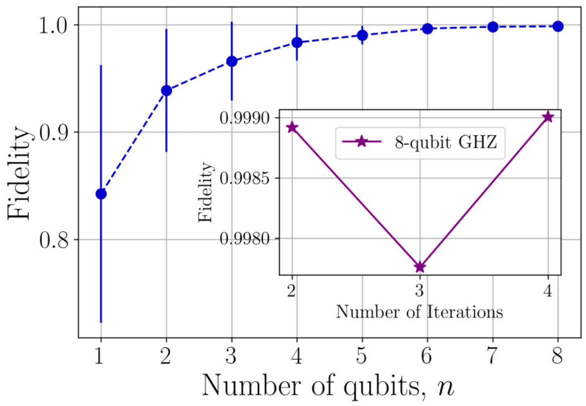

FIG. 9. Fidelity between the generator's state and the target state after two rounds of the minimax game. We plot the mean and the standard deviation of the fidelity between the generator's state and the target state after two rounds of the minimax game for learning 50 Haar random states (blue). As expected, the generator generates a state with a high fidelity with the target state after two rounds of the game, and this fidelity approaches 1 as the system size increases. For example, when generating eight-qubit GHZ state (purple), the generator after the second iteration already gives a very high fidelity ≈0.999. Thus, only two more iterations are needed to achieve a desired fidelity >0.999.

and the target state after two rounds of the minimax game for learning 50 Haar random states. The fidelity approaches 1 as the system size increases. As previously stated, the primary reason for this is that two random states, $\rho_1$ and $\sigma$, in general have small overlaps, which decreases exponentially as the system size increases. Therefore, the generator in the second round of the game will have more support on $\sigma$ rather than $\rho_1$ as the system size grows. In fact, as shown in the figure, the fidelity for the six- seven-, and eight-qubit system is approximately 0.9965, 0.9976, and 0.9986 respectively. As a result, by switching the cost function to Eq. (14), it takes only a few additional iterations to achieve the desired fidelity of 0.999.

When the discriminator uses quantum optimal control methods to find the measurement, we can still achieve the advantage by initializing $D_0$ as a rank-2 matrix because the rank-2 Helstrom measurement is unique (we have numerically validated this in Fig. 2.) However, if $D_0$ has a higher rank, the aforementioned advantage cannot be necessarily achieved. This is due to the fact that the optimal choice of the discriminator will have a higher rank than two, and therefore the generator's state in the next round can have support on eigenvectors of the discriminator that are not proportional to $\rho_1 - \sigma$. It is also worth noting that if either the generator's state or the target state is a mixed state, then the hybrid approach no longer accelerates convergence, as the optimal Helstrom measurement must have a rank greater than 2.

## APPENDIX C: MODE COLLAPSE

A QuGAN might not always converge to a good Nash equilibrium point due to the mode collapse phenomenon. In this Appendix we review this issue raised in Ref. [10] and study how alternative cost functions can remedy this problem. To be consistent with the notations used in Ref. [10], we assume all discriminator operators $D$ are POVMs (only in this section).

### 1. Mode collapse example

Below we review a concrete example of mode collapse presented in [10], by considering the minimax game

$$\min_{\theta_g} \max_{D} \mathrm{Tr}[D(\sigma - \rho(\theta_g))]. \tag{C1}$$

It is important to note that for this cost function, the generator always tries to align his state $\rho$ with the discriminator's previous operator $D$, independent of the target state $\sigma$, and therefore can overshoot. More concretely, starting from the following generator's initial state $\rho$ and target state $\sigma$:

$$\sigma = \frac{1 + \cos(\pi/6)X + \sin(\pi/6)Y}{2}, \tag{C2}$$

$$\rho = \frac{1 + \cos(\pi/6)X - \sin(\pi/6)Y}{2}, \tag{C3}$$

the players will follow the following steps:

*Step 1 (Discriminator)*: Since $\sigma - \rho = Y/2$, the optimal Helstrom measurement operator is $D_1 = P^+(\sigma - \rho) = (1 + Y)/2$.

*Step 2 (Generator)*: Given $D_1$, the generator tries to minimize (C1), or equivalently to maximize $\mathrm{Tr}[D_1\rho(\theta_g)]$. If we set $\rho_1 = (1 + a_xX + a_yY + a_zZ)/2$, then $\mathrm{Tr}(D_1\rho_1) = 1/2 + a_y/2$ is maximized when $a_y = 1$, which yields $\rho_1 = D_1 = (1 + Y)/2$ as the unique solution.

*Step 3 (Discriminator)*: The optimal Helstrom measurement operator is $D_2 = P^+(\sigma - \rho_1) = \rho$.

*Step 4 (Generator)*: Again, the generator tries to align his state $\rho_2$ with $D_2 = \rho$ to maximize $\mathrm{Tr}(D_2\rho_2)$, which is achieved uniquely by choosing $\rho_2 = \rho$. Therefore, we are back to *Step 1* and the algorithm repeats forever.

### 2. Cost function with absolute value

In this section we consider a cost function that is minimized only if the generated state has the same expected value as the target state $\sigma$ with respect to the discriminator $D$,

$$\min_{\theta_g} \max_{D} |\mathrm{Tr}[D(\sigma - \rho(\theta_g))]|^2, \tag{C4}$$

and first show that this cost function can prevent the mode collapse issue discussed in the previous section.

*Step 1 (Discriminator)*: This round is the same as the previous section; the (optimal) Helstrom measurement operator is $D_1 = P^+(\sigma - \rho) = (1 + Y)/2$.

*Step 2 (Generator)*: Unlike before, the generator this time tries to find $\rho(\theta_g)$ such that $\mathrm{Tr}[D_1\rho(\theta_g)]$ is equal to $\mathrm{Tr}(D_1\sigma) = 3/4$. Since $\mathrm{Tr}(D_1\rho_1) = 1/2 + \mathrm{Tr}(\rho_1 Y/2)$ must be $3/4$, any $\rho_1$ satisfying $\mathrm{Tr}(\rho_1 Y/2) = 1/4$ is a solution or equivalently any $\rho_1 = (1 + a_xX + Y/2 + a_zZ)/2$ where $a_x^2 + a_z^2 = 3/4$ (assuming unitary dynamics) satisfies this constraint. Note that in contrast to the previous section, there are infinitely many values of $a_x$ and $a_z$ that the generator chooses to produce $\rho_1$, and therefore the probability of a random choice of generator in the next round producing $\rho_2 = \rho$ is zero.

It is important to remark that the modified cost function in Eq. (C4) can cure mode collapse more generally, beyond the example shown above. The main observation is that the equation $\mathrm{Tr}(D_i\rho_i) = \mathrm{Tr}(D_i\sigma)$ is always underdetermined, and therefore there are infinitely many $\rho_i$ satisfying this equation. [The only exception is when $D_i$ is rank one, and $\mathrm{Tr}(D_i\rho_i) = \mathrm{Tr}(D_i\sigma) = 1$, but this is only possible if $D_i = \sigma = \rho_i$, which is the desired fixed point.] Although there are infinitely many choices for $\rho_i$, mode collapse only occurs when $\rho_i = \rho_{i-2}$. This indicates that the set of states that cause mode collapse using the modified cost function has measure zero. Indeed, using the modified cost function, mode collapse is never observed in our numerical experiments.

## APPENDIX D: HQUGANS USING KROTOV'S METHOD

In this Appendix we describe more details about Krotov's method [22] as well as additional numerical experiments using HQuGANs with Krotov's method. Krotov's method rigorously investigates the conditions for updating a time-continuous control field such that the updated field guarantees to decrease a cost function. To achieve this task, the method carefully updates a control field at time $t_j$ based on all of the control fields at $t_k$ for $\forall k < j$ that are previously updated. This guarantee of monotonic convergence for time-continuous control fields is what distinguishes Krotov's method from other quantum optimal control methods. We discuss Krotov's method more in detail below.

### 1. Krotov's method

Krotov's method [22] is based on the rigorous examination of the conditions for calculating the updated control fields such that it always guarantees *monotonic convergence* of $J[\{\epsilon_i(t)\}]$ by construction. Krotov's method considers a more standard form of the cost functional [22],

$$J[\{\epsilon_i(t)\}, \{|\psi(t)\rangle\}] = J_T[\{|\psi(T)\rangle\}] + \int_0^T dt\; g[\{\epsilon_i(t)\}, \{\rho(t)\}, t], \quad \text{(D1)}$$

where $J_T$ is the main objective functional that depends on the final time $T$ [e.g., $F$ in Eq. (7)] and $g = g_a[\{\epsilon_l(t)\}, t] + g_b[\{|\psi_k(t)\rangle, t]$ captures additional costs or constraints at intermediate times, for instance, by restricting the field spectra or by penalizing population in certain subspaces.

To minimize the field intensity and to smoothly switch the field on and off, $g_a$ can be chosen to be in the following form [59]:

$$g_a[\{\epsilon(t)\}] = \frac{\lambda}{S(t)}[\epsilon(t) - \epsilon_{\mathrm{ref}}(t)]^2, \quad \text{(D2)}$$

where $\epsilon_{\mathrm{ref}}(t)$ denotes some reference field, $S(t)$ is a shape function and $\lambda$ is a step size (we discuss more details on these parameters later). Given such a specific choice of the functional $g_a$, Krotov's method updates control fields [22,78]

$$\Delta\epsilon_i^{(k)}(t) = \frac{S_i(t)}{\lambda_i}\mathrm{Im}\left[\langle\chi^{(k-1)}(t)\left|\left(\frac{\partial H}{\partial\epsilon_i(t)}\right)\right|_{(k)}|\phi^{(k)}(t)\rangle\right]. \quad \text{(D3)}$$

$|\phi^{(k)}(t)\rangle$ and $|\chi^{(k-1)}(t)\rangle$ are forward-propagated and backward-propagated under the guess controls $\{\epsilon_i^{(k-1)}(t)\}$ and optimized controls $\{\epsilon_i^{(k)}(t)\}$ in each iteration $k$, respectively, through

$$\frac{\partial}{\partial t}|\phi^{(k)}(t)\rangle = -\frac{i}{\hbar}H^{(k)}|\phi^{(k)}(t)\rangle, \quad \text{(D4)}$$

$$\frac{\partial}{\partial t}|\chi^{(k-1)}(t)\rangle = -\frac{i}{\hbar}H^{\dagger(k-1)}|\chi^{(k-1)}(t)\rangle + \left.\frac{\partial g_b}{\partial\langle\phi|}\right|_{(k-1)}, \quad \text{(D5)}$$

with the boundary condition of

$$|\chi^{(k-1)}(T)\rangle = -\left.\frac{\partial J_T}{\partial\langle\phi(T)|}\right|_{(k-1)}. \quad \text{(D6)}$$

The optimization process of Krotov's method proceeds as follows: It first constructs $|\chi^{(k-1)}(T)\rangle$ according to Eq. (D6), which is propagated through the backward propagation of Eq.(D5) over the entire time grid from $t = [T, 0]$. During the back-propagation stage, all states $|\chi^{(k-1)}(t)\rangle$ at each time $t = t_k$ must be stored in a memory. Then, starting from a given initial state $|\phi^{(k)}(0)\rangle$, the method updates a control pulse at the first time grid $t = t_1$ using the update equation (D3), where $\langle\chi^{(k-1)}(0)|$ is one of the back-propagated states we stored in the first step. From this updated control field, we obtain a time-evolved state $|\phi^{(k)}(t_1)\rangle$. We then update the next control field at $t = t_2$ via the update equation in Eq. (D3) using the previously obtained $|\phi^{(k)}(t_1)\rangle$. The procedure is repeated until control fields at all $N$ time grids are updated. This updated control field guarantees to decrease the cost functional $J$ [45].

In a single iteration, Krotov's method thus requires more resources compared to GRAPE because it needs to solve the Schrödinger equation $2N$ times where $N$ is the number of time grids. In addition, the method requires an exponentially large memory to store all the backward-propagated states. However, due to the monotonic convergence of Krotov's method, the method not only is more stable but can also converge faster than other quantum optimal control techniques depending on the cost functions [79].

### 2. Numerical experiments with limited control amplitudes

In Sec. VI E we presented the numerical results of generating various entangled states using the HQuGAN with Krotov's method. In this section, we perform two additional numerical experiments by constraining amplitudes of control fields to show that the HQuGAN successfully works for larger systems and can be experimental-friendly. First, we bound the control amplitudes by ±1 and try to generate generalized GHZ states using the HQuGAN with the optimal discriminator. To reduce the simulation time, we use less number of steps for the generator for a high number of qubits. As summarized in Table V, the HQuGAN using Krotov's method successfully generates (up to) the nine-qubit GHZ state. As we monotonically increase the evolution time $T$ by 10 for one-qubit increment, the number of iterations of the HQuGAN grows exponentially. To see how $T$ affects the number of iterations, we generate the nine-qubit GHZ state with three different $T = 60,\ 70,$

TABLE V. HQuGANs using Krotov's method for learning $n$-qubit GHZ states using the optimal discriminator. The number of iterations required to learn $n$-qubit GHZ states up to $n = 9$, with limited control amplitudes using the Krotov's method. Gen. iter. and Tot. iter. refer to the maximum number of generator's steps in each round and the total number of iterations taken by the HQuGAN to converge respectively. (To reduce simulation time, an optimal discriminator for all instances and a smaller number of generator steps for larger systems are used.)

| $n$ | Gen. iter. | Tot. iter. | $T$ | $N$ |
|---|---|---|---|---|
| 1 | 10 | 3 | 5 | 50 |
| 2 | 10 | 8 | 5 | 50 |
| 3 | 10 | 18 | 5 | 50 |
| 4 | 10 | 52 | 10 | 100 |
| 5 | 10 | 128 | 10 | 100 |
| 6 | 10 | 264 | 20 | 200 |
| 7 | 5 | 530 | 30 | 300 |
| 8 | 3 | 1330 | 40 | 400 |
| 9 | 3 | 1234 | 60 | 600 |
| 9 | 3 | 911 | 70 | 700 |
| 9 | 3 | 415 | 100 | 1000 |

and 100. When $T$ is increased, the number of iterations of the HQuGAN reduces significantly. This behavior numerically validates the intuition that longer $T$ introduces more parameters that assist to achieve a faster convergence rate, and provides a way to examine a more rigorous relationship between the convergence rate and the number of parameters.

We next consider an experimental setup where the coefficient for $ZZ$-interaction term in Eq. (15) is set to $J = 100$ (MHz) and the amplitudes of control fields are limited by $|\epsilon_i(t)| \leqslant 1$ (GHz). We try to learn the GHZ state for various total evolution times from $T = 20$ to 100 (ns). We use the optimal discriminator and set the generator's optimization steps to be 10 for all instances. As shown in Fig. 10, the HQuGAN successfully produces the GHZ state for all instances. As we increase evolution time from $T = 20$ to 70, the number of iterations decreases monotonically and stays around similar values after then, which again verifies that more evolution time improves the convergence rate of the algorithm.

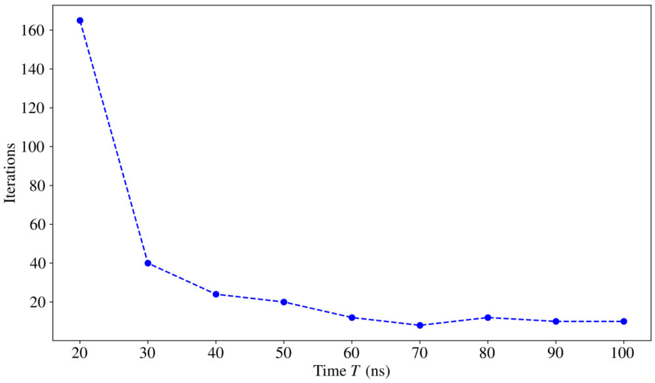

FIG. 10. HQuGANs for learning the GHZ state using Krotov's method under experimental parameters. For the LFTIM Hamiltonian described in Eq. (15), we set $J = 100$ (MHz) and aim to generate the GHZ state by limiting control amplitudes as $|\epsilon_i(t)| \leqslant 1$. The learning task was performed for various evolution time from $T = 20$ to 100 (ns). The step size is kept the same for all cases ($\lambda = 10$).

### 3. Parameters of Krotov's method

In this section, we describe the parameters of Krotov's method used in the numerical experiments. First, note that Krotov's method primarily requires backpropagating the Schrödinger equation from the boundary condition $|\chi(T)\rangle$ in Eq. (D6), which depends on the cost function $J_T$. Since the generator of the HQuGAN minimizes $J_T = |\mathrm{Tr}[D(\rho - \sigma)]|^2$, the boundary condition becomes

$$|\chi_k(T)\rangle = -\frac{\partial J_T}{\partial \langle \psi(T)|} = -2\mathrm{Tr}[D(\rho(T) - \sigma)][D|\psi(T)\rangle]. \tag{D7}$$

We can define the boundary condition similarly for the discriminator.

In addition, there are two main hyperparameters of Krotov's method that we need to set: the shape function $S(t)$ and the step width $\lambda$, as introduced earlier in Eq. (D2). The shape function contributes to the update size of the control pulses at each time grid and is encouraged to be smoothly switched on and off around $t = 0$ and $T$ to make the optimized pulses smooth, ensuring the boundary condition of pulses, if needed. The step width $\lambda$ determines the overall magnitude of $\Delta\epsilon$ as can be observed in Eq. (D3). If $\lambda$ is too large, then the size of the pulse update $\Delta\epsilon$ becomes very small, causing a slow convergence. If $\lambda$ is too small, on the other hand, then $\Delta\epsilon$ becomes very large, causing the optimization unstable [45].

For all numerical experiments that we have performed using Krotov's method, the shape function $S(t)$ is chosen as the following flat-top function:

$$S(t) = \begin{cases} \sin^2[\pi t/(2\,\mathrm{t.rise})], & \text{if } t \leqslant \mathrm{t.rise} \\ \sin^2[\pi(t-T)/(2\,\mathrm{t.fall})], & \text{if } t \geqslant T - \mathrm{t.fall} \\ 0 & \text{if } t = 0 \text{ or } t = T \\ 1 & \text{otherwise,} \end{cases} \tag{D8}$$

where $\mathrm{t.rise} = \mathrm{t.fall} = T/20$ (This shape function, which has been used in previous studies [45,79], ensures a boundary condition and switches on and off smoothly around $t = 0$ and $t = T$.) As there is no rigorous method to find an ideal value for the step width $\lambda$, we found proper values of $\lambda$ for different numerical experiments by trials and errors. For learning $n = 1, \ldots, 6$-qubit GHZ states shown in Table III, we set $\lambda = 2, 5, 10, 10, 50, 50$, respectively. For generating Table V and Fig. 10, we set $\lambda = 2$ and 10 to generate, respectively.

## APPENDIX E: BANDWIDTH LIMITATIONS

Generating bandwidth-limited control fields is an important task as precise high-bandwidth control pulses are difficult to implement in many experiments. In this

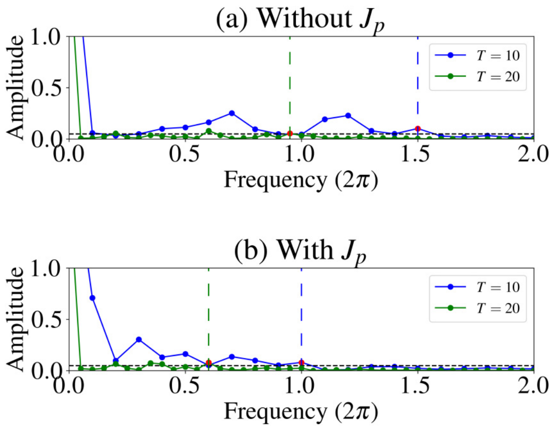

FIG. 11. (Zoomed) The fast Fourier transforms (FFTs) of the optimized control fields. The FFTs of optimized control fields that generate the GHZ state under the LFTIM Hamiltonian with a single control field (E1) for $T = 10$ and 20, where the optimization is performed (a) without the penalty term and (b) with the penalty term. In (b) the HQuGAN generates low-bandwidth control fields [compared to the free optimization case in (a)] with the assistance of the penalty term. In addition, as the evolution time gets doubled, for both cases, the maximum frequency or the bandwidth of controls decreased significantly, which is in agreement with the time-bandwidth quantum speed limit.

section, we show that the proposed HQuGAN with GRAPE can generate low-bandwidth control fields by proposing the penalty term $J_p$ described earlier in Eq. (11). Also, we verify that the algorithm obeys the time-bandwidth quantum speed limit in Eq. (12).

We consider a three-qubit LTFIM Hamiltonian with only a single control field $\epsilon(t)$ that controls all local Pauli terms in the Hamiltonian,

$$H(t) = \epsilon(t)\sum_{i=1}^{n}(X_i + Z_i) - \sum_{i=1}^{n-1} Z_i Z_{i+1}, \tag{E1}$$

where $\epsilon(0) = 1$ is set to be a constant pulse. The reason for having only one control field is to compare the bandwidths of optimized control fields in different cases more directly. The goal of the HQuGAN is to produce the GHZ state, and we consider two different evolution times $T = 10$ and 20 to examine how the bandwidth of optimal control fields depends on $T$. To estimate the bandwidth of a control field, we perform the fast Fourier transform (FFT) of the control field and then record the maximum value of frequency where its amplitude component is greater than 0.05. For accurate FFT, the number of samples (i.e., Trotter number) is set to be $100T$ in both cases.

We first optimize the HQuGAN without the penalty term $J_p$ to generate the GHZ state. The FFTs of the optimized control fields are displayed in Fig. 11(a). In the figure, the bandwidth $w_{\max}$ for each case is marked using a red dot and a dashed line: $w_{\max} = 1.5 \times 2\pi$ for $T = 10$ and $w_{\max} = 0.95 \times 2\pi$ for $T = 20$. We then conduct the same task by adding the penalty term $J_p$ to the cost function. Since such constrained optimization highly depends on the values of the hyper-parameter $\alpha$ in $J_p$, we try different values of $\alpha$ and report the case that gives the smallest value of $J_p$ in Fig. 11(b). The bandwidths are reported as $w_{\max} = 2\pi$ for $T = 10$ and $w_{\max} = 0.6 \times 2\pi$ for $T = 20$. This clearly shows that introducing $J_p$ to the HQuGAN leads the algorithm to produce a control field with lower bandwidth. We also observe that doubling $T$ lowers the bandwidth of the control field almost by half. This numerically validates the time-bandwidth limit in Eq. (12), which allows us to understand a rigorous relationship between a number of parameters of the HQuGAN that depends on total evolution time $T$ and permissible values of the bandwidth of controls.

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Generative adversarial nets, *Neural Information Processing Systems* (MIT Press, Cambridge, 2014), pp. 2672–2680

[2] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, Generative adversarial nets, in *Advances in Neural Information Processing Systems* (Curran Associates, 2016), Vol. 29, pp. 2234–2242.

[3] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, Photo-realistic single image super-resolution using a generative adversarial network, arXiv:1609.04802 (2016).

[4] C. Spampinato, S. Palazzo, P. D'Oro, D. Giordano, and M. Shah, Adversarial framework for unsupervised learning of motion dynamics in videos, Int. J. Comput. Vision **128**, 1378 (2020).

[5] E. Putin, A. Asadulaev, Y. Ivanenkov, V. Aladinskiy, B. Sanchez-Lengeling, A. Aspuru-Guzik, and A. Zhavoronkov, Reinforced adversarial neural computer for de novo molecular design, J. Chem. Inf. Model. **58**, 1194 (2018).

[6] S. Lloyd and C. Weedbrook, Quantum generative adversarial learning, Phys. Rev. Lett. **121**, 040502 (2018).

[7] C. Zoufal, A. Lucchi, and S. Woerner, Quantum generative adversarial networks for learning and loading random distributions, npj Quantum Inf. **5**, 103 (2019).

[8] B. T. Kiani, G. De Palma, M. Marvian, Z.-W. Liu, and S. Lloyd, Learning quantum data with the quantum earth mover's distance, Quantum Sci. Technol. **7**, 045002 (2022).

[9] H. Situ, Z. He, Y. Wang, L. Li, and S. Zheng, Quantum generative adversarial network for generating discrete distribution, Inf. Sci. **538**, 193 (2020).

[10] M. Y. Niu, A. Zlokapa, M. Broughton, S. Boixo, M. Mohseni, V. Smelyanskyi, and H. Neven, Entangling quantum generative adversarial networks, Phys. Rev. Lett. **128**, 220505 (2022).

[11] H.-L. Huang, Y. Du, M. Gong, Y. Zhao, Y. Wu, C. Wang, S. Li, F. Liang, J. Lin, Y. Xu *et al.*, Experimental quantum generative adversarial networks for image generation, Phys. Rev. Appl. **16**, 024051 (2021).

[12] J. Preskill, Quantum Computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[13] L. Hu, S.-H. Wu, W. Cai, Y. Ma, X. Mu, Y. Xu, H. Wang, Y. Song, D.-L. Deng, C.-L. Zou, and L. Sun, Quantum generative adversarial learning in a superconducting quantum circuit, Sci. Adv. **5**, eaav2761 (2019).

[14] K. Huang, Z.-A. Wang, C. Song, K. Xu, H. Li, Z. Wang, Q. Guo, Z. Song, Z.-B. Liu, D. Zheng *et al.*, Quantum generative adversarial networks with multiple superconducting qubits, npj Quantum Inf. **7**, 165 (2021).

[15] D. D'Alessandro, *Introduction to Quantum Control and Dynamics* (Chapman & Hall, Boca Raton, FL, 2007).

[16] S. J. Glaser, U. Boscain, T. Calarco, C. P. Koch, W. Köckenberger, R. Kosloff, I. Kuprov, B. Luy, S. Schirmer, T. Schulte-Herbrüggen *et al.*, Training Schrödinger's cat: Quantum optimal control, Eur. Phys. J. D **69**, 279 (2015).

[17] J. Werschnik and E. K. U. Gross, Quantum optimal control theory, J. Phys. B: At. Mol. Opt. Phys. **40**, R175 (2007).

[18] G. Waldherr, Y. Wang, S. Zaiser, M. Jamali, T. Schulte-Herbrüggen, H. Abe, T. Ohshima, J. Isoya, J. F. Du, P. Neumann, and J. Wrachtrup, Quantum error correction in a solid-state hybrid spin register, Nature (London) **506**, 204 (2014).

[19] F. Dolde, V. Bergholm, Y. Wang, I. Jakobi, B. Naydenov, S. Pezzagna, J. Meijer, F. Jelezko, P. Neumann, T. Schulte-Herbrüggen *et al.*, High-fidelity spin entanglement using optimal control, Nat. Commun. **5**, 3371 (2014).

[20] J. Cheng, H. Deng, and X. Qian, AccQOC: Accelerating quantum optimal control based pulse generation, arXiv:2003.00376 (2020).

[21] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, Optimal control of coupled spin dynamics: Design of NMR pulse sequences by gradient ascent algorithms, J. Magn. Reson. **172**, 296 (2005).

[22] J. P. Palao and R. Kosloff, Optimal control theory for unitary transformations, Phys. Rev. A **68**, 062308 (2003).

[23] T. Caneva, T. Calarco, and S. Montangero, Chopped random-basis quantum optimization, Phys. Rev. A **84**, 022326 (2011).

[24] Y. Shi, N. Leung, P. Gokhale, Z. Rossi, D. I. Schuster, H. Hoffmann, and F. T. Chong, Optimized compilation of aggregated instructions for realistic quantum computers, in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems* (ACM, New York, 2019), pp. 1031–1044.

[25] D. Willsch, M. Nocon, F. Jin, H. De Raedt, and K. Michielsen, Gate-error analysis in simulations of quantum computers with transmon qubits, Phys. Rev. A **96**, 062302 (2017).

[26] O. R. Meitei, B. T. Gard, G. S. Barron, D. P. Pappas, S. E. Economou, E. Barnes, and N. J. Mayhall, Gate-free state preparation for fast variational quantum eigensolver simulations, npj Quantum Inf. **7**, 155 (2021).

[27] A. Asthana, C. Liu, O. R. Meitei, S. E. Economou, E. Barnes, and N. J. Mayhall, Leakage reduces device coherence demands for pulse-level molecular simulations, Phys. Rev. Appl. **19**, 064071 (2023).

[28] R. de Keijzer, O. Tse, and S. Kokkelmans, Pulse based Variational Quantum Optimal Control for hybrid quantum computing, Quantum **7**, 908 (2023).

[29] Y. Balaji, M. A. Sajedi, N. M. Kalibhat, M. Ding, D. Stöger, M. Soltanolkotabi, and S. Feizi, Understanding overparameterization in generative adversarial networks, arXiv:2104.05605 (2021).

[30] Z. Allen-Zhu, Y. Li, and Z. Song, A convergence theory for deep learning via over-parameterization, in *Proceedings of the 36th International Conference on Machine Learning* edited by K. Chaudhuri and R. Salakhutdinov (PMLR, 2019), Vol. 97, pp. 242–252.

[31] M. Larocca, N. Ju, D. García-Martín, P. J. Coles, and M. Cerezo, Theory of overparametrization in quantum neural networks, Nat. Comput. Sci. **3**, 542 (2023).

[32] E. R. Anschuetz and B. T. Kiani, Quantum variational algorithms are swamped with traps, Nat. Commun. **13**, 7760 (2022).

[33] X. You, S. Chakrabarti, and X. Wu, A convergence theory for over-parameterized variational quantum eigensolvers, arXiv:2205.12481 (2022).

[34] M. Larocca, P. Czarnik, K. Sharma, G. Muraleedharan, P. J. Coles, and M. Cerezo, Diagnosing barren plateaus with tools from quantum optimal control, Quantum **6**, 824 (2022).

[35] M. Arjovsky and L. Bottou, Towards principled methods for training generative adversarial networks, arXiv:1701.04862 (2017).

[36] A. Srivastava, L. Valkov, C. Russell, M. U. Gutmann, and C. Sutton, VEEGAN: Reducing mode collapse in GANs using implicit variational learning, arXiv:1705.07761 (2017).

[37] M. Arjovsky, S. Chintala, and L. Bottou, Wasserstein gan, arXiv:1701.07875 (2017).

[38] A. Odena, C. Olah, and J. Shlens, Conditional image synthesis with auxiliary classifier GANs, arXiv:1610.09585 (2017).

[39] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, Cost function dependent barren plateaus in shallow parametrized quantum circuits, Nat. Commun. **12**, 1791 (2021).

[40] G. D. Palma, M. Marvian, D. Trevisan, and S. Lloyd, The quantum Wasserstein distance of order 1, IEEE Trans. Inf. Theory **67**, 6627 (2021).

[41] N. Leung, M. Abdelhafez, J. Koch, and D. Schuster, Speedup for quantum optimal control from automatic differentiation based on graphics processing units, Phys. Rev. A **95**, 042318 (2017).

[42] C. Arenz and H. Rabitz, Drawing together control landscape and tomography principles, Phys. Rev. A **102**, 042207 (2020).

[43] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. (Springer, New York, 2006).

[44] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, A limited memory algorithm for bound constrained optimization, SIAM J. Sci. Comput. **16**, 1190 (1995).

[45] M. H. Goerz, D. Basilewitsch, F. Gago-Encinas, M. G. Krauss, K. P. Horn, D. M. Reich, and C. P. Koch, Krotov: A Python implementation of Krotov's method for quantum optimal control, SciPost Phys. **7**, 080 (2019).

[46] V. M. Frey, S. Mavadia, L. M. Norris, W. de Ferranti, D. Lucarelli, L. Viola, and M. J. Biercuk, Application of optimal band-limited control protocols to quantum noise sensing, Nat. Commun. **8**, 2189 (2017).

[47] C.-C. Shu, T.-S. Ho, X. Xing, and H. Rabitz, Frequency domain quantum optimal control under multiple constraints, Phys. Rev. A **93**, 033417 (2016).

[48] M. Hirose and P. Cappellaro, Time-optimal control with finite bandwidth, Quantum. Inf. Process **17**, 88 (2018).

[49] R. W. Heeres, P. Reinhold, N. Ofek, L. Frunzio, L. Jiang, M. H. Devoret, and R. J. Schoelkopf, Implementing a universal gate set on a logical qubit encoded in an oscillator, Nat. Commun. **8**, 94 (2017).

[50] S. Lloyd and S. Montangero, Information theoretical analysis of quantum optimal control, Phys. Rev. Lett. **113**, 010502 (2014).

[51] N. Rach, M. M. Müller, T. Calarco, and S. Montangero, Dressing the chopped-random-basis optimization: A bandwidth-limited access to the trap-free landscape, Phys. Rev. A **92**, 062343 (2015).

[52] D. Lucarelli, Quantum optimal control via gradient ascent in function space and the time-bandwidth quantum speed limit, Phys. Rev. A **97**, 062346 (2018).

[53] J. Zhang, G. Pagano, P. W. Hess, A. Kyprianidis, P. Becker, H. Kaplan, A. V. Gorshkov, Z. X. Gong, and C. Monroe, Observation of a many-body dynamical phase transition with a 53-qubit quantum simulator, Nature (London) **551**, 601 (2017).

[54] T. F. Rønnow, Z. Wang, J. Job, S. Boixo, S. V. Isakov, D. Wecker, J. M. Martinis, D. A. Lidar, and M. Troyer, Defining and detecting quantum speedup, Science **345**, 420 (2014).

[55] P. L. McMahon, A. Marandi, Y. Haribara, R. Hamerly, C. Langrock, S. Tamate, T. Inagaki, H. Takesue, S. Utsunomiya, K. Aihara *et al.*, A fully programmable 100-spin coherent ising machine with all-to-all connections, Science **354**, 614 (2016).

[56] A. A. Ovchinnikov, D. V. Dmitriev, V. Y. Krivnov, and V. O. Cheranovskii, Antiferromagnetic ising chain in a mixed transverse and longitudinal magnetic field, Phys. Rev. B **68**, 214406 (2003).

[57] J. Johansson, P. Nation, and F. Nori, Qutip: An open-source python framework for the dynamics of open quantum systems, Comput. Phys. Commun. **183**, 1760 (2012).

[58] J. Haah, A. W. Harrow, Z. Ji, X. Wu, and N. Yu, Sample-optimal tomography of quantum states, IEEE Trans. Inf. Theory, **63**, 5628 (2017).

[59] D. M. Reich, M. Ndong, and C. P. Koch, Monotonically convergent optimization in quantum control using Krotov's method, J. Chem. Phys. **136**, 104103 (2012).

[60] A. Bisio, G. Chiribella, G. M. D'Ariano, S. Facchini, and P. Perinotti, Optimal quantum learning of a unitary transformation, Phys. Rev. A **81**, 032324 (2010).

[61] I. Marvian and S. Lloyd, Universal quantum emulator, arXiv:1606.02734 (2024).

[62] S. Khatri, R. LaRose, A. Poremba, L. Cincio, A. T. Sornborger, and P. J. Coles, Quantum-assisted quantum compiling, Quantum **3**, 140 (2019).

[63] M. R. Geller, Z. Holmes, P. J. Coles, and A. Sornborger, Experimental quantum learning of a spectral decomposition, Phys. Rev. Res. **3**, 033200 (2021).

[64] K. Heya, Y. Suzuki, Y. Nakamura, and K. Fujii, Variational quantum gate optimization, arXiv:1810.12745 (2018).

[65] H.-Y. Huang, S. Chen, and J. Preskill, Learning to predict arbitrary quantum processes, arXiv:2210.14894 (2022).

[66] A. Gilchrist, N. K. Langford, and M. A. Nielsen, Distance measures to compare real and ideal quantum processes, Phys. Rev. A **71**, 062310 (2005).

[67] J. Li, X. Yang, X. Peng, and C.-P. Sun, Hybrid quantum-classical approach to quantum optimal control, Phys. Rev. Lett. **118**, 150503 (2017).

[68] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, Phys. Rev. A **98**, 032309 (2018).

[69] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Evaluating analytic gradients on quantum hardware, Phys. Rev. A **99**, 032331 (2019).

[70] C. Daskalakis, S. Skoulakis, and M. Zampetakis, The complexity of constrained min-max optimization, arXiv:2009.09623 (2020).

[71] S. T. Flammia, D. Gross, Y.-K. Liu, and J. Eisert, Quantum tomography via compressed sensing: error bounds, sample complexity and efficient estimators, New J. Phys. **14**, 095022 (2012).

[72] S. Chen, B. Huang, J. Li, and A. Liu, Tight bounds for state tomography with incoherent measurements, arXiv:2204.07155 (2022).

[73] A. M. Childs and Y. Su, Nearly optimal lattice simulation by product formulas, Phys. Rev. Lett. **123**, 050503 (2019).

[74] R. Chakrabarti and H. Rabitz, Quantum control landscapes, Int. Rev. Phys. Chem. **26**, 671 (2007).

[75] C. Brif, R. Chakrabarti, and H. Rabitz, Control of quantum phenomena: past, present and future, New J. Phys. **12**, 075008 (2010).

[76] C. Ferrie, Self-guided quantum tomography, Phys. Rev. Lett. **113**, 190404 (2014).

[77] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms, Nat. Rev. Phys. **3**, 625 (2021).

[78] S. E. Sklarz and D. J. Tannor, Loading a Bose-Einstein condensate onto an optical lattice: An application of optimal control theory to the nonlinear Schrödinger equation, Phys. Rev. A **66**, 053619 (2002).

[79] G. Jäger, D. M. Reich, M. H. Goerz, C. P. Koch, and U. Hohenester, Optimal quantum control of Bose-Einstein condensates in magnetic microtraps: Comparison of gradient-ascent-pulse-engineering and Krotov optimization schemes, Phys. Rev. A **90**, 033628 (2014).