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, NM 87131, USA*

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

Inspired by the success of generative adversarial networks (GANs) in learning high-dimensional distributions, recently quantum GAN (QuGAN) has been proposed as quantum algorithm to learn quantum states. Whether QuGAN is capable of performing meaningful tasks in realistic physical systems, however, is less understood. In this work, we propose a new variant of QuGAN, Hamiltonian QuGAN (HQuGAN), that generates desired quantum resources by controlling time-continuous quantum dynamics using quantum optimal control. Such continuous pulse model not only can fit naturally to near-term hardwares, but also has the potential to provide a better convergence of the algorithm due to overparameterization comparing to the gate-based model. Numerically we show that HQuGAN successfully learns various quantum states up to nine qubits. Furthermore, we describe how to implement HQuGAN on quantum computers and analyze its computational cost. We also show how QuGAN algorithm can be modified to learn unitary evaluations.

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 the data generated from the generator and the real data. Such adversarial game can be deemed as a minimax game that converges to some Nash equilibrium 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], and so on.

Inspired by the success of classical GANs, a quantum mechanical counterpart of GANs, a quantum GAN (QuGAN) [6, 7], has been proposed recently. Unlike classical GANs, both input and output datasets in QuGAN are quantum mechanical such as an ensemble of quantum states. 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 cost functions that generator and the discriminator both perform convex-optimizations within compact and convex sets, there always exists Nash equilibrium point in the generator-discriminator strategy space [6]. In fact, such equilibrium point is unique and achieved when the discriminator is unable to tell the difference between the true ensemble and the generated ensemble [6]. Similar to the classical GANs, QuGAN has been used in learning random distributions [7], discrete distributions [8], quantum states [9], and generating images [10]. Such applications make use of Variational Quantum Algorithms (VQAs) to train QuGAN: the generator and the discriminator are parameterized quantum circuits, where the parameters are optimized via classical optimizers. This makes QuGAN feasible to be implemented on near-term Noisy Intermediate-Scale Quantum computers (NISQ) [11]. In fact, QuGAN has already been explored in quantum devices such as superconducting quantum processors to learn quantum states of small system sizes [12, 13].

While the aforementioned applications on QuGAN are successful in learning classical distributions and stationary states, whether QuGAN can perform meaningful tasks in realistic physical systems has not been studied yet. This motivates our *Hamiltonian* QuGAN (HQuGAN) that deals with continuous time evolution of quantum systems: given continuous time-dependent Hamiltonian, the proposed HQuGAN aims to generate desired quantum resources, such as quantum states or unitary transformations, by controlling the Hamiltonian.

How to control such continuous time-dependent Hamiltonian is a well-studied field — *quantum optimal control* (QOC) [14]. The goal of QOC is to find optimal sets of control parameters, or pulses, to achieve a predefined goal by steering the dynamics of given quantum systems. Examples of such objectives include optimizing the fidelity between two quantum states, average gate fidelity, and expectation values of an observable [15, 16]. In fact, some applications of QOC give promising results in field of quantum computation such as designing high-fidelity quantum entangling gates [17, 18]. In addition, QOC can effectively reduce the latency of groups of quantum gates, which current gate-based compilations for quantum systems suffer from [19]. There have been extensive studies on applications and modifications of popular methods of QOC including gradient-based methods such as GRAPE [20] and Krotov [21] and gradient-free methods such as CRAB [22], to many different quantum systems.

The proposed HQuGAN has several favorable properties compared to the existing models of QuGAN. First, unlike earlier implementations of QuGANs using parameterized gates, the HQuGAN performs the learning task by changing the native parameters of the Hamiltonian itself. This is indeed an appealing aspect for mainly two reasons. First, in the gate-based approaches to quantum computations, each quantum gate must be translated into control pulses, which are the electrical signals that implement the specified operations on the underlying quantum device. However, such underlying hardware may implement a different set of operations that are inconsistent with the signals produced from the gate set. This creates a barrier between the expressive logical gates and the set of operational instructions that can be efficiently implemented on the real experimental system [23]. Besides, even if the translation process is successful, the algorithm suffers from possibly large gate errors accumulated by each translated gates in current quantum devices, causing most of the mismatch between the ideal gates and the implemented pulses [24]. Controlling the Hamiltonian itself, however, would avoid both barriers.

Moreover, the HQuGAN can benefit from *overparameterization* due to a continuous nature of control parameters, leading to a better convergence on the minimax game. For classical GANs, recently it has been shown overparameterization appears to be a key factor in successful training of GANs to global saddle points [25]. 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 [26, 27]. Since QNNs (gate level) and QOC (pulse level) can be unified in a single framework, the advantage of the overparameterization phenomenon also applies to QOC models [27, 28]. Subsequently, we expect both overparamaterized generator and discriminator would explore their loss landscapes sufficiently and thus converge to a global Nash equilibrium.

With this unification of QuGAN and QOC in mind, we now describe the remainder of the paper. We begin by introducing the concepts of GANs and QuGAN in Section II, followed by introductions to QOC in Section III with specific focus on GRAPE method. Then we present our numerical simulations on generating different quantum states using the proposed HQuGAN in Section VI. Due to extremely intractable computational complexities of QOC for large number of qubits, however, classical computations at some point cannot handle the optimizations on the HQuGAN. Hence, we propose a way to use quantum computers as subroutines of the HQuGAN to avoid such intractability in Section VII. We analyze complexities of several computational resources of the HQuGAN and compare them to the ones from relevant current methods.

II. QUANTUM GAN

In classical GANs, to learn the generator's distribution p_g over data x, we define a prior on input noise variables $p_z(z)$, then represent a mapping to data space as $G(\theta_g, z)$, where G is a parameterized generative neural network with parameters θ_g. We also define another parameterized discriminative neural network $D(\theta_d, x)$ that represents the probability that x came from the data rather than p_g. The goal of the generator is to fool the discriminator by generating $G(\theta_g, z)$ that it resembles $D(x)$ as much as possible. The discriminator then tries to restrain herself from getting fooled by the generator. Hence, GANs alternate between the discriminator maximizing 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 θ_d is maximizing. Formally, the two players play the following minimax game via solving:

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

In each player's turn, each set of parameters is optimized by taking its gradient. After several steps of training, if G and D have enough capacity, they will reach a Nash equilibrium point at which both cannot improve when $p_g = p_{data}$ [1].

In a quantum GAN (QuGAN), both generator and discriminator output quantum states so that the true data is described by an ensemble of quantum states σ. After generator updates its parameters θ_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 the discriminating measurement. In other words, the discriminator attempts to find a Hermitian operator D that separates the expectation value of D between the two quantum states as much as possible, i.e. maximizing $\mathrm{Tr}(D(\rho - \sigma))$. As a consequence, the game of QuGAN can be expressed as solving [6]

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

The core idea of QuGAN, 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 σ without using classical descriptions of σ but using D. While originally $\|D\|_1 \leq 1$ had been proposed to be a constraint on D in [6], QuGAN based on quantum Wasserstein distance of order 1 (or quantum W_1 distance) has been proposed recently [29], which puts quantum Lipschitz constraint to the operator $\|D\|_L \leq 1$. The quantum W_1 distance is based on notions of neighboring quantum states. Two states are neighboring if they differ only by one qubit. The quantum W_1 distance is then induced by the maximum norm that assigns distance at most one to every couple of neighboring states. Using its dual formulation [29], minimizing the quantum W_1 distance can be expressed as the following minimax game,

$$\min_{\theta_g}\max_{D}\{\mathrm{Tr}(D(\rho(\theta_g) - \sigma)), \|H\|_L \leq 1\}, \tag{3}$$

where the quantum Lipschitz constant of H is defined as

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

In fact, [30] shows that loss landscapes induced by QuGAN using the quantum W_1 distance can potentially give advantages in learning certain structured states like the GHZ states more efficiently compared to other metrics such as fidelity.

In addition, we will also use the following cost function,

$$\min_{\theta_g} \max_{D} |\mathrm{Tr}(D(\rho(\theta_g) - \sigma))|^2. \tag{5}$$

We also note such new cost function still guarantees the existence of Nash equilibrium in the same location as before on $\rho = \sigma$, in Section IV.

III. QUANTUM OPTIMAL CONTROL

The goal of quantum optimal control (QOC) is to find control parameters, or control pulses $\{\epsilon_i(t)\}$, to achieve a predefined goal under given quantum dynamics. The set of pulses used in QOC is subjected to time-dependent Hamiltonians,

$$H(t) = H_0 + \sum_i \epsilon_i(t) H_i, \tag{6}$$

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

$$\min_{\{\epsilon_i\}} J[\{\epsilon_i(t)\}] \tag{7}$$

subject to Schrödinger equation of the Hamiltonian. $J[\{\epsilon_i(t)\}]$ typically consists of main objective functional and functional describing a running cost of control fields, and other physical constraints depending on characterizations of hardware devices or laboratory resources, if necessary [14]. The main functional refers to a final time functional such as infidelity between fully time-evolved quantum states and a target state, e.g.

$$F = 1 - |\langle\psi_{targ}|\psi(T)\rangle|^2 = 1 - |\langle\psi_{targ}|\, U(T)\, |\psi_0\rangle|^2, \tag{8}$$

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.

The most popular QOC technique is Gradient Ascent Pulse Engineering (GRAPE) [20]. Since $U(T)$ is hard to be obtained analytically, the algorithm first discretizes the time domain into N equal pieces of intervals ($\Delta t = T/N$) and approximates Hamiltonian to be piecewise constant within each interval $[t_j, t_j + \Delta t)$. The total time-evolution operator then can be expressed as multiplications of N unitary operations,

$$U(T) = \prod_{j=1}^{N} U(t_j) = \prod_{j=1}^{N} \exp\Big[- i\Delta t(H_0 + \sum_i \epsilon_i(t_j)H_i)\Big]. \tag{9}$$

Then, control fields for all time grids are updated concurrently in directions of their gradients with respect to the objective function J,

$$\epsilon_i(t_j) \longleftarrow \epsilon_i(t) + \alpha \frac{\partial J}{\partial \epsilon_i(t_j)}, \tag{10}$$

where the control gradient is typically obtained via approximating propagator derivatives to the first order of Δt [20],

$$\frac{\partial U(t_j)}{\partial \epsilon_i(t_j)} \approx -i\Delta t U(t_j) H_i. \tag{11}$$

which makes the computation very affordable. To achieve faster and stable convergence of the optimization process, GRAPE utilizes a quasi-Newton method, particularly Broyden–Fletcher–Goldfarb–Shanno algorithm (BFGS) [31] or Limited-memory BFGS (L-BFGS) [32], which makes use of an approximated Hessian matrix of the cost function.

The other optimal control protocol we consider here is the Krotov's method [21]. 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. Another appealing property of Krotov's method is that it mathematically guarantees that control fields are continuous in time [33]. However, it is computationally more expensive than GRAPE since a single optimization step requires to solve the Schrödinger equations $2N$ (N is the number of time grids) times, which would vastly be intractable as system size grows. We demonstrate more details of Krotov's method and its applications our algorithm in Appendix C.

IV. HAMILTONIAN GAN

We now merge the two pictures of QuGAN and QOC together to introduce how the proposed HQuGAN learns an arbitrary unknown quantum state σ based on the minimax game described in Section II, Eq.(5). Given descriptions of physical systems $H(t)$ and initializations on relevant parameters, the generator finds his optimal sequences of control fields θ_g to generate a time-evolved quantum state $\rho(\theta_g)$ such that $|\mathrm{Tr}(D(\rho(\theta_g)-\sigma))|^2$ is minimized using one of QOC techniques while fixing D. After the generator's turn, the discriminator now tries to find an observable D such that $|\mathrm{Tr}[D(\rho - \sigma)]|^2$ is maximized while fixing ρ. Similar to the generator's protocol, the discriminator finds her optimal sequences of control fields

Algorithm 1 Hamiltonian Generative Adversarial Networks for Learning Arbitrary Quantum State

Input: time-dependent Hamiltonian $H(t)$, unknown target state σ, initial control fields $\{\epsilon_i(0)\}$, initial state ρ_0, evolution time T, trotter steps N, fixed measurement operator D_0
Output: quantum state $\rho(T)$

```
procedure HAMILTONIAN GENERATIVE ADVERSARIAL NETWORKS
    while $F(\rho(T),\sigma) \leq 0.999$ do                         ▷ Terminates when $F(\rho(T),\sigma) > 0.999$
        procedure GENERATOR
            if first round of QuGAN then
                Initialize initial control pulse $\{\epsilon_i(0)\}_g = \{\epsilon_i(0)\}$ and fixed measurement operator $D = D_0$
            Minimize $|\mathrm{Tr}(D(\rho(T)-\sigma))|^2$ using QOC
            $\{\epsilon_i(0)\}_g \leftarrow \{\epsilon_i(t)\}_g$     ▷ Updates optimize control fields as initial control fields for next round of generator
            return $\rho(T)$                                         ▷ Returns time-evolved quantum state with optimized control fields
        procedure DISCRIMINATOR
            Maximize $|\mathrm{Tr}(D_0U(\rho-\sigma)U^\dagger)|^2$ using QOC
            $D \leftarrow \mathrm{U}^\dagger D_0 U$                  ▷ Updates new observable $D$
            return $D$
    return $\rho(T)$
```

to generates a unitary U such that $|\mathrm{Tr}[D_0U(\rho-\sigma)U^\dagger]|^2$ is maximized for a fixed measurement operator D_0. On quantum computers, this is precisely the discriminator creates a unitary transformation before some fixed measurement, to differentiate the measurement outcomes as much as possible. After discriminator successfully finds her optimal sequences, the generator begins his optimization from the optimal θ_g he found in the last round, but now with new observable $D = U^\dagger D_0 U$. Such iterations between the two players continue until the Uhlmann fidelity squared between the generator's state and the target state exceeds 0.999. The algorithm is described in Algorithm 1. Consequently, the proposed Hamiltonian QuGAN solves the following minimax game:

$$\min_{\theta_g}\max_{\theta_d}|\mathrm{Tr}(D(\theta_d)(\rho(\theta_g)-\sigma))|^2, \tag{12}$$

where $D(\theta_d) = U^\dagger(\theta_d)D_0U(\theta_d)$ for fixed D_0, and θ_g and θ_d are control parameters of the generator and the discriminator, respectively. In fact, the minimax game described above has a Nash equilibrium point in the desired location. The Nash equilibrium is a stationary point where no player can benefit by changing strategy while the other player keeps hers unchanged. In other words, the Nash equilibrium is a point (θ_g^*, θ_d^*) where θ_g^* gives a global minimum of $f(\cdot,\theta_d^*)$ and θ_d^* gives a global maximum of $f(\theta_g^*,\cdot)$. Henceforth, the minimax game has the Nash equilibrium in the desired location of $\rho(\theta_g) = \sigma$.

V. OVERPARAMETERIZATION AND BANDWIDTH LIMITATIONS

To be added.

VI. NUMERICAL RESULTS

In this section, we present our numerical results of the HQuGAN using GRAPE to learn different quantum states under given closed quantum systems. (The results for the Krotov's method can be found in Appendix C.)

A. Setups

We consider an n-qubits closed quantum system under 1D continuous time-dependent Transverse Field Ising Model (TFIM) Hamiltonian with open boundary conditions,

$$H(t) = \sum_{i=1}^{n}\epsilon_i(t)X_i + \sum_{i=1}^{n}\epsilon_{i+n}(t)Z_i - J\sum_{i=1}^{n-1}Z_iZ_{i+1}, \tag{13}$$

for evolution time $t \in [0, T]$, for both the generator and the discriminator, and set $J = 1$. (We set $\hbar = 1$, and therefore the coupling are expressed in hertz. e.g. If T is in nanoseconds then $\epsilon_i(t)$ is in gigahertz.) For n-qubits systems, hence, each player optimizes over $2n$ control pulses of local Pauli Hamiltonians.

The initial control fields are chosen as sinusoidal shapes for both players,

$$\epsilon_i(t=0) = \begin{cases}\sin(10t/T), & \text{for } i = 1, ..., n,\\ \cos(10t/T), & \text{for } i = n+1, ..., 2n.\end{cases} \tag{14}$$

We set an initial state to be the groundstate of $H(t)$ at $t = 0$, which is $|1\rangle^{\otimes n}$, so that it is easy to prepare. We keep control pulses at $t = 0$ unchanged for generator by setting the gradients at $t = 0$ to 0 so that the generator always begins with state $|1\rangle^{\otimes n}$. In addition, we set the fixed measurement D_0 to be a 1-local computational basis measurement, i.e. $D_0 = Z \otimes I^{\otimes n-1}$.

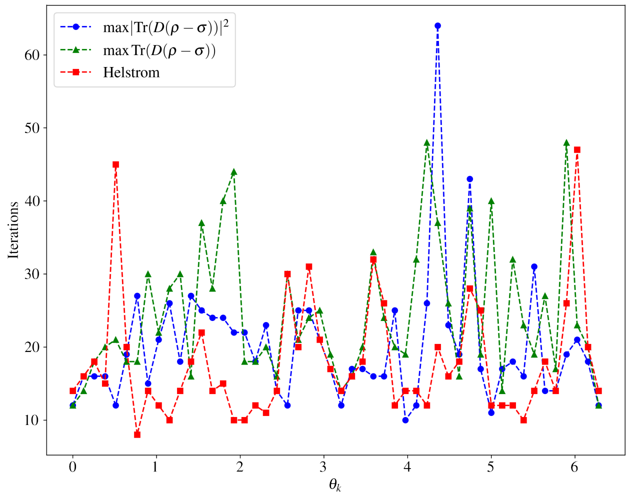

FIG. 1: **HQuGAN for learning 3-qubits states.** Here we present our numerical results on generating 50 different 3 qubit pure states described in Eq.(15) for three distinct discriminators: a discriminator with the standard maximization of Eq.(12) (blue), with the cost function without absolute value (green), and being optimal by choosing the Helstrom measurement operator (red). While the optimal discriminator performs overall the best, the other discriminators using GRAPE give comparable number of iterations, ensuring that the structures of the HQuGAN discriminators are indeed appropriate. Our numerical results also emphasize that the discriminator maximizing $\mathrm{Tr}(D(\rho-\sigma))$ (green) still makes the algorithm to converge to the proper Nash equilibrium without using a lot more iterations compare to the original cost function. Since such maximization significantly reduces amounts of resources to calculate the gradients during the optimization process, we expect to exploit such discriminative model for larger systems.

Regarding optimization conditions, we use the minimax cost function in Eq.(12), where the gradients are approximated to the first order of $\Delta t = T/N$ for both generator and discriminator. As mentioned earlier, the HQuGAN terminates once the generator generates a quantum state that gives > 0.999 fidelity with the target state.

B. Learning Superposition States

We first attempt to learn 50 different 3-qubits superposition states,

$$|\psi_{targ}\rangle = \cos\theta_k |000\rangle + \sin\theta_k |111\rangle \,, \tag{15}$$

where $\theta_k = 2\pi k/50$ for $k = 0, 1, \ldots, 49$. The learning is performed classically using the HQuGAN described in Algorithm. 1 equipped with GRAPE. We take the total evolution time as $T = 5$ with trotter steps $N = 50$, for both the generator and the discriminator. Both players perform full training: the generator terminates when the cost function is below than 10^{-5} and the discriminator terminates the norm of the gradients is smaller than 10^{-5} or the maximum iteration steps of 30 is reached.

The HQuGAN successfully generated all 50 states as shown in Fig.1, where the y axis of the plot represents the total number of iterations of the HQuGAN until the generator generates the quantum state that reaches > 0.999 fidelity with each of the target states. To illustrate how well discriminators perform, we conduct the same task with optimal discriminator by always choosing the Helstrom measurement operator (red lines). The Helstrom measurement operator is chosen to be a sum of two projection operators onto positive and negative eigenspaces of $\rho - \sigma$, as explained in Eq.(A2) with $p = 1$. In fact, the discriminator using GRAPE cannot produce the Helstrom measurement because $D = U^\dagger D_0 U$ will always have the same eigenspectrum with D_0, which is ± 1. Nevertheless, for almost all instances, the number of iterations do not differ by more than 10, indicating that the discriminative models of our HQuGAN are indeed expressive and properly structured.

We also consider a case where the discriminator maximizes $\mathrm{Tr}(D(\theta_d)(\rho-\sigma))$ (green line) instead of original discriminator's cost function (blue line, Eq. (12)). Noting that maximizing $\mathrm{Tr}(D(\theta_d)(\rho-\sigma))$ always leads to maximizing $|\mathrm{Tr}(D(\theta_d)(\rho-\sigma))|^2$, we emphasize that such maximization is more efficient than the original maximization because 1) it substantially reduces amounts of resources to calculate the gradients, 2) it numerically gives proper convergence to the desired Nash equilibrium, and 3) the mode collapse problem can be cured by applying the absolute value symbol only to the generator's cost function and hence the absolute value symbol for the discriminator's cost function is unnecessary (See Appendix B). Subsequently, we conclude that the algorithm reaches the equilibrium more efficiently via discriminator maximizing $\mathrm{Tr}(D(\theta_d)(\rho-\sigma))$, as we exploit it in the next section for ease of computations.

C. Learning Generalized GHZ States

Since learning different superposition states is successful, we now shift gears to a more challenging task of generating the generalized Greenberger-Horne-Zeilinger (GHZ) states, which are extremely useful in many fields such as quantum information and quantum metrology due to its maximally entangled property. We hence focus on generating n-qubits GHZ states,

$$|\psi_{targ}\rangle = \frac{1}{\sqrt{2}}(|0\rangle^{\otimes n} + |1\rangle^{\otimes n}), \tag{16}$$

where every settings on the HQuGAN remain unchanged from previous section, and the discriminator is set to maximize $\mathrm{Tr}(D(\theta_d)(\rho-\sigma))$. We take different T with respect to different system sizes and number of time grids

n	Iter (GRAPE)	Iter (Helstrom)	T	N
1	3	3	5	50
2	6	6	5	50
3	11	10	5	50
4	26	35	10	100
5	43	48	20	200
6	111	128	30	300

TABLE I: **HQuGAN for learning generalized GHZ states.** We summarize our results on successfully learning the generalized n-qubits GHZ states using the HQuGAN with GRAPE, where the discriminator either also uses GRAPE or always becomes an optimal discriminator by giving Helstrom operator. We set number of time grids to be $N = 10T$ for all instances. Consequently, we observe that numbers of iterations of the HQuGAN for both cases are very similar.

to be $N = 10T$ for all instances. As before, we use both GRAPE equipped and optimal discriminators to see how well our HQuGAN performs.

Our numerical results are summarized in Table I. The HQuGAN is able to produce n-qubits GHZ states for $n = 1, \ldots, 6$ for both of the discriminators with similar numbers of iterations. This indicates that the discriminator using GRAPE properly maximizes the expectation values between the generated state and the true state and thus discriminates the two quantum states accordingly. Furthermore, as shown in Fig.2, both cases give a similar shape of the optimized control pulses. It is important to note that TFIM Hamiltonian with open boundary conditions is a *controllable* system [28], meaning its Dynamical Lie Algebra is full rank. This implies that the control system can generate every unitary in $\mathcal{SU}(2^n)$ up to a phase by appropriately choosing control parameters, which gets benefits from overparameterization of the discriminator. Subsequently, we conclude the controllability of the system, mingled with overaparameterization, make our discriminators sufficiently expressive and lead to the desired convergence of the algorithm.

D. Remarks

In this section, we describe additional numerical simulations we have performed.

1. *Krotov's Method*

As said earlier, Krotov's method is another popular line of gradient-based QOC technique. In contrast to GRAPE, Krotov's method mathematically guarantees that the control pulse sequences are already time-continuous throughout its optimization process [34]. We thus conduct the same tasks of generating different quantum states using Krotov's method and present numerical results in Appendix C 2. In addition, we perform the HQuGAN with limited control amplitudes. We bounded the amplitude of the control fields to ±1 and optimized the HQuGAN with optimal discriminator to generate (up to) 9-qubits GHZ states. In addition, we perform the HQuGAN to generate the GHZ states under the same TFIM Hamiltonian model with experimental parameters for its coefficients. These results are summarized in Appendix C 3.

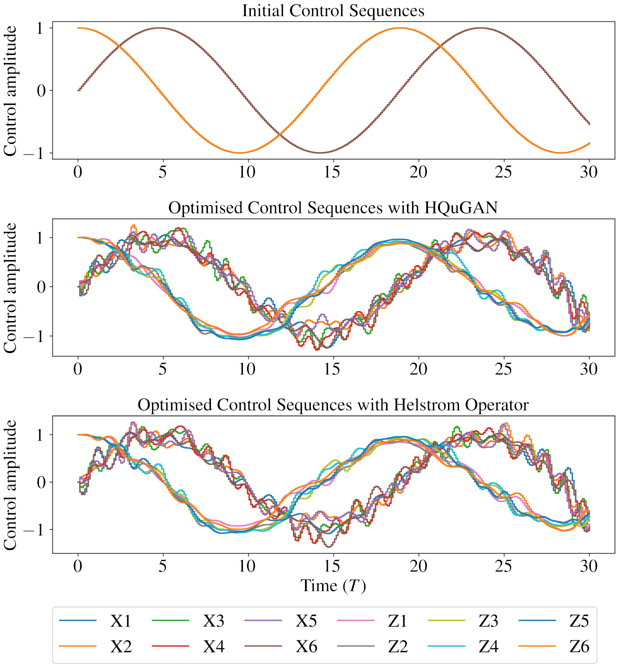

FIG. 2: **Optimized control fields for 6-qubits GHZ state.** We illustrate final control pulse sequences optimized by the HQuGAN using GRAPE with two different discriminators that generate 6-qubits GHZ state, with the same initial control sequences in Eq.(14). Remarkably, we observe that the two optimized sequences resemble each other, likewise their similar number of iterations as shown in Table I.

2. *Quantum Wasserstein Distance of Order* 1

As briefly mentioned earlier, the dual formalism of quantum W_1 distance makes it possible to express in terms of minimax game described in Eq.(3). It has been studied that such quantum Wasserstein GAN (qWGAN) exhibits more favorable loss landscapes compared to other conventional metrics such as fidelity in learning specific structures of quantum states [30]. An intuition behind such advantage lies in the fact that while common metrics such as fidelity capture global property of cost functions, which induces *barren plateaus* [35], the quantum W_1 distance induces optimization over local operations. Accordingly, we provide numerical results of our HQuGAN using the quantum W_1 distance to learn up to 5-qubits GHZ states in Appendix D. For both GRAPE and Krotov's method, the HQuGAN with the quantum W_1 distance converged faster for generating 5-qubits GHZ states. (More in-depth analysis on such results are to be added)

3. Mode Collapse

The fundamental reason of the mode collapse in QuGAN is based on the form of cost function described in Eq.(2), where the generator tries to minimizes $\mathrm{Tr}(D\rho)$ and thus aligns ρ with D, i.e. $\rho \sim D$. If D in future rounds of discriminator ever happens to be one of previous generator's states, then the generator would output that state, falling into a loop that the generator repeatedly learns only a small set of states in periods. This is in agreement with results in classical machine learning, where the generator of classical GANs tends to characterize only a few modes of the true distribution, but misses many other important modes [36]. We display a specific example of the mode collapse in 1-qubit system, which was suggested in [9]. In fact, we observe that the mode collapse occurs in almost every instances of the HQuGAN simulations. To prevent this, we numerically show that the modified cost function we described in Eq.5 is able to remedy the issue for all instances. These results are summarized in Appendix B. (More in-depth analysis on such results are to be added)

4. Learning Unitary Transformation

We also extend our methods on learning an arbitrary unknown quantum state to learn an arbitrary unknown unitary transformation in Appendix F. We consider two different setups. First, we aim to learn an unknown unitary transformation given pairs of input states of the unitary and their corresponding output states. This can be achieved by performing the same protocol of the HQuGAN equipped with multiple distinct discriminators. Second, we try to learn a given Choi matrix for the unknown unitary transformation. By considering the Choi matrix as a unknown quantum state, we can use the HQuGAN directly to generate the Choi matrix and thus the generator learns the unitary transformation. As an example, we show that the HQuGAN is able to learn the Hadamard gate properly for both setups. (More in-depth analysis on such results are to be added)

VII. IMPLEMENTATION ON A QUANTUM COMPUTER

So far, we have described how to use quantum optimal control techniques to estimate the cost functions and their gradients in classical computers. Although it is successful in low-dimensional systems, the amount of computations required becomes vastly intractable as the system size grows due to *the curse of dimensionality*. Therefore, in this section, we propose a way to prevent such intractability by using quantum annealers: both the generator and the discriminator use quantum annealers to estimate their cost functions and the corresponding gradients efficiently compared to the classical case. First, we derive gradients that the HQuGAN with GRAPE requires, and show that they can be efficiently calculated using quantum annealers. We then analyze computational costs of the algorithm such as sample complexity and other classical/quantum resources.

A. Estimation on Gradients

The gradients for the generator's cost function in Eq.(12) to the first order of Δt becomes,

$$\begin{aligned}&\frac{\partial}{\partial \epsilon_i(t_j)} |\mathrm{Tr}(D(\rho(T)-\sigma))|^2 \\ &= 2\Delta t \Big(\mathrm{Tr}(D(\rho(T)-\sigma)) \Big) \Big[\mathrm{Tr}(\rho^{kj}_{i_+} D) - \mathrm{Tr}(\rho^{kj}_{i_-} D) \Big],\end{aligned} \tag{17}$$

where $\rho^{kj}_{i_\pm} = U(t_N)\cdots U(t_{j+1}) R^k_\alpha(\pm\pi/2) U(t_j)\cdots U(t_1)\rho_0$ $(U(t_N)\cdots U(t_{j+1}) R^k_\alpha(\pm\pi/2) U(t_j)\cdots U(t_1))^\dagger$ and R_α is a rotation about α axis that is determined by which Hamiltonian term ϵ_i is acting on. Hence, the gradient at time t_j can be calculated by estimating expectation values of operator D with respect to two quantum states $\rho^{kj}_{i_+} D$ and $\rho^{kj}_{i_-} D$. These quantum states can be prepared by implementing 3 unitary transformations on quantum annealers: 1) $U(t_N)\cdots U(t_{j+1})$, 2) $R^k_\alpha(\pm\pi/2)$, and 3) $U(t_j)\cdots U(t_1)$. Likewise, calculating cost functions only requires to use the annealer once.

B. Complexity Analysis

In this section, we analyze four different computational costs of the algorithm, of which quantum state tomography (QST) particularly considers: sample complexity, quantum resources, classical storage, and classical post-processing time. In fact, it is important to remark that both sample complexity and classical post-processing time heavily depend on a total number of iterations of the HQuGAN, denote by m, which is unknown even for general classical GANs. While the computational complexity of solving approximate local solutions in GAN setting has been studied [37], a precise bound is not known. This sets a barrier to compare the sample and time complexity of the HQuGAN to previously known bounds in a similar setting such as QST under adaptive measurement settings [38, 39] or quantum online learning [40].

1. Sample Complexity

In terms of sample complexity, recall that we need four distinct expectation terms to estimate in calculating Eq.(17), where each term would take $O(\|D\|^2/\epsilon^2)$ copies of either ρ_0 or σ with measurement precision ϵ. Similarly, the discriminator would take $O(\|D\|^2/\epsilon^2)$ copies of either ρ_0 or σ. Since we need to estimate the gradients for all N time grids in a single optimization step and $\|D\|^2 = 1$ ($\because D_0 = Z \otimes I^{\otimes n-1}$), the total number of copies of quantum states (either ρ_0 or σ) would be

$$O(\frac{mN}{\epsilon^4}), \tag{18}$$

where m is again the total number of iterations of the HQuGAN. In the first-order product formula, the trotter number $N = O(nT^2/\delta)$ to achieve trotter error δ [41]. For fixed T, therefore, the sample-optimal bounds on the sample complexity of QST [42] tells that m scales at least exponentially with n.

2. Quantum Resources

Regarding quantum resources, (to be added)

3. Classical Storage and Post-Processing Time

The maximum storage amount that HQuGAN requires is a gradient vector (or Hessian matrix if we utilize quasi-Newton optimization methods) of the cost function at each time grids for all control pulses. Since the HQuGAN can forget about past gradients values, the required classical storage is $O(\text{poly}(n))$. This shows a substantial improvement compared to the classical storage required by QST or self-guided QST, which are (at least) exponential to n. Classical post processing time taken by each round of HQuGAN scales polynomially with n as it manipulates $O(\text{poly}(n))$ sizes of matrix with $O(k)$ optimization steps, where k scales at most polynomially with n. In particular, for the L-BFGS method that we utilize for GRAPE in numerical simulations, k becomes the number of parameters, which is number of trotter steps times number of control fields so that $k = \text{poly}(n)$. Hence, the total classical post-processing time of the HQuGAN becomes $O(m\text{poly}(n))$.

As a summary, while the required number of iterations for convergence of the HQuGAN to learn general states is not know, there exists a clear, substantial advantage in terms of classical memory, which is polynomial in n. This is true for Krotov's method, and as well as for *any* Hamiltonian, as we are only interested in pulse sequences at each round of the algorithms.

VIII. CONCLUSION

We have introduced a new structure of QuGAN, the Hamiltonian QuGAN (HQuGAN), which updates continuous control pulse sequences of given time-dependent Hamiltonian, to learn arbitrary quantum states. This framework leverages techniques from QuGANs and quantum optimal controls (QOC), leading to new insights and methodology for learning quantum states under time-continuous quantum dynamics that can possibly extended to many real experimental settings. This was complemented with numerical results of generating various quantum states from arbitrary superposition states to 9-qubits GHZ state using two most popular gradient-based QOC methods, GRAPE and Krotov's method. In fact, by comparing the results to the ones with optimal discriminators, we numerically confirm that our discriminative models using QOC methods are indeed well-structured. In addition, we have explicitly shown that the mode collapse happens and disturbs the convergence of the HQuGAN, and provide a way to prevent the issue by introducing an absolute value symbol to the cost function of the minimax game, while still guaranteeing an existence of Nash equilibrium in the desired location.

Since computations for QOC methods become vastly intractable as system size increases, we remark that HQuGAN can use quantum annealers to estimate required cost functions and gradients efficiently compared to classical case. We also analyze computational costs of the algorithm. Since the total number of iterations of the HQuGAN is unknown, we notice that it is difficult to directly compare the costs to other algorithms such as quantum state tomography. Nevertheless, we remark that HQuGAN model only requires $O(\text{poly}(n))$ memories to store gradient (or Hessian) matrices, whereas every quantum state tomography methods requires a memory size of at least exponential to n.

We expect this work to lay the groundwork for potential explorations on QuGAN for many more interesting tasks under various setups. A natural follow-up study would be an application on our model to other uncontrollable Hamiltonian systems with various experimental setups, or an extension of our model to open quantum systems with different noise models. One can also ask to extend our ideas on learning an arbitrary unitary transformation to a more generalized setup [43] or even to learning quantum channels. Finally, it would be extremely intriguing to ask about finding rigorous bounds on the total iterations of (H)QuGAN for at least learning an arbitrary unknown quantum state, which would enable a more comprehensive study of generative quantum machine learning models.

[1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., 2014) p. 2672.

[2] W. Z. V. C. A. R. X. C. T. Salimans, I. Goodfellow and X. Chen, in *Advances in Neural Information Processing Systems*, Vol. 29 (Curran Associates, Inc., 2016) 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 (2016).

[4] C. Spampinato, S. Palazzo, P. D'Oro, D. Giordano, and M. Shah, Int. J. Comput. Vision **128**, 1378–1397 (2020).

[5] E. Putin, A. Asadulaev, Y. Ivanenkov, V. Aladinskiy, B. Sanchez-Lengeling, A. Aspuru-Guzik, and A. Zhavoronkov, *Journal of Chemical Information and Modeling*, Journal of Chemical Information and Modeling **58**, 1194 (2018).

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

[7] C. Zoufal, A. Lucchi, and S. Woerner, npj Quantum Information **5**, 103 (2019).

[8] H. Situ, Z. He, Y. Wang, L. Li, and S. Zheng, Information Sciences **538**, 193 (2020).

[9] M. Y. Niu, A. Zlokapa, M. Broughton, S. Boixo, M. Mohseni, V. Smelyanskyi, and H. Neven, Entangling quantum generative adversarial networks (2021).

[10] H.-L. Huang, Y. Du, M. Gong, Y. Zhao, Y. Wu, C. Wang, S. Li, F. Liang, J. Lin, Y. Xu, R. Yang, T. Liu, M.-H. Hsieh, H. Deng, H. Rong, C.-Z. Peng, C.-Y. Lu, Y.-A. Chen, D. Tao, X. Zhu, and J.-W. Pan, Physical Review Applied **16**, 10.1103/physrevapplied.16.024051 (2021).

[11] J. Preskill, Quantum **2**, 79 (2018).

[12] 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, Science advances **5**, eaav2761 (2019).

[13] K. Huang, Z.-A. Wang, C. Song, K. Xu, H. Li, Z. Wang, Q. Guo, Z. Song, Z.-B. Liu, D. Zheng, D.-L. Deng, H. Wang, J.-G. Tian, and H. Fan, npj Quantum Information **7**, 165 (2021).

[14] D. D'Alessandro, *Introduction to Quantum Control and Dynamics* (Chapman amp; Hall, 2007).

[15] S. J. Glaser, U. Boscain, T. Calarco, C. P. Koch, W. Köckenberger, R. Kosloff, I. Kuprov, B. Luy, S. Schirmer, T. Schulte-Herbrüggen, D. Sugny, and F. K. Wilhelm, The European Physical Journal D **69**, 279 (2015).

[16] J. Werschnik and E. K. U. Gross, Journal of Physics B: Atomic, Molecular and Optical Physics **40**, R175 (2007).

[17] 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, Nature **506**, 204 (2014).

[18] F. Dolde, V. Bergholm, Y. Wang, I. Jakobi, B. Naydenov, S. Pezzagna, J. Meijer, F. Jelezko, P. Neumann, T. Schulte-Herbrüggen, J. Biamonte, and J. Wrachtrup, Nature Communications **5**, 3371 (2014).

[19] J. Cheng, H. Deng, and X. Qia, in *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)* (2020) pp. 543–555.

[20] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, Journal of Magnetic Resonance **172**, 296 (2005).

[21] J. P. Palao and R. Kosloff, Phys. Rev. A **68**, 062308 (2003).

[22] T. Caneva, T. Calarco, and S. Montangero, Phys. Rev. A **84**, 022326 (2011).

[23] Y. Shi, N. Leung, P. Gokhale, Z. Rossi, D. I. Schuster, H. Hoffmann, and F. T. Chong, in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems* (ACM, 2019).

[24] D. Willsch, M. Nocon, F. Jin, H. De Raedt, and K. Michielsen, Phys. Rev. A **96**, 062302 (2017).

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

[26] M. Larocca, N. Ju, D. García-Martín, P. J. Coles, and M. Cerezo, Theory of overparametrization in quantum neural networks (2021).

[27] E. R. Anschuetz and B. T. Kiani, Beyond barren plateaus: Quantum variational algorithms are swamped with traps (2022).

[28] M. Larocca, P. Czarnik, K. Sharma, G. Muraleedharan, P. J. Coles, and M. Cerezo, Diagnosing barren plateaus with tools from quantum optimal control (2021).

[29] G. D. Palma, M. Marvian, D. Trevisan, and S. Lloyd, IEEE Transactions on Information Theory **67**, 6627 (2021).

[30] B. T. Kiani, G. De Palma, M. Marvian, Z.-W. Liu, and S. Lloyd, Learning quantum data with the quantum earth mover's distance (2021).

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

[32] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, SIAM Journal on Scientific Computing **16**, 1190 (1995), https://doi.org/10.1137/0916069.

[33] M. H. Goerz, D. Basilewitsch, F. Gago-Encinas, M. G. Krauss, K. P. Horn, D. M. Reich, and C. P. Koch, SciPost Phys. **7**, 80 (2019).

[34] D. M. Reich, M. Ndong, and C. P. Koch, The Journal of Chemical Physics **136**, 104103 (2012).

[35] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, Nature Communications **12**, 10.1038/s41467-021-21728-w (2021).

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

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

[38] S. T. Flammia, D. Gross, Y.-K. Liu, and J. Eisert, New Journal of Physics **14**, 095022 (2012).

[39] S. Chen, B. Huang, J. Li, A. Liu, and M. Sellke, Tight bounds for state tomography with incoherent measurements (2022).

[40] S. Aaronson, X. Chen, E. Hazan, S. Kale, and A. Nayak, Journal of Statistical Mechanics: Theory and Experiment **2019**, 124019 (2019).

[41] A. M. Childs and Y. Su, Physical Review Letters **123**, 10.1103/physrevlett.123.050503 (2019).

[42] J. Haah, A. W. Harrow, Z. Ji, X. Wu, and N. Yu, IEEE Transactions on Information Theory , 1 (2017).

[43] I. Marvian and S. Lloyd, Universal quantum emulator (2016).

[44] S. Tannor, Phys. Rev. A **66** (2002).

Appendix A: Helstrom Measurement Operator and Schatten p-norm

Here we show that the Helstrom measurement operator D, which always maximizes $\mathrm{Tr}(D(\rho-\sigma))$, has a simple form that depends on Schatten-p norm constraint of the operator when ρ and σ are pure states.

Theorem 1. *For any pure states ρ and σ, there exists a Hermitian operator D_{opt} such that*

$$\max_{||D||_p \leq 1} \mathrm{Tr}(D(\rho-\sigma)) = \mathrm{Tr}(D_{opt}(\rho-\sigma)) \tag{A1}$$

for $\forall p = 1, ..., \infty$. Indeed, such D_{opt} is

$$D_{opt} = 2^{-\frac{1}{p}}(|P+\rangle\langle P+| - |P-\rangle\langle P-|), \tag{A2}$$

where $|P\pm\rangle\langle P\pm|$ are projections onto positive and negative eigenspaces of $\rho-\sigma$ respectively.

Proof. We check whether D_{opt} saturates the bound of Holder's inequality for all $p = 1, ..., \infty$.

We denote $\rho - \sigma = \lambda(|P+\rangle\langle P+| - |P-\rangle\langle P-|)$ and therefore $||\rho-\sigma||_p = 2^{\frac{1}{p}}\lambda$. Also note that

$$||D_{opt}||_p = (|2^{-\frac{1}{p}}|^p + |2^{-\frac{1}{p}}|^p)^{1/p} = 1. \tag{A3}$$

Then observe:

$$\begin{aligned}\mathrm{Tr}(D_{opt}(\rho-\sigma)) &= \mathrm{Tr}(2^{-\frac{1}{p}}(|P+\rangle\langle P+| - |P-\rangle\langle P-|) \\ &\quad \cdot \lambda(|P+\rangle\langle P+| - |P-\rangle\langle P-|)) \\ &= 2^{-\frac{1}{p}} 2\lambda \\ &= 1 \cdot 2^{1-\frac{1}{p}}\lambda \\ &= ||D_{opt}||_p ||\rho-\sigma||_{\frac{1}{1-\frac{1}{p}}},\end{aligned} \tag{A4}$$

which always saturates the bound of the Holder's inequality for any $p = 1, ..., \infty$ ($\because 1 - \frac{1}{p} = \frac{1}{q}$). □

Appendix B: Mode Collapse

QuGAN might not always converge to a good Nash equilibrium point due to mode collapse. In this section, we review this issue raised in [9] and study how alternative cost functions can remedy this problem.

1. Mode Collapse Example

A major cause of the mode collapse lies in the fact that the generator ρ tries to always be aligned with the discriminator's previous operator D (assume D is a POVM) so that it perhaps overshoots the target state σ. A concrete example suggested in [9] considers a generator's initial state ρ and a target state σ as

$$\sigma = \frac{1 + \cos(\pi/6)X + \sin(\pi/6)Y}{2}, \tag{B1}$$

$$\rho = \frac{1 + \cos(\pi/6) - \sin(\pi/6)Y}{2}. \tag{B2}$$

Note that the optimization is subjected to

$$\min_{\theta_g} \max_{D} \mathrm{Tr}(D(\sigma - \rho(\theta_g))). \tag{B3}$$

Starting from discriminator, the minimax game goes as follows:

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

Step 2 (Generator): Generator tries to minimize (B3), which means maximizing $\mathrm{Tr}(D\rho(\theta_g))$, trying to make $\rho(\theta_g)$ parallel to the Helstrom measurement operator we found D_1. Hence, $\rho_1 = D_1 = (1+Y)/2$.

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

Step 4 (Generator): Again, the generator tries to align its state with $D_2 = \rho$, so that $\rho_2 = \rho$. Then, we go back to step 1 and repeat forever.

2. Cost Function with Absolute Value

We can prevent the aforementioned overshooting issue by applying the absolute value symbol to the cost function that the generator minimizes. Hence, instead of the generator always getting aligned with D, it now depends on $\mathrm{Tr}(D\sigma)$, preventing the overshooting issue. For the same example, we show that such cost function is able to prevent the mode collapse.

Suppose the generator now uses a new cost function

$$\min_{\theta_g} |\mathrm{Tr}(D(\sigma - \rho(\theta_g)))|^2. \tag{B4}$$

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

Step 2 (Generator): Unlike before, the generator this time tries to make $\mathrm{Tr}(D_1\rho(\theta_g))$ equal to $\mathrm{Tr}(D_1\sigma)$, which is in this case $3/4$. The generator will make $\rho(\theta_g)$ such that $\mathrm{Tr}(D_1\rho(\theta_g)) = 3/4$, which obviously has many solutions. Since $\mathrm{Tr}(D_1\rho_1) = 1/2 + \mathrm{Tr}(\rho_1 Y/2)$ should be $3/4$, solutions from the generator ρ_1 are density matrices that satisfy $\mathrm{Tr}(\rho_1 Y/2) = 1/4$. Hence, the general solution would be $\rho_1 = (1 + a_x X + Y/2 + a_z Z)/2$, where $a_x^2 + a_z^2 \leq 3/4$. Setting $a_x = \cos(\pi/6)$ and $a_z = 0$ gives us $\rho_1 = (1 + \cos(\pi/6)X + Y/2)/2$, which is σ.

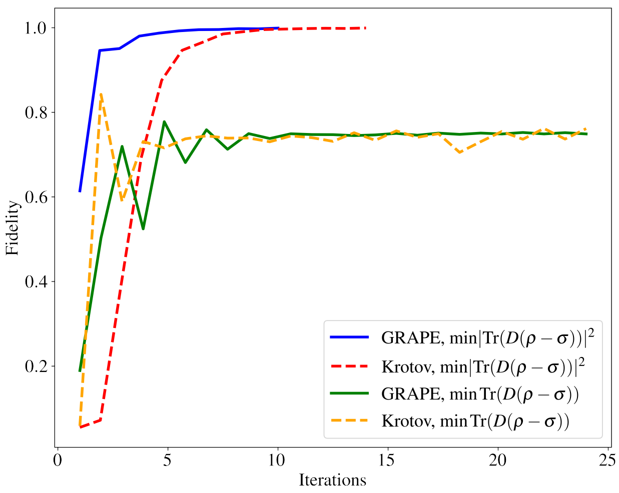

FIG. 3: **Mode collapse on the HQuGAN.** Here we present our numerical results of running HQuGAN with both GRAPE (solid line) and Krotov's method (dashed line) with two different cost functions: the original cost function of Eq.(B3) and our modified cost function of Eq.(B4), to generate the GHZ state. Regardless of which QOC methods is used, the mode collapse occurs when the generator minimizes the original cost function, oscillating between two states give constant overlaps with the GHZ state. However, introducing the absolute value to the generator's cost function solves the mode collapse issue.

3. Mode Collpase on the HQuGAN

Here we give a concrete example of the mode collapse issue in both GRAPE and Krotov's method to generate the GHZ state, where the system setups remain unchanged from SectionVI A. Using the original cost function without the absolute value symbol, the HQuGAN happens to fall into the mode collapse problem, oscillating between two states with constant overlap of around 3/4 to the GHZ state, as shown in Fig.3. On the other hand, the plot shows that the generators minimizing the cost function with the absolute value symbol indeed make the algorithm to converge to the Nash equilibrium. As said earlier, the new cost function makes the generator to not get aligned to the discriminator's operator, avoiding the overshooting issue. In fact, the mode collapse also occurs for generating arbitrary quantum states other than just the GHZ states.

Appendix C: Krotov's Method

In this section, we describe more details on Krotov's method, another popular line of gradient-based QOC technique. Krotov's method is fundamentally different from GRAPE in a sense that it is a sequential algorithm: an update of pulse at time t_j depends on previously updated pulses at t_k for $\forall k < j$. Also, it is highly constructive method. One appealing aspect of Krotov's method is that it assumes time-continuous control fields. So we conduct the same tasks of generating different quantum states using Krotov's method. In addition, we perform simulations on similar task but with limited control amplitudes.

1. Krotov's Method

Krotov's method [21] is based on rigorous examination of the conditions for calculating the updated 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 [21],

$$J[\{\epsilon_i(t)\},\{|\psi(t)\rangle\}] = J_T[\{|\psi(T)\rangle\}] + \int_0^T dt\; g[\{\epsilon_i(t)\},\{\rho(t)\},t], \tag{C1}$$

where J_T is the main objective functional that depends on the final time T (e.g. F in Eq. (8)) 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.

In order to minimize the field intensity and to switch the field smoothly on and off, g_a is typically chosen as the following form [34],

$$g_a[\{\epsilon(t)\}] = \frac{\lambda_a}{S(t)}[\epsilon(t) - \epsilon_{ref}(t)]^2, \tag{C2}$$

where $\epsilon_{ref}(t)$ denotes some reference field, $S(t)$ is a shape function and λ is a weight. Given such specific choice of the functional g_a, Krotov's method updates control fields [21, 44]

$$\Delta\epsilon_i^{(k)}(t) = \frac{S_i(t)}{\lambda_i}\mathrm{Im}\left[\langle\chi^{(k-1)}(t)|\left(\frac{\partial H}{\partial\epsilon_i(t)}\right)\bigg|_{(k)}|\phi^{(k)}(t)\rangle\right], \tag{C3}$$

where $S_i(t)$ and λ_i are shape functions and step width, respectively. $|\phi^{(k)}(t)\rangle$ 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:

$$\frac{\partial}{\partial t}|\phi^{(k)}(t)\rangle = -\frac{i}{\hbar}H^{(k)}|\phi^{(k)}(t)\rangle, \tag{C4}$$

$$\frac{\partial}{\partial t}|\chi^{(k-1)}(t)\rangle = -\frac{i}{\hbar}H^{\dagger(k-1)}|\chi^{(k-1)}(t)\rangle + \frac{\partial g_b}{\partial\langle\phi|}\bigg|_{(k-1)}, \tag{C5}$$

with boundary condition of

$$|\chi^{(k-1)}(T)\rangle = -\frac{\partial J_T}{\partial\langle\phi(T)|}\bigg|_{(k-1)}. \tag{C6}$$

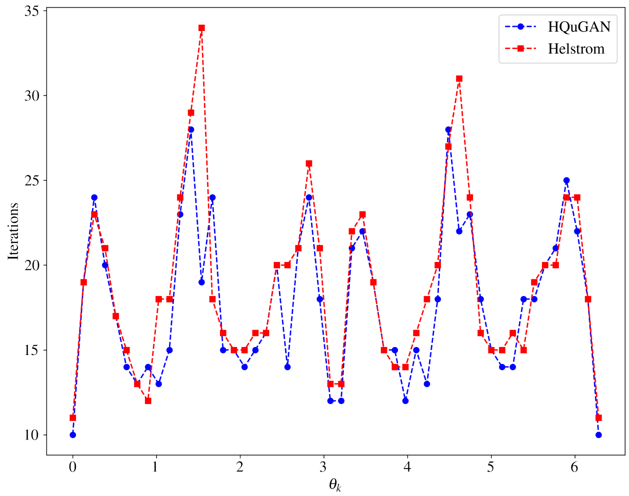

FIG. 4: **HQuGAN for learning 3-qubits states using Krotov's method.** We display our numerical results on generating 50 different 3 qubit pure states described in Eq.(15) for two different discriminators: a discriminator with the cost function without absolute value (blue) and an optimal discriminator with the Helstrom measurement operator (red), using Krotov's method. The two cases give extremely similar behaviors.

The optimization scheme of Krotov's method proceeds by first constructing $|\chi^{(k-1)}(T)\rangle$ according to Eq.(C6) and perform a backward propagation using Eq.(C4) over entire time grid from $t = T$ to 0, storing all states at each time grid. Then, starting from a given initial state $|\phi^{(k)}(0)\rangle$, update a control pulse at first time grid $t = t_1$ using the update equation Eq.(C3). From this updated control field, we obtain a time-evolved state $|\phi^{(k)}(t_1)\rangle$. Then, we update the next pulse (at second time grid) by using the update equation with $|\phi^{(k)}(t_1)\rangle$ we just calculated. After repeating this procedure for all N time grids, we are finished with a single update procedure of Krotov's method.

Hence, Krotov's method is sequential: the update at time t depends on the states forward-propagated using the updated controls at all previous times, whereas GRAPE updates control pulse for all time grids concurrently. Consequently, Krotov's method is computationally much heavier as system size increases, as its single optimization step requires to classically solve Schrödinger equations $2N$ times. To prevent intractable classical computations of solving the equations multiple times, we can also consider implementing Krotov's method on quantum computers. For every update on pulse at each time grid, we only need 1 unitary operation to implement using continuous control pulse. If we were to perform this task in circuit model, on the other hand, we need to implement k quantum gates to update a pulse at t_k.

n	Iter (HQuGAN)	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	(still running)	361	30	300

TABLE II: **HQuGAN for learning generalized GHZ states using Krotov's method.** Here we present our results on learning the generalized n-qubits GHZ states using the HQuGAN with Krotov's method and the discriminator being optimal. Likewise to GRAPE experiments, the HQuGAN with Krotov's method also displays a comparable number of iterations for all instances.

2. Numerical Results using Krotov's Method

In this section, we present our numerical results of generating quantum states using the HQuGAN with Krotov's method under the same setups described in Section VI A. Similar to GRAPE experiments, the generator uses Krotov's method to minimize $|\mathrm{Tr}(D(\rho(\theta_g)-\sigma))|^2$ and the discriminator maximizes $\mathrm{Tr}(D(\theta_d)(\rho-\sigma))$, where $D(\theta_d) = U^\dagger D_0 U$ with $D_0 = Z \otimes I^{\otimes n-1}$. First, we reproduce the same task of generating 50 different 3 qubit states as in Eq.(15). As shown in Fig.4, not only the HQuGAN with Krotov's method successfully generates all of the pure states, but it also gives an extremely similar result to the one with optimal discriminator. In addition, generating a generalized n-qubits GHZ states is summarized in Table II. As the table presents, the HQuGAN with Krotov's method also gives a similar number of iterations to the ones that use optimal discriminator.

3. Numerical Results with Limited Control Fields

We present two additional numerical results with bounded control fields, as some experimental setups may require limited resources or constraints on the pulse amplitudes. First, we bound the control amplitudes by ±1 and try to generate the n-qubits GHZ states using the Helstrom measurement operator. To reduce simulation time, we use less number of steps for generator for high number of qubits. In Table III, we summarize the numerical results of learning up to 9-qubits GHZ states. Second, we consider an experimental setup by setting a coefficient for ZZ-interaction term in Eq.(13) to be $J = 100$ (MHz) and limiting control fields $|\epsilon_i(t)| \leq 1$ (GHz). We try to learn the GHZ states for various terminal time from $T = 20$ to 100 (ns). We use the Helstrom measurement operators with fixed generator's step of 10 for all instances. As shown in Fig.5, the HQuGAN is able to produce the GHZ state properly for all instances. As we

n	Gen. it	Tot. it	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

TABLE III: **HQuGAN with Krotov's method for learning n-qubits GHZ states using the Helstrom measurement operator.** We display our numerical results of the HQuGAN with Krotov's method to learn n-qubits GHZ states up to $n = 9$ with limited control amplitudes. For ease of computations, we use the Helstrom measurement operator for all instances, which we expect to give similar results to the HQuGAN with discriminator using Krotov's method, as we numerical confirm in previous results. We reduce the number of generator's steps for ease of computations as well.

increase evolution time from $T = 20$ to 70, a number of iterations decreases monotonically, and stays around the similar numbers after then.

4. Parameters of Krotov's Method

In this section, we describe several parameters for Krotov's method we set to achieve previous numerical results. First, in order for the method to begin the optimization process for either generator or discriminator, it primarily requires to backpropagate the Schrödinger equation starting from the boundary condition state $|\chi(T)\rangle$ in Eq.(C6) (for $k = 1$) that 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

$$|\chi_k(T)\rangle = -\frac{\partial J_T}{\partial \langle\psi(T)|} = -2\,\mathrm{Tr}(D(\rho(T) - \sigma))(D\,|\psi(T)\rangle), \tag{C7}$$

and we can similarly define the boundary state for the discriminator.

In addition, there are two main hyperparameters we have to set carefully, which are shape function $S(t)$ and step width λ. The shape function contributes how much the pulse is updated in each time grid and is encouraged to be smoothly switched on and off around $t = 0, T$ to make the optimized pulse smooth as well, ensuring the boundary condition of pulses, if needed. The step width λ determines the overall magnitude of $\Delta\epsilon$ as can be observed in Eq.(C3). If λ is too large, then the amount of pulse updates $\Delta\epsilon$ would be very small, causing a slow convergence. If λ is too small, on the other hand, then $\Delta\epsilon$ would be very large, causing instability in numerical simulations [33].

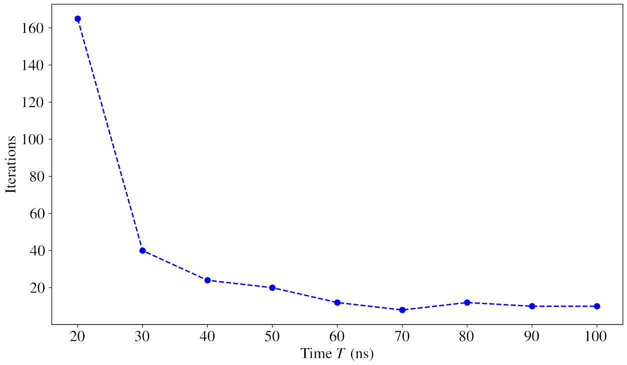

FIG. 5: **HQuGAN for learning the GHZ states using Krotov's Method under experimental parameters.** For Hamiltonian described in Eq.(13), we set $J = 100$ (MHz) and aim to generate the GHZ state by limiting all control amplitudes to be $|\epsilon_i(t)| \leq 1$. The algorithm was performed for different evolution time from $T = 20$ to 100 (ns). The step width is set to be the same for all cases, as $\lambda = 10$.

Shape function $S(t)$ is chosen to be consistent throughout all simulations done via Krotov's method, as a flattop function:

$$S(t) = \begin{cases} \sin^2(\pi t/(2\ \mathrm{t.rise})), & \text{if } t \leq \mathrm{t.rise} \\ \sin^2(\pi(t-T)/(2\ \mathrm{t.fall})), & \text{if } t \geq T - \mathrm{t.fall} \\ 0 & \text{if } t = 0 \text{ or } t = T \\ 1 & \text{otherwise}, \end{cases} \tag{C8}$$

where t.rise = t.fall = $T/20$. As there is no rigorous method to find an ideal value for the step width λ, we found proper values of λ for different cases by trials and errors. For learning $n = 1, \ldots, 6$ qubits GHZ states, we set $\lambda = 2, 5, 5, 10, 50, 50$, respectively. For numerical results with limited controls, we set $\lambda = 2$ and 10 to generate Table III and Fig.5, respectively.

Appendix D: Quantum W_1 Distance

In this section, we display numerical results on the HQuGAN for learning n-qubits GHZ states from $n = 1, \ldots, 5$ using the quantum W_1 distance described in Eq.(3). The simulation was performed with both GRAPE and Krotov's method. To make fair comparisons to previous simulation results, we kept every parameters to be the same as previous simulations, except, for Krotov's method, we set step widths λ to be larger compared to λ we used in previous numerical results in Section C 2 and C 3. Since a norm of an operator that discriminator using quantum W_1 distance is typically larger than the norm of the Helstrom operator, we observe that λ is too small for some cases, causing the numerical instability. So we choose $\lambda = 5, 10, 20, 50, 50$ for quantum W_1 dis-

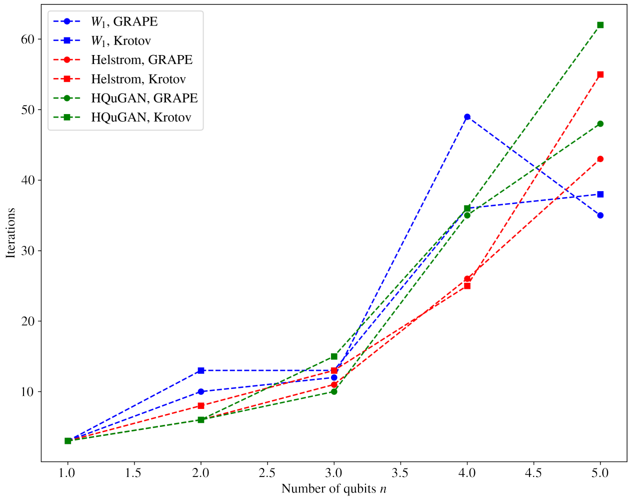

FIG. 6: **HQuGAN for learning n-qubits GHZ states using quantum W_1 distance.** We compare number of iterations of HQuGAN using quantum W_1 distance to the previous results of learning n-qubits GHZ states. To make fair comparisons, we set every hyper-paramters to be unchanged, except step widths λ for Krotov's method due to larger norms of operators that discriminator with quantum Lipschitz constraint produces. Unlike previous results, which show monotonic increasing behaviors of numbers of iterations of HQuGAN, the results of quantum W_1 distance gives non-monotonic behaviors and even less number of iterations for learning $n = 5$-qubits GHZ states.

tance with Krotov's method. The results are summarized and compared with previous results in Fig.6. While numbers of iterations for both discriminators using the Helstrom operator (red) and QOC techniques (green) show monotonically increasing behaviors, discriminators using quantum W_1 distance non-monotonic behaviors and even result less number of iterations for $n = 5$-qubits system.

Appendix E: Trotter Error

In this section, we numerically analyze a trotter error on control pulses that the HQuGAN with GRAPE optimizes to achieve > 0.999 fidelity by calculating a fidelity with a continuous version of the optimized pulse sequences. As shown in Table IV, we first train the HQuGAN to achieve > 0.999 fidelity with different trotter steps N from 50 to 500 for fixed terminal time $T = 5$. We then check whether the continuous version of the pulse still gives > 0.999 fidelity by transforming the optimized pulses to the continuous pulses via linear interpolations for $N = 5000$. We observe that from $N = 200$, the optimized pulses and their continuous versions give similar fidelity > 0.999 and thus ensure that our optimized pulses from the HQuGAN is able to considered as continuous control pulses. For general n-qubits system, the trotter error $\delta = O(T^2/N)$ [41]. (More in-depth analysis on such results are to be added)

Number of time grids (N)	Fidelity $\lvert\langle\psi(T)\vert\psi_{targ}\rangle\rvert^2$	
	Trotterization (GRAPE)	Continuous ($N = 5000$)
50	0.9997	0.9746
100	0.9992	0.9942
150	0.9992	0.9972
200	0.9994	0.9994
250	0.9994	0.9995
300	0.9991	0.9991
350	0.9994	0.9994
400	0.9990	0.9992
450	0.9994	0.9994
500	0.9994	0.9994

TABLE IV: **Comparisons between discrete and continuous control sequences.** We compare results between optimized pulse with different trotter steps N and its corresponding continuous version of pulse sequences. We primarily train the HQuGAN to learn the GHZ state with > 0.999 fidelity for different N. Then, we interpolate the optimized pulse sequences to make it continuous and re-calculate the fidelity between the GHZ state and the state generated from the continuous pulse sequences. As shown in the table, from $N = 200$, we numerically confirm that the optimized control sequences from the HQuGAN can be treated as continuous pulse sequences.

Appendix F: Other Applications of HQuGAN

1. Learning Unitary Transformation

In this section, we extend our previous methods on learning arbitrary quantum states to learn arbitrary unitary transformation using the HQuGAN with GRAPE. We consider learning a desired unitary operation given pairs of quantum states or the Choi matrix for the unitary. Below, we aim to generate a Hadamard gate based on the two different settings.

First, we are given with three pairs of quantum states $\{|0\rangle, |+\rangle\}, \{|1\rangle, |-\rangle\}$, and $\{|+\rangle, |0\rangle\}$. While the first two pairs are insufficient to uniquely characterize the Hadamard gate due to non-zero values of relative phases between the pairs, the third pair ensures that relative phase to be 0 and thus the three pairs uniquely defines the Hadamard gate. Given such pairs, the HQuGAN now optimizes the following cost function:

$$\min_{\theta_g}\max_{\theta_{d_k}}\sum_{k=1}^{3}\lvert\mathrm{Tr}(D_k(\theta_k)(U(\theta_g)\rho_k U^{\dagger}(\theta_g)-\sigma_k))\rvert^2, \tag{F1}$$

where $(\rho_k, \sigma_k) \in \{(\left|0\right\rangle\left\langle 0\right|, \left|+\right\rangle\left\langle +\right|), (\left|1\right\rangle\left\langle 1\right|, \left|-\right\rangle\left\langle -\right|), (\left|+\right\rangle\left\langle +\right|, \left|0\right\rangle\left\langle 0\right|)\}$ and $D_k(\theta_d) = U^\dagger(\theta_{d_k}) D_0 U(\theta_{d_k})$. So this simply is a linear combination of cost functions that we use to generate a single quantum state. The generator tries to find a unitary that maps the three input states to the proper final states respectively, and three discriminators find each D_k that separates the corresponding pair of quantum states.

In the second setting, we aim to generate the Hadamard gate where the Choi matrix for the Hadamard gate,

$$\mathcal{C}(H) = (I \otimes H) \left|\Omega\right\rangle\left\langle\Omega\right| (I \otimes H^\dagger), \tag{F2}$$

where $\left|\Omega\right\rangle$ is the Bell pair, is given. So we now consider the Choi matrix $\mathcal{C}(H)$ as a target state. That being said, the generator aims to generate a unitary $U(\theta_g)$ such that $|\mathrm{Tr}(D(\rho(\theta_g) - \mathcal{C}(H)))|^2$ is minimized, where $\rho(\theta) = (I \otimes U(\theta_g) \left|\Omega\right\rangle\left\langle\Omega\right| (I \otimes U^\dagger(\theta_g))$, and the discriminator finds a unitary $U(\theta_d)$ such that $\mathrm{Tr}(D(\theta_g)(\rho(\theta_g) - C(H)))$ is maximized.

In both settings, the HQuGAN with GRAPE successfully generates the Hadamard gate. We take $T = 5$ and $N = 1000$ for both cases. The HQuGAN took 2 and 1 iterations to achieve the Hadamard gate for the first and the second settings, respectively. The first setting requires to use 3 distinct discriminators that discriminate 3 different pairs of quantum states. This implies that for given k pairs of quantum states, this would require k distinct discriminators. The second setting needs only 1 discriminator, but the HQuGAN is performed on $2n$-qubits system to learn n-qubits unitary transformation.