Efficient Online Quantum Generative Adversarial Learning Algorithms with Applications

Yuxuan Du[*], Min-Hsiu Hsieh[†], Dacheng Tao[*]

[*]UBTECH Sydney Artificial Intelligence Centre and the School of Information Technologies, Faculty of Engineering and Information Technologies, The University of Sydney, Australia

[†]Centre for Quantum Software and Information, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia

Abstract

The exploration of quantum algorithms that possess quantum advantages is a central topic in quantum computation and quantum information processing. One potential candidate in this area is quantum generative adversarial learning (QuGAL), which conceptually has exponential advantages over classical adversarial networks. However, the corresponding learning algorithm remains obscured. In this paper, we propose the first quantum generative adversarial learning algorithm—the quantum multiplicative matrix weight algorithm (QMMW)—which enables the efficient processing of fundamental tasks. The computational complexity of QMMW is polynomially proportional to the number of training rounds and logarithmically proportional to the input size. The core concept of the proposed algorithm combines QuGAL with online learning. We exploit the implementation of QuGAL with parameterized quantum circuits, and numerical experiments for the task of entanglement test for pure state are provided to support our claims.

I. INTRODUCTION

The principal interest in quantum computation is the exploration of potential applications that outperform their classical counterparts. The rapid development of quantum hardware divides this interest into short-term and long-term goals. The short-term goal is to devise quantum algorithms that not only possess quantum advantages but can also be implemented on near-term devices [36]. The long-term goal is to employ fault-tolerant quantum computers that are capable of providing remarkable quantum speedups over classical methods [38] to tackle practical real-world problems.

Quantum machine learning is one of the most promising candidates for achieving both short-term and long-term goals [5], and the proposed quantum generative adversarial learning (QuGAL) strengthens this

belief [29]. The main theoretical conclusion of QuGAL is that exponential quantum advantages may exist under the assumption that the target data distribution can be efficiently encoded into a density matrix [29]. Conceptually, QuGAL involves two players, a generator and a discriminator, which play a zero-sum game. At each training round, the generator tries to approximate the target data to fool the discriminator, while the discriminator tries to distinguish the fake data from the real data. When the generator and discriminator are both constructed by quantum operations, the adversarial quantum learning game has the potential to converge to Nash equilibrium with an exponential speedup.

Despite promising theoretical results, two issues related to QuGAL have not been explored. First, it is unclear what kinds of learning tasks can be accomplished by QuGAL to potential advantages. Second, an explicit learning algorithm of QuGAL that can fast converge to the equilibrium remains unexplored. Previous studies mainly focus on the implementation of QuGAL under near-term quantum devices as so-called quantum generative adversarial networks (QuGANs) [14], [39], [45], [37], [47]. In particular, the generator and discriminator of QuGANs are constructed by employing parameterized quantum circuits (PQCs) that are composed of a set of trainable parameterized single qubit gates and two-qubit CNOT gates [16]. However, the intrinsic optimization mechanism of PQCs that iteratively updates each gate destroys the required convex-concave property in QuGAL, which implies that the obtained result may not converge to Nash equilibrium and may induce additional training difficulties, e.g., mode collapse and vanishing gradients [2]. Two key issues therefore exist for QuGANs, i.e., how to improve stability and convergence in training QuGANs, and whether QuGANs deliver potential quantum advantage.

To tackle the aforementioned issues, we revisit the theory of QuGAL in this paper from the perspective of online learning [23]. The integration of online learning with QuGAL is motivated by the fact that online learning algorithms can efficiently approximate the optimal result for the zero-sum game associated with the convex-concave property, and the training of QuGAL satisfies this condition. This observation enables us to devise a quantum adversarial learning algorithm with online learning features and to theoretically analyze its potential quantum advantages. Additionally, online learning has been employed as a powerful tool for relieving training difficulties in classical generative adversarial networks (GANs) [20], which motivates us to introduce such a method in optimizing QuGANs. Lastly, we investigate how to use QuGAL to accomplish learning quantum information processing tasks, such as the quantum entanglement test for pure state and quantum state discrimination [25], [12], [10], [11], [13]. Our study opens avenues for exploring quantum information processing tasks using quantum generative adversarial learning models.

We summarize the main results of this work as follows.

- We propose a quantum generative adversarial learning algorithm, the quantum multiplicative matrix weight (QMMW) algorithm, which rapidly converges to Nash equilibrium as expected from QuGAL.

QMMW is inspired by the multiplicative matrix weight algorithm, which is a popular online learning algorithm that efficiently finds optimal solutions to the zero-sum game [28]. We prove that the convergence rate of QMMW is $\mathcal{O}(\sqrt{N/T})$, where N is the number of qubits corresponding to the target density matrix and T is the number of training rounds. An attractive feature of QMMW is that the output states of both the generator and discriminator can be viewed as Gibbs states. By exploiting the efficient Gibbs sampling method proposed by [41], we prove that the computational complexity of QMMW is $\mathcal{O}(N^3 T^4)$.

- We introduce a multiplicative weight training method to overcome the training difficulty encountered in QuGANs. The core ingredient of this method is to seek the most possible optimized direction for achieving global equilibrium through the inherent mechanism of online learning. In the training process, a multiplicative weight training method puts more weight to the gradient that is more probable to fool the discriminator. Since the multiplicative weight training method only focuses on re-weighting the gradient, it can be seamlessly embedded into other optimization methods used in QuGANs.

- We investigate the potential quantum advantages by applying QMMW and QuGANs to solve quantum information tasks, i.e., the pure state entanglement test and the quantum state discrimination. In particular, we numerically validate that QuGANs are capable of accomplishing the pure state entanglement test with modest quantum resources, which sheds light on using QuGANs to handle other quantum information learning tasks. All numerical simulations demonstrated in this paper are implemented in Python, leveraging the pyQuil and QuTiP libraries to access the numerical simulators [40], [27].

A. Related Works

Online convex optimization has been broadly applied to the study of linear programming, semidefinite programming, and zero-sum game [28], [23]. Recently, it has been employed to study shadow quantum tomography [1]. An advanced meta-algorithm in online convex learning, the so-called multiplicative weight, has been introduced to study the quantum zero-sum game algorithm [42], non-interactive zero-sum quantum games [26], the parallel approximation of semidefinite programs and minmax problems [21], and quantum semi-definite programming [7]. Despite their similarities, the various studies, including this work, have adopted different update rules and focused on different tasks, leading to distinct theoretical results on, for example, convergence rate.

In the rest of this paper, we first introduce QuGAL and discuss its applications on quantum information processing in Section II. In Section III, we describe QMMW and theoretically analyze its computational

cost. In Section IV, we give the multiplicative weight training method for QuGANs. In Section V, we explain how to employ QMMW and QuGANs to tackle quantum information processing tasks. In Section VI, we numerically validate the effectiveness of applying QuGANs to accomplish quantum information processing tasks. Section VII concludes the paper.

II. QUANTUM GENERATIVE ADVERSARIAL LEARNING AND ITS APPLICATIONS

We formally define the quantum generative adversarial learning (QuGAL) problem and devise a general framework for using QuGAL to accomplish quantum information processing tasks. Suppose that a given mixed state ρ is represented by N qubits, the goal of QuGAL is to reproduce ρ. QuGAL employs two players to set up a zero-sum game [34]: The first player refers to a generator, which generates a mixed state σ to approximate ρ; the second player refers to the discriminator $\mathcal{D}$, which aims to maximally distinguish ρ from σ. Such a zero-sum game is evaluated by a loss function $\mathcal{L}(\cdot, \cdot)$, where its physical meaning is the classification error. In the training process, the generator tries to minimize the loss function while the discriminator tries to maximize it. By labeling the state ρ as 'True' and the generated state σ as 'False', we have

$$\mathcal{L}(\sigma, \mathcal{D}) = P(\text{True}|\sigma)P(G) + P(\text{False}|\rho)P(R) \ , \tag{1}$$

where $P(G)$ ($P(R)$) refers to the prior of operating the discriminator with σ (ρ), and $P(\text{True}|\sigma)$ ($P(\text{False}|\rho)$ refers to the likelihood that the discriminator will classify σ (ρ) as 'True' ('False'). Throughout this paper, we set $P(G) = P(R) = 1/2$.

When the discriminator is assigned to be positive operator value measurement (POVM), we have $P(\text{True}|\sigma) = \text{Tr}((\mathbb{I} - \mathcal{D})\sigma)$ and $P(\text{False}|\rho) = \text{Tr}(\mathcal{D}\rho)$, where the corresponding loss function possesses the convex-concave property. This property immediately indicates that equilibrium always exists, guaranteed by the theoretical result of the convex optimization [6]. Denoting the optimal solution as $(\sigma^*, \mathcal{D}^*)$ with $(\sigma^*, \mathcal{D}^*) = \arg\max_\sigma \min_\mathcal{D} \mathcal{L}(\sigma, \mathcal{D})$, we have $\sigma^* = \rho$ with $\mathcal{L}(\sigma^*, \mathcal{D}^*) = 1/2$ at the equilibrium point.

Despite this promising property, the means of applying QuGAL to solve certain problems with an exponential quantum advantage is unknown. In the following, we propose a general principle for employing QuGAL to solve quantum information processing problems [33]. Let us recall a common strategy of conventional methods in quantum information processing tasks, e.g., quantum entanglement test or quantum state discrimination [3], [25]. A conventional method generally has two steps: Extraction of quantum information into the classical forms, followed by manipulation of the collected classical data into the desired result. Due to the curse of dimensionality, the number of measurements required to collect a sufficient amount of quantum information grows exponentially with respect to the number of qubits. In contrast, the desired result is often unrelated to the size of the input and can be represented in a low

dimensional space. For instance, the outcome of an entanglement test is binary, indicating whether the input state is entangled or not. Applying QuGAL to manipulate quantum data and output the result directly could immediately have exponential quantum advantage, assured by circumventing the enormous amount of quantum measurements required by conventional methods.

The key issue, given this observation, is identifying how to reformulate a given quantum information processing task as quantum generative adversarial learning language. Here we devise a general framework to achieve this goal. The central idea behind this framework is to conditionally limit the expressive power of the generator or the discriminator for a given task. Adopting the constraint operation aims to distinguish the desired answer from other results, where the training loss of QuGAL will conditionally converge to the Nash equilibrium if and only if the given input directly relates to the desired answer. In other words, the convex-concave property of the training loss defined in Eqn. (1) is conditionally preserved when the input directly relates to the desired answer. We outline the framework as follows. First, we translate a quantum information processing task into a binary decision problem, which can be effectively achieved by employing the 'one-versus-all' strategy [44]. We then constrain the expressive power of the generator or the discriminator, e.g., the generator can only well approximate all possible inputs corresponding to the desired answer. If the discrepancy between the obtained loss and the optimal loss is below a certain threshold after training,, the desired outcome is obtained. The restriction of the expressive power particularly depends on the detailed setting and implementation of QuGAL, and we will illustrate how to limit the expressive power in the following sections.

We illustrate how to tackle entanglement test problems under the proposed QuGAL framework. Let us briefly review the entanglement test. The entanglement test targets to the detection of whether a given quantum state is entangled or separable. Devising an efficient separability criteria to distinguish entanglement for specific quantum states is fundamentally important for quantum applications. Previous separability criteria can be roughly divided into two classes [25]: (1) The separability criteria are efficient but incomplete, that is, some entangled states could be misclassified as separable states, e.g., positive partial transposition [24]. (2) The separability criteria are complete in the sense that they are capable of correctly identifying any entangled states at any dimensions, but the computational cost is very high, e.g., symmetric extension [15]. Besides these two conventional classes, machine learning, which has been explored as an effective tool in many physics problems [9], [43], also provides novel insights into the tasks of the entanglement test [30], [31]. A common weakness of the above methods is the requirement for quantum state tomography to construct the classical density matrix, which leads to an exponential runtime with a linearly increased number of qubits.

QuGAL is a potential candidate for overcoming the aforementioned issue by directly manipulating

the quantum data to circumvent the time-consuming quantum state tomography. The separability rule of QuGAL is reflected by the training loss such that the given state is classified as being entangled if the training loss cannot converge to the equilibrium below a threshold after a certain number of training rounds. The detailed procedure of employing QuGAL to tackle the entanglement test is as follows. Given an unknown quantum state ρ, the entanglement test asks if ρ is entangled or not, which is a binary decision problem. In this setting, we restrict the expressive power of the generator to only generate separable states. If ρ is separable, the convex-concave property of QuGAL is preserved, where σ output by the generator can efficiently approximate ρ and the loss converges to the Nash equilibrium very quickly. Otherwise, the convex-concave property is lost and the Nash equilibrium can never be reached. The fact that the loss of QuGAL can be efficiently calculated by two outcome measurements immediately gives QuGAL an exponential quantum advantage over conventional methods, which require exponential measurements with respect to the number of qubits.

III. Quantum Multiplicative Matrix Weight

The convex-concave property of QuGAL enables us to ues the results of convex optimization, under the no-regret framework for online learning [23], to develop an advanced quantum algorithm that is capable of fast convergence to the equilibrium. We first give the definition of *regret* before moving on to explain how no-regret learning algorithms work. Given a sequence of convex loss functions $\{\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_T\} : \mathcal{K} \to \mathbb{R}$, an algorithm $\mathcal{A}$ selects a sequence of $K_t \in \mathcal{K}$'s with $\mathcal{K}$ being the input space, each of which may only depend on previously observed $\{\mathcal{F}_1, ..., \mathcal{F}_{t-1}\}$. The algorithm $\mathcal{A}$ is said to have *no regret* if its minimized regret $\min_K R_T(K) = \mathcal{O}(T)$, where we define $R_T(K) := \sum_{t=1}^{T}(\mathcal{F}_t(K_t) - \min_{K \in \mathcal{K}} \mathcal{F}_t(K))$.

Here we propose a no-regret quantum generative adversarial learning algorithm—the quantum multiplicative matrix weight (QMMW) algorithm—to efficiently reconstruct the given mixed state under the fault-tolerant quantum circuits setting. Conceptually, QMMW is inspired by the multiplicative matrix weight algorithm [28], an advanced meta-algorithm with the no-regret property that is broadly used in online convex optimization [23].

Before presenting the technical treatment, we explicitly define the generated state, the discriminator, and the loss function in Eqn. (1) used in QMMW. We denote the output states of the generator and the discriminator as σ_G and σ_D, respectively. The loss function at t-th round $\mathcal{L}(\sigma_G^{(t)}, \sigma_D^{(t)})$ is

$$\mathcal{L}(\sigma_G^{(t)}, \sigma_D^{(t)}) = \frac{1}{2} \left(\text{Tr}\left(\sigma_D^{(t)}\rho\right) - \text{Tr}\left(\sigma_D^{(t)}\sigma_G^{(t)}\right) \right) + \frac{1}{2} . \tag{2}$$

The physical meaning of this loss function is the evaluation of the overlap between ρ and σ_G using $\sigma_D^{(t)}$ [1]. The convexity of trace calculation implies that the loss function defined in Eqn. (2) has the convex-concave property, where the optimal solution is $\sigma_G^* = \rho$ with the equilibrium $\mathcal{L}(\sigma_G^*, \sigma_D^*) = 1/2$.

Following the theoretical results of approximating Nash equilibrium, an algorithm that has the no-regret property will quickly converge to the equilibrium [17]. We denote that the regret for the generator and discriminator as $R_T(\sigma_G)$ and $R_T(\sigma_D)$, respectively. We will prove later that QMMW possesses the no-regret property with $R_T(\sigma_G) = \mathcal{O}(T)$ and $R_T(\sigma_D) = \mathcal{O}(T)$, which implies that the optimal result can be efficiently located. Mathematically, the minimized regret for the generated state σ_G during T training rounds is defined as

$$R_T(\sigma_G) = -\sum_{t=1}^{T} \mathcal{L}(\sigma_G^{(t)}, \sigma_D^{(t)}) + \min_{\sigma_G} \sum_{t=1}^{T} \mathcal{L}(\sigma_G, \sigma_D^{(t)}) \ .$$

Similarly, we can define the regret for the discriminator as

$$R_T(\sigma_D) = \sum_{t=1}^{T} \mathcal{L}(\sigma_G^{(t)}, \sigma_D^{(t)}) - \min_{\sigma_D} \sum_{t=1}^{T} \mathcal{L}(\sigma_G^{(t)}, \sigma_D) \ .$$

We now explain QMMW. QMMW consists of three steps. First, given a targeted state ρ represented by N qubits, we set the total number of training rounds as T and let the tolerable error be $\epsilon = \sqrt{N/T}$ with $\epsilon \leq 1/2$. We also initialize the discriminator $\sigma_D^{(1)}$ as the maximally mixed state $\sigma_D^{(1)} = \mathbb{I}/2^N$. Second, we iteratively update the generator and the discriminator T training rounds. The update rule for the generated state $\sigma^{(t)}$ at t-th round is

$$\sigma_G^{(t)} = \frac{\exp^{\sum_{\tau=1}^{t}(-\epsilon\sigma_D^{(\tau)})}}{\mathrm{Tr}\left(\exp^{\sum_{\tau=1}^{t}(-\epsilon\sigma_D^{(\tau)})}\right)} \ . \tag{3}$$

The update rule for the discriminator is

$$\sigma_D^{(t+1)} = \frac{e^{\sum_{\tau=1}^{t} -\epsilon\left(\rho-\sigma_G^{(\tau)}\right)}}{\mathrm{Tr}\, e^{\sum_{\tau=1}^{t} -\epsilon\left(\rho-\sigma_G^{(\tau)}\right)}} \ . \tag{4}$$

Third, we calculate the loss $\mathcal{L}(\bar{\sigma}_D, \bar{\sigma}_D)$ defined in Eqn. (2) with $\bar{\sigma}_G = \sum_{t=1}^{T} \sigma_G^{(t)}/T$ and the averaged discriminator $\bar{\sigma}_D = \sum_{t=1}^{T} \sigma_D^{(t)}/T$ during T training rounds. This convergence rate of QMMW is assured by the following theorem:

Theorem 1. *Given a mixed state ρ represented by N qubits, and setting the training rounds as T, QMMW yields*

$$|\mathcal{L}(\bar{\sigma}_G, \bar{\sigma}_D) - \mathcal{L}(\sigma_G^*, \sigma_D^*)| \leq 3\sqrt{\frac{N}{T}}. \tag{5}$$

---

[1] As discussed in Section 3, the mixed state will be purified in the implementation of QMMW, where the physical meaning of the loss will be clearer.

The proof of Theorem 1 is given in the supplementary material SM (A).

QMMW can be efficiently executed on fault-tolerant quantum circuits, since both $\sigma_G^{(t)}$ and $\sigma_D^{(t)}$ are Gibbs states that can be prepared by using efficient Gibbs sampling methods [7], [41]. We elaborate how to carry out the proposed QMMW algorithm in the supplementary material SM(B). The efficiency of the Gibbs sampling methods proposed in [41] presents another attractive advantage of QMMW:

Theorem 2. *Given an N-qubit state, let U_ρ be the unitary that prepares the purification state of ρ. Denote T as the total number of training rounds. If there is quantum query access to U_ρ, the computation cost of the QMMW algorithm is $\mathcal{O}(N^3 T^4)$.*

The proof of Theorem 2 is given in the supplementary material SM (C). We remark that allowing access the Gibbs sampler U_ρ has also been used in [7].

IV. QuGANs with Multiplicative Weight Training Method

The investigation of applying QuGANs to tackle quantum information processing problems is of practical interest in the near term when there are only limited available qubits and shallow quantum circuit depth [36]. Although several studies have confirmed the feasibility of using QuGANs to achieve certain tasks, the variational optimization method collapses the desired convex-concave property and heavily challenges the performance of QuGANs. The disappearance of the convex-concave property results in an inevitable difficulty, since the optimization may get stuck in local minima. This topic has been widely investigated in classical GANs [46]. Inspired by the weighted training algorithm proposed by [35], which has demonstrated its effectiveness in classical GANs, we propose the multiplicative weight training method [35] to relieve the training difficulty in QuGANs. The proposed training method can be seamlessly embedded into advanced optimization algorithms used to train parameterized quantum circuits (PQCs).

Before illustrating how the multiplicative weight training method works, we first set up the QuGAN used in this paper. The generator U_G and discriminator U_D of our QuGAN are two trainable unitaries that are implemented by PQCs. Mathematically, the trainable unitary U_G and U_D are defined as

$$U_G = \prod_{i=1}^{L_1} U(\boldsymbol{\theta_i}), \; U_D = \prod_{i=1}^{L_2} U(\boldsymbol{\gamma_i}) \; , \tag{6}$$

where L_1 (L_2) refers to the number of blocks in G (D) and each block $U(\boldsymbol{\theta_i})$ ($U(\boldsymbol{\gamma_i})$) has an identical arrangement of quantum gates. Suppose that the target state ρ is represented by N qubits, the generated

state is formulated as $|\phi\rangle = U_G |0\rangle^{\otimes N'}$ with $N' = N + N_a$ and N_a being the number of ancillary qubits [2]. The generated mixed state σ_G can be obtained by partial tracing the ancillary system, i.e.,

$$\sigma_G = \text{Tr}_a(U_G(|0\rangle\langle 0|)^{\otimes N'}U_G^\dagger), \tag{7}$$

supported by Stinespring's dilation theorem [33]. The discriminator of our QuGANs is defined as

$$M_D = U_D^\dagger(\mathbb{I} \otimes E_F)U_D , \tag{8}$$

where a two-outcome positive-operator valued measurement $\mathcal{D}$ defined in Eqn. (1) is reformulated as U_D followed by a partial measurement $E_F = |0\rangle\langle 0|$ on an ancillary qubit. Following the loss function of QuGAL defined in Eqn. (1), the loss function of QuGAN yields

$$\mathcal{L}(U_G, U_D) = \text{Tr}((M_D(\rho \otimes |0\rangle\langle 0|))P(R) + \text{Tr}((\mathbb{I} - M_D)(\sigma_G \otimes |0\rangle\langle 0|))P(G) , \tag{9}$$

where σ_G is defined in Eqn. (7) and M_D is defined in Eqn. (8). The loss function of QuGAN gives the following theorem:

Lemma 3. *The loss function $\mathcal{L}(U_G, U_D)$ defined in Eqn. (9) has the convex-concave property with the equilibrium value $\mathcal{L}(U_G^*, U_D^*) = 1/2$.*

The proof of Theorem 3 is given in the supplemental material SM (D).

We now illustrate how to use the multiplicative weight training method to facilitate the optimization of QuGAN. Intuitively, this method aims to put more weight on generated states that are more likely to fool the discriminator in updating U_G. We summarize the multiplicative weight training method in Algorithm 1.

The four hyper-parameters of the multiplicative weight training method are the total number of training rounds T, the total number of inner iterations K, the learning rate $\alpha \in \mathbb{R}$, and the scale parameter $\eta \in (0, 1)$. At each training round t with $t \in [T]$, we introduce K inner iterations to obtain a better gradient for updating $\boldsymbol{\theta}^{(t+1)}$. For ease of understanding, we denote the updated parameters in K iterations as $\tilde{\boldsymbol{\theta}}$ and $\tilde{\boldsymbol{\gamma}}$. As indicated in Lines 5-7 of Algorithm 1, we iteratively update $\tilde{\boldsymbol{\theta}}$ and $\tilde{\boldsymbol{\gamma}}$, and record a set of training losses $\{\mathcal{L}(U_G^{(k)}, U_D^{(k)})\}_{k=1}^K$ and a set of gradients $\{\nabla_{\boldsymbol{\theta}}\mathcal{L}(U_G^{(k)}, U_D^{(k)})\}_{k=1}^K$. After conducting the inner iterations, we calculate the multiplicative weights and employ them to update $\boldsymbol{\theta}$, as indicated by Lines 9-10 in Algorithm 1. We note that the multiplicative weight training method differs from the weighted training algorithm proposed in [35]. The major difference is in the mechanism of QuGAN and classical GANs, i.e., classical GANs support nonlinear mapping, whereas QuGAN can only conduct linear mapping (see more details about classical GANs in the supplementary material SM (E)).

---

[2]If the given state is a pure state, we have $N_a = 0$. The value of N_a is no larger than N.

Algorithm 1 The Multiplicative Weight Training Method

    Input: T; K; $\alpha \in \mathbb{R}$; $\eta \in (0, 1)$.

    Output: The trainable parameters $\boldsymbol{\theta}^{(T)}$ and $\boldsymbol{\gamma}^{(T)}$.

1: Initialize trainable parameters $\boldsymbol{\theta}^{(0)}$, $\boldsymbol{\gamma}^{(0)}$;   ▷ Randomly sampled from uniform distribution.

2: **for** $t = 1;\ t \leq T;\ t \leftarrow t + 1$ **do**

3:      $\tilde{\boldsymbol{\theta}}^{(1)} \leftarrow \boldsymbol{\theta}^{(t)}$ and $\tilde{\boldsymbol{\gamma}}^{(1)} \leftarrow \boldsymbol{\gamma}^{(t)}$ ▷ Initialize $\tilde{\boldsymbol{\theta}}^{(k)}$ and $\tilde{\boldsymbol{\gamma}}^{(k)}$ for $k = 1$

4:      **for** $k = 1;\ k \leq K;\ k \leftarrow k + 1$ **do**

5:          $\{\mathcal{L}(U_G^{(k)}, U_D^{(k)}), \nabla_{\tilde{\boldsymbol{\theta}}} \mathcal{L}(U_G^{(k)}, U_D^{(k)})\}$ ▷ Record the training loss and gradients

6:          $\tilde{\boldsymbol{\theta}}^{(k+1)} \leftarrow \tilde{\boldsymbol{\theta}}^{(k)} + \alpha \nabla_{\tilde{\boldsymbol{\theta}}} \mathcal{L}(U_G^{(k)}, U_D^{(k)})$ ▷ Update the virtual parameters $\tilde{\boldsymbol{\theta}}$

7:          $\tilde{\boldsymbol{\gamma}}^{(k+1)} \leftarrow \tilde{\boldsymbol{\gamma}}^{(k)} - \alpha \nabla_{\tilde{\boldsymbol{\gamma}}} \mathcal{L}(U_G^{(k)}, U_D^{(k)})$ ▷ Update the virtual parameters $\tilde{\boldsymbol{\gamma}}$

8:      **end for**

9:      $w_k = \eta \frac{\mathcal{L}(U_G^{(k)}, U_D^{(k)})}{\sum_{k=1}^{K} \mathcal{L}(U_G^{(k)}, U_D^{(k)})}$ ▷ Calculate the multiplicative weights $\{w_k\}_{k=1}^{K}$

10:      $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} + \alpha \sum_{k=1}^{K} w_k \nabla_{\boldsymbol{\theta}} \mathcal{L}(U_G^{(k)}, U_D^{(k)})$ ▷ Update the trainable parameters for U_G

11:      $\boldsymbol{\gamma}^{(t+1)} \leftarrow \boldsymbol{\gamma}^{(t)} - \alpha \nabla_{\boldsymbol{\gamma}} \mathcal{L}(U_G^{(t)}, U_D^{(t)})$ ▷ Update the trainable parameters for U_D

12: **end for**

V. THE APPLICATION OF QMMW AND QUGANS FOR ENTANGLEMENT TEST

Following the observation in Section II, a core ingredient of employing QuGAL to tackle a given quantum information processing problem is to conditionally restrict the expressive power of the generator or discriminator. The restriction method is varied for different settings and implementations of QuGAL. In this section, we discuss how to conditionally restrict the expressive power of the generator or discriminator for QMMW and QuGAN can be conditionally restricted to tackle a given quantum information processing problem.

For QMMW, an extra 'constraint' step should be involved in the update rule to restrict the expressive power. Naive QMMW is capable of approximating any quantum state without the imposition of any constraint, as proved in Theorem 1. The 'constraint' step ensures that only the desired answer formulated in Section II can be efficiently approximated by QMMW. Two standard rules govern the design of the 'constraint' step, namely, that it does not destroy the no-regret property of QMMW and that it can be efficiently implemented by quantum operations.

For QuGAN, the restriction of the expressive power can be achieved by adjusting the quantum circuit structure, so that only the desired answer formulated in Section II can be efficiently simulated. In particular, the arrangement of the quantum gates of each block $U(\boldsymbol{\theta_i})$ and $U(\boldsymbol{\gamma_i})$ defined in Eqn. (6) should be

redesigned. Although QuGAN cannot guarantee an effective convergence rate as QMMW does, it may still have quantum advantages, since QuGAN does not demand expensive measurements and can be efficiently implemented on near-term quantum devices.

To facilitate understanding, we show how to use QMMW and QuGAN can be used to accomplish the entanglement test for a bipartite pure state. The formal definition of the separable bipartite pure state as follows [25]. Suppose that a given bipartite pure state $|\Psi\rangle_{AB}$ is represented by $N_A + N_B$ qubits, we say that $|\Psi\rangle_{AB} \in \mathcal{H}_A \otimes \mathcal{H}_B$ is separable if it can be written as $|\Psi\rangle_{AB} = |\phi\rangle_A |\phi\rangle_B$ with $|\phi\rangle_A \in \mathcal{H}_A$ $(|\phi\rangle_B \in \mathcal{H}_B)$.

When QMMW is employed to distinguish entanglement from a bipartite pure state, we impose an 'constraint' step in updating the generated state. We define the target state as $\rho_{AB} = |\Psi\rangle\langle\Psi|_{AB}$ at each training rounds, and two copies of σ_G are generated as defined in Eqn. (3). The 'constrained' step refers to a partial trace step, i.e., by partial trace system A for the first copy and system B for the second copy, we have the product state $\mathrm{Tr}_A(\sigma_G) \otimes \mathrm{Tr}_B(\sigma_G)$. The integration of the 'constraint' step and naive QMMW naturally results in Nash equilibrium being reached if and only if the input state is separable, since the generated state must be separable. Meanwhile, the 'constraint' step satisfies the two standards rules. It is easy to prove that the no-regret property of the varied QMMW is conserved. The partial trace can be executed with $\mathcal{O}(1)$ complexity.

When QuGAN is employed to distinguish entanglement from a bipartite pure state, we redesign the arrangement of quantum gates in each block of U_G defined in Eqn. (6). No CNOT gate exists whose controlled qubit is in system A and whose target qubit is in system B. The detailed quantum circuit architecture is shown in the right panel of Figure 1. The modified quantum circuit structure indicates that Nash equilibrium can be reached if and only if the input state is separable, since U_G can only generate a separable state.
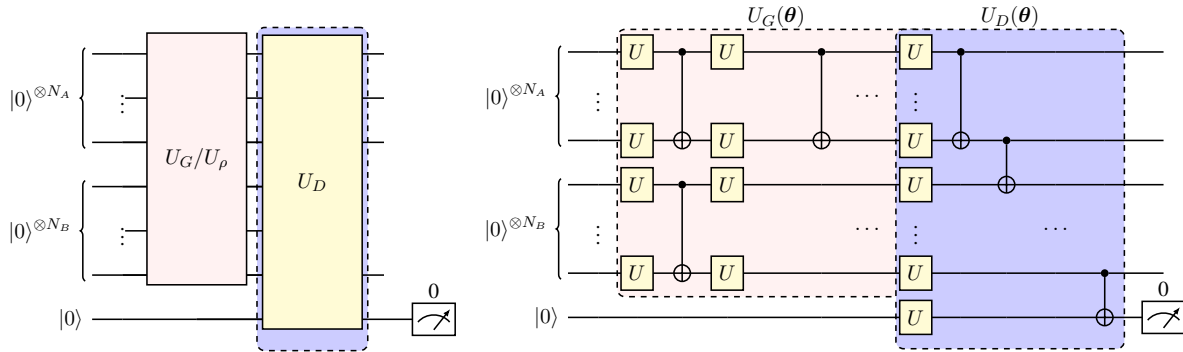
Fig. 1: The quantum circuit of QuGAN to accomplish the entanglement test for bipartite pure states. In the left panel, U_ρ (or U_G) is selected to to produce the real (or generated) state with prior $P(R)$ (or $P(G)$). In the right panel, the circuit architecture of U_G and U_D is expanded. The notation U is defined as $U = R_X \circ R_Y \circ R_Z$, where R_X, R_Y, and R_Z are trainable parameterized single qubit gates along X, Y, Z axis.

It is valuable to compare QuGAN with another advanced method, the self-testing method [22], which can also accomplish the pure entanglement test task. The core ingredient of self-testing is the controlled-SWAP test; however, the controlled-swap test has several disadvantages under near-term devices [4]. To perform the self-testing method, $2N + 1$ qubits are required, two copies should be accessed simultaneously, and the ability to conduct nontrivial controlled gates and error correction is required. In contrast to the self-testing method, QuGAN can flexibly select the number of controlled gates, which is more suitable for near-term quantum devices.

VI. NUMERICAL SIMULATIONS

QMMW is a powerful tool for approximating a given state. We validate its performance by approximating a separable mixed state $\rho_{sep} = \frac{1}{2} |0000\rangle \langle 0000| + \frac{1}{2} |1111\rangle \langle 1111|$. The total number of training rounds is set as $T = 400$ and $T = 1600$, respectively. As illustrated in Figure 2, the final training loss for $T = 400$ is 0.561 with fidelity of 0.929. The final training loss for $T = 1600$ is 0.532 with fidelity of 0.965. The simulation results indicate that the training loss rapidly converges to the equilibrium value and the fidelity between the generated state and ρ_{sep} tends to be 1 with increased T. The simulation results are in accordance with the conclusion of Theorem 1, where the theoretical results are $1/2 + 3\sqrt{4/400} = 0.7$ and $1/2 + 3\sqrt{4/1600} = 0.65$, respectively. The numerical simulations are implemented in Python in conjunction with QuTiP [27].

We then benchmark the performance of the QuGANs to accomplish the entanglement test for bipartite pure states. The detailed procedure for constructing QuGAN is as follows. The trainable parameters $\boldsymbol{\theta}$ (for U_G) and $\boldsymbol{\gamma}$ (for U_D) are randomly initialized and updated by the zero-order differential method

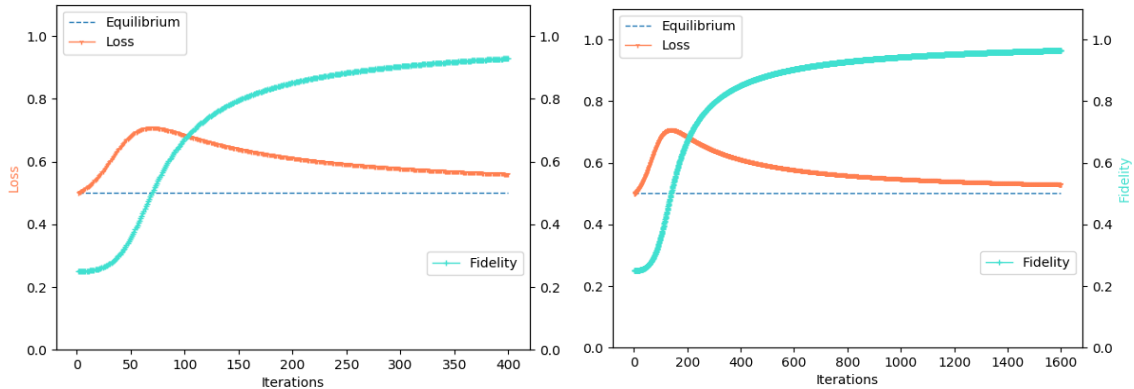## The Training Loss and Fidelity with ρ_{sep}

Fig. 2: The left panel is the simulation result of QMMW with setting $T = 400$. The right panel is the simulation result of QMMW with setting $T = 1600$.

[32]. We set the total number of training rounds T as 500. The prior defined in Eqn. (9) is set as $P(G) = P(R) = 1/2$. The detailed quantum circuit structure is illustrated in Figure 1. The number of blocks required to implement U_G and U_D as defined in Eqn. (6) is set as $L_1 = 7$ and $L_2 = 3$, respectively. The expressive power of U_G is constrained as explained in Section V. The quantum circuit architecture is demonstrated in Figure 1. All numerical simulations are implemented in Python in conjunction with the PyQuil library [40].

We now employ QuGAN to accomplish the entanglement test for two bipartite pure states, i.e., a separable state $|\Psi\rangle = (|00\rangle_A + |10\rangle_A) \otimes |00\rangle_B / \sqrt{2}$ and an entangled state $|\text{GHZ}\rangle = (|00\rangle_A \otimes |00\rangle_B + |11\rangle_A \otimes |11\rangle_B)/\sqrt{2}$, where A and B refer to the bipartite system. When the input state is separable state $|\Psi\rangle$, the training loss oscillates around the optimal value after around 100 steps and ranges from 0.444 to 0.559, as shown in the outer plot. The corresponding fidelity between the target state and the generated state is always larger than 0.702. The training loss for the entanglement state case is far away from the optimal value, which oscillates around 0.850 after 300 steps, as shown in the inner plot. The fidelity between the generated state and the given state $|\text{GHZ}\rangle$ is always below 0.250. The simulation results echo the analysis in Section V. The simulation results are illustrated in Figure 3. To accomplish the simulation, QuGAN requires 143 single and two qubit quantum gates, while self-testing method requires 240 quantum gates.
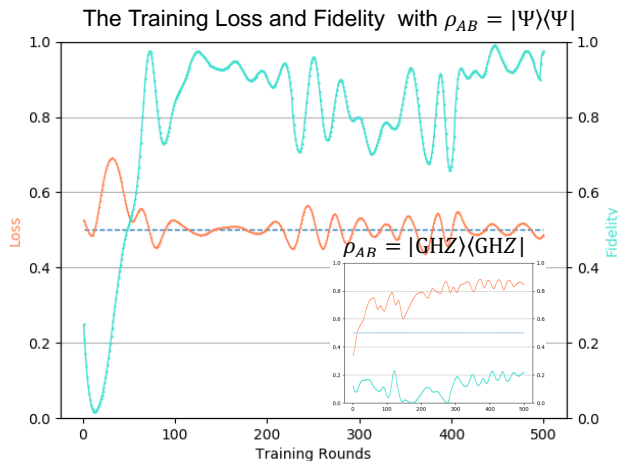
Fig. 3: The outer plot is the simulation result of QuGAN when the input is $|\Psi\rangle$. The inner plot is the simulation result of QuGAN when the input is $|GHZ\rangle$.

VII. CONCLUSION

In this paper, we have presented the first attempt to approach quantum information processing problems by employing QuGAL. We have proposed a general framework that enables quantum information processing problems to be tackled by using QuGAL. A major advantage of QuGAL is its capability to process quantum data directly, where the required number of quantum measurements is irrelevant to the size of quantum state. This advantage is significant in accomplishing quantum information process tasks, since conventional methods generally demand exponential measurements to extract sufficient quantum information into a classical form.

Encouraged by the similarity between QuGAL, online learning, and zero-sum game, we have exploited advanced online learning methods to conquer two issues in QuGAL, i.e., finding a quantum generative adversarial learning algorithm that can rapidly converge to Nash equilibrium, and how the performance in training QuGANs QuGAN can be improved. To resolve the former issue, we proposed QMMW and proved that its training loss can effectively converge to Nash equilibrium with the increased number of training rounds. The computational complexity of QMMW is polynomially proportional to the number of qubits and training rounds. To solve the latter issue, we introduced the multiplicative weight training method. The proposed method has the ability to relieve the dilemma encountered in training QuGANs such that the optimization may get stuck in local minima.

Lastly, we have described how to apply QMMW and QuGANs to solve quantum information processing tasks. We have shown that QMMW and QuGANs can be employed to accomplish entanglement test task

for pure states. Several numerical simulations were conducted to validate that QuGANs is capable of accomplishing the entanglement test with modest quantum resources.

Our future work has two key directions. First, we will focus on applying QMMW and QuGANs to tackle more fundamental quantum information problems, e.g., the identification of quantum correlation. Second, we will investigate whether other advanced online learning methods exist that may improve the training performance of QuGAN. We believe that combining QuGAL with quantum information processing will benefit the fields of quantum machine learning and quantum information.

REFERENCES

[1] Scott Aaronson, Xinyi Chen, Elad Hazan, Satyen Kale, and Ashwin Nayak. Online learning of quantum states. In *Advances in Neural Information Processing Systems*, pages 8962–8972, 2018.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 2017.

[3] Stephen M Barnett and Sarah Croke. Quantum state discrimination. *Advances in Optics and Photonics*, 1(2):238–278, 2009.

[4] Marcello Benedetti, Edward Grant, Leonard Wossnig, and Simone Severini. Adversarial quantum circuit learning for pure state approximation. *arXiv preprint arXiv:1806.00463*, 2018.

[5] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195, 2017.

[6] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[7] Fernando G. S. L. Brandão, Amir Kalev, Tongyang Li, Cedric Yen-Yu Lin, Krysta M. Svore, and Xiaodi Wu. Quantum SDP solvers: Large speed-ups, optimality, and applications to Quantum Learning. *arXiv e-prints*, page arXiv:1710.02581, October 2017.

[8] Harry Buhrman, Richard Cleve, John Watrous, and Ronald De Wolf. Quantum fingerprinting. *Physical Review Letters*, 87(16):167902, 2001.

[9] Juan Carrasquilla and Roger G Melko. Machine learning phases of matter. *Nature Physics*, 13(5):431, 2017.

[10] Eric Chitambar, Runyao Duan, and Min-Hsiu Hsieh. When do local operations and classical communication suffice for two-qubit state discrimination? *IEEE Transactions on Information Theory*, 60(3):1549–1561, 2014.

[11] Eric Chitambar and Min-Hsiu Hsieh. Revisiting the optimal detection of quantum information. *Physical Review A*, 88(2):020302, 2013.

[12] Eric Chitambar and Min-Hsiu Hsieh. Asymptotic state discrimination and a strict hierarchy in distinguishability norms. *Journal of Mathematical Physics*, 55(11):112204, 2014.

[13] Eric Chitambar and Min-Hsiu Hsieh. Round complexity in the local transformations of quantum and classical states. *Nature Communications*, 8(1):2086, 2017.

[14] Pierre-Luc Dallaire-Demers and Nathan Killoran. Quantum generative adversarial networks. *arXiv preprint arXiv:1804.08641*, 2018.

[15] Andrew C Doherty, Pablo A Parrilo, and Federico M Spedalieri. Complete family of separability criteria. *Physical Review A*, 69(2):022308, 2004.

[16] Yuxuan Du, Min-Hsiu Hsieh, Tongliang Liu, and Dacheng Tao. The expressive power of parameterized quantum circuits. *arXiv preprint arXiv:1810.11922*, 2018.

[17] Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Regret minimization in behaviorally-constrained zero-sum games. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1107–1116. JMLR. org, 2017.

[18] Ian Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[20] Paulina Grnarova, Kfir Y Levy, Aurelien Lucchi, Thomas Hofmann, and Andreas Krause. An online learning approach to generative adversarial networks. In *International Conference on Learning Representations*, 2018.

[21] Gus Gutoski and Xiaodi Wu. Parallel approximation of min-max problems. *Computational Complexity*, 22(2):385–428, 2013.

[22] Aram W Harrow and Ashley Montanaro. Testing product states, quantum Merlin-Arthur games and tensor optimization. *Journal of the ACM (JACM)*, 60(1):3, 2013.

[23] Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[24] Pawel Horodecki. Separability criterion and inseparable mixed states with positive partial transposition. *arXiv preprint quant-ph/9703004*, 1997.

[25] Ryszard Horodecki, Paweł Horodecki, Michał Horodecki, and Karol Horodecki. Quantum entanglement. *Reviews of Modern Physics*, 81(2):865, 2009.

[26] Rahul Jain and John Watrous. Parallel approximation of non-interactive zero-sum quantum games. In *2009 24th Annual IEEE Conference on Computational Complexity*, pages 243–253. IEEE, 2009.

[27] J Robert Johansson, Paul D Nation, and Franco Nori. Qutip: An open-source python framework for the dynamics of open quantum systems. *Computer Physics Communications*, 183(8):1760–1772, 2012.

[28] S. Kale. *Efficient algorithms using the multiplicative weights update method*. Princeton University, 2007.

[29] Seth Lloyd and Christian Weedbrook. Quantum generative adversarial learning. *arXiv preprint arXiv:1804.09139*, 2018.

[30] Sirui Lu, Shilin Huang, Keren Li, Jun Li, Jianxin Chen, Dawei Lu, Zhengfeng Ji, Yi Shen, Duanlu Zhou, and Bei Zeng. Separability-entanglement classifier via machine learning. *Physical Review A*, 98(1), 2018.

[31] Yue-Chi Ma and Man-Hong Yung. Transforming Bell's inequalities into state classifiers with machine learning. *npj Quantum Information*, 4(1):34, 2018.

[32] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum circuit learning. *arXiv preprint arXiv:1803.00745*, 2018.

[33] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2010.

[34] Martin J Osborne and Ariel Rubinstein. *A course in game theory*. MIT Press, 1994.

[35] Yannis Pantazis, Dipjyoti Paul, Michail Fasoulakis, and Yannis Stylianou. Training generative adversarial networks with weights. *arXiv preprint arXiv:1811.02598*, 2018.

[36] John Preskill. Quantum computing in the NISQ era and beyond. *arXiv preprint arXiv:1801.00862*, 2018.

[37] Jonathan Romero and Alan Aspuru-Guzik. Variational quantum generators: Generative adversarial quantum machine learning for continuous distributions. *arXiv preprint arXiv:1901.00848*, 2019.

[38] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):303–332, 1999.

[39] Haozhen Situ, Zhimin He, Lvzhou Li, and Shenggen Zheng. Adversarial training of quantum born machine. *arXiv preprint arXiv:1807.01235*, 2018.

[40] Robert S Smith, Michael J Curtis, and William J Zeng. A practical quantum instruction set architecture. *arXiv preprint arXiv:1608.03355*, 2016.

[41] Joran van Apeldoorn and András Gilyén. Improvements in quantum SDP-solving with applications. *arXiv preprint arXiv:1804.05058*, 2018.

[42] Joran van Apeldoorn and András Gilyén. Quantum algorithms for zero-sum games. *arXiv preprint arXiv:1904.03180*, 2019.

[43] Evert PL Van Nieuwenburg, Ye-Hua Liu, and Sebastian D Huber. Learning phase transitions by confusion. *Nature Physics*, 13(5):435, 2017.

[44] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[45] Jinfeng Zeng, Yufeng Wu, Jin-Guo Liu, Lei Wang, and Jiangping Hu. Learning and inference on generative adversarial quantum circuits. *arXiv preprint arXiv:1808.03425*, 2018.

[46] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[47] Christa Zoufal, Aurélien Lucchi, and Stefan Woerner. Quantum generative adversarial networks for learning and loading random distributions. *arXiv preprint arXiv:1904.00043*, 2019.

APPENDIX

A. SM(A) Proof of Theorem 1

The analysis of the convergence of quantum multiplicative matrix weight (QMMW) relies mainly on conclusions drawn from game theory and online learning. To provide an illustrative proof, we first introduce the necessary concepts from these two fields and then build the connection to Theorem 1.

The formal definition of the zero-sum game is:

Definition 1 (Zero-sum Game, [17]). *A two-player zero-sum game is a tuple $(\mathcal{X}, \mathcal{Y}, u)$ where $\mathcal{X}$ represents the finite set of actions that player 1 can play, $\mathcal{Y}$ represents the finite set of actions that player 2 can play, and $u : \mathcal{X} \times \mathcal{Y} \to R$ is the payoff function for player 1, mapping the pair of actions $(x, y) \in (\mathcal{X}, \mathcal{Y})$ of the players into the payoff for player 1. The corresponding payoff for player 2 is given by $-u(x, y)$.*

Exploiting the definition of the zero-sum game, we introduce two concepts in game theory. The first concept is *approximated best response*, defined as follows:

Definition 2 (Approximate best response). *Given a zero-sum game $(\mathcal{X}, \mathcal{Y}, u)$, we say that $x \in \mathcal{X}$ is an ε-best response to y for player 1 if $u(x, y) + \varepsilon \geq u(\hat{x}, y)$ for all $\hat{x} \in \mathcal{X}$. Symmetrically, given $x \in \mathcal{X}$, we say that $y \in \mathcal{Y}$ is an ε-best response to x for player 2 if $-u(x, y) + \varepsilon \geq -u(x, \hat{y})$ for all $\hat{y} \in Y$.*

Another concept is *approximated Nash equilibrium*, defined as:

Definition 3 (Approximate Nash equilibrium). *Given a zero-sum game $(\mathcal{X}, \mathcal{Y}, u)$, the strategy pair $(x, y) \in (\mathcal{X} \times \mathcal{Y})$ is an ε-Nash equilibrium for the game if x is an ε-best response to y for player 1, and y is an ε-best response to x for player 2. Note that Nash equilibrium can be treated as a 0-Nash equilibrium.*

A well-known conclusion between regret and approximate Nash equilibria is as follows [17]:

Proposition 4. *In a zero-sum game, if the average regrets of the players up to step T are such that $R_T(\hat{x}) := \sum_{t=1}^{T}(u(x_t, y_t) - u(\hat{x}, y_t))$ with $R_T(\hat{x})/T \leq \varepsilon_1$, $R_T(\hat{y}) := \sum_{t=1}^{T}(-u(x_t, y_t) + u(x_t, \hat{y}))$ with $R_T(\hat{y})/T \leq \varepsilon_2$, for all actions $\hat{x} \in \mathcal{X}$, $\hat{y} \in \mathcal{Y}$, then the strategy pair $(\bar{x}_T, \bar{y}_T) := (\sum_{t=1}^{T} x_t/T, \sum_{t=1}^{T} y_t/T) \in (\mathcal{X}, \mathcal{Y})$ is a $(\varepsilon_1 + \varepsilon_2)$-Nash equilibrium.*

We now connect the Proposition 4 with Theorem 1. In QMMW, the minimized regret for the generator is

$$R_T(\sigma_G) = -\sum_{t=1}^{T} \mathcal{L}(\sigma_G^{(t)}, \sigma_D^{(t)}) + \min_{\sigma_G} \sum_{t=1}^{T} \mathcal{L}(\sigma_G, \sigma_D^{(t)}) . \tag{10}$$

The minimized regret for the discriminator is

$$R_T(\sigma_D) = \sum_{t=1}^{T} \mathcal{L}(\sigma_G^{(t)}, \sigma_D^{(t)}) - \min_{\sigma_D} \sum_{t=1}^{T} \mathcal{L}(\sigma_G^{(t)}, \sigma_D) \ . \tag{11}$$

Suppose that the optimal strategy pair is (σ_G^*, σ_D^*) and the corresponding Nash equilibrium is $\mathcal{L}(\sigma_G^*, \sigma_D^*) = 1/2$. Following the statement of Proposition 4 and Definition 2, with setting ε_G and ε_G that satisfies $R_T(\sigma_G)/T \leq \varepsilon_G$ and $R_T(\sigma_D)/T \leq \varepsilon_D$, the strategy pair $(\bar{\sigma}_G, \bar{\sigma}_D)$ with $\bar{\sigma} = \sum_{t=1}^{T} \sigma^{(t)}/T$ and the $\bar{\sigma}_D = \sum_{t=1}^{T} \sigma_D^{(t)}/T$, is a $(\varepsilon_G + \varepsilon_D)$-Nash equilibrium, i.e.,

$$\mathcal{L}(\sigma^*, \bar{\sigma}_D) - (\varepsilon_G + \varepsilon_D) \leq \mathcal{L}(\bar{\sigma}_G, \bar{\sigma}_D) \leq \mathcal{L}(\bar{\sigma}_G, \sigma_D^*) + (\varepsilon_G + \varepsilon_D) \ . \tag{12}$$

These two inequalities come from Definition 2. According to the definition of Nash equilibrium, which is an 0-Nash Equilibrium with an optimal strategy pair (σ^*, σ_D^*), we rewrite Eqn. (12) as

$$\mathcal{L}(\bar{\sigma}_G, \sigma_D^*) \leq \mathcal{L}(\sigma_G^*, \sigma_D^*) \leq \mathcal{L}(\sigma_G^*, \bar{\sigma}_D) \ . \tag{13}$$

Connecting Eqn. (12) with Eqn. (13), we have

$$|\mathcal{L}(\bar{\sigma}_G, \bar{\sigma}_D) - \mathcal{L}(\sigma_G^*, \sigma_D^*))| \leq \varepsilon_G + \varepsilon_D \ . \tag{14}$$

The goal of QMMW, or Theorem 1, is to prove $\varepsilon_G + \varepsilon_D \leq 3\sqrt{N/T}$, which implies that $\mathcal{L}(\bar{\sigma}_G, \bar{\sigma}_D)$ converges to the optimal value $\mathcal{L}(\sigma_G^*, \sigma_D^*) = 1/2$ with increasing T.

Proof of Theorem 1. As discussed above, we aim to prove that QMMW possesses the no-regret property when the original state ρ is separable.

We employ the following two claims to quantify two regrets $R_T(\sigma_G)$ and $R_T(\sigma_D)$ (Proof of Claim 5 and Claim 6 provided later in this section.):

Claim 5. *The regret of $R_T(\sigma_G)$ based on the update rule of the Quantum Matrix Multiplicative Weights algorithm (to maximize the loss) is bounded by ε_G, i.e.,*

$$\frac{R_T(\sigma_G)}{T} \leq \varepsilon_G = \frac{\epsilon^2 T + N}{2\epsilon T} \ . \tag{15}$$

Claim 6. *The regret of $R_T(\sigma_D)$ based on the update rule of the Quantum Matrix Multiplicative Weights algorithm (to minimize the loss) is bounded by ε_D, i.e.,*

$$\frac{R_T(\sigma_D)}{T} \leq \varepsilon_D = \frac{\epsilon^2 T + (N+1)}{2\epsilon T} \ . \tag{16}$$

Combining the above two claims and with setting $\epsilon = 2\sqrt{N/T}$, we have

$$\varepsilon_G + \varepsilon_D \leq \frac{2\epsilon^2 T + 2(N+1)}{2\epsilon T} \leq 3\sqrt{\frac{N}{T}} \ , \tag{17}$$

where the first inequality comes from the results of Claim 5 and Claim 6, and the second equality results from $N + 1 \leq 2N$ and the definition of ϵ. The discrepancy between the loss $\mathcal{L}(\bar{\sigma}_G, \bar{\sigma}_D)$ and the equilibrium value $\mathcal{L}(\sigma_G^*, \sigma_D^*)$ yields

$$|\mathcal{L}(\bar{\sigma}_G, \bar{\sigma}_D) - \mathcal{L}(\sigma_G^*, \sigma_D^*)| \leq 3\sqrt{\frac{N}{T}}. \tag{18}$$

$\square$

Before giving the proof of Claim 6 and Claim 5, we introduce the following two results to facilitate the proof.

Corollary 7 (Corollary 2, [28])**.** *For any $\epsilon \leq 1$, let $\epsilon_1 = 1 - e^{-\epsilon}$ and $\epsilon_2 = e^\epsilon - 1$. We then have the following matrix inequalities:*

- *If all eigenvalues of a symmetric matrix A lie in $[0, 1]$, then $e^{\epsilon A} \preceq \mathbb{I} - \epsilon_1 A$;*
- *If all eigenvalues of a symmetric matrix A lie in $[-1, 0]$, then $e^{\epsilon A} \preceq \mathbb{I} - \epsilon_2 A$;*

Proof of Claim 5. Following the observation of Eqn. (14) and Proposition 4, we hope the regret $R_T(\sigma_G)$ possesses the no-regret property with $R_T(\sigma_G) \leq \varepsilon_G \sim O(1/\sqrt{T})$, i.e.,

$$\sum_{t=1}^{T} \mathcal{L}(\sigma_G^{(t)}, \sigma_D^{(t)}) \geq \sum_{t=1}^{T} \mathcal{L}(\sigma_G^*, \sigma_D^{(t)}) - \varepsilon_G T . \tag{19}$$

To quantify $R_T(\sigma_G)$, we define the potential function $\Theta^{(t)}$ and track its evolution with varying t, which is analogous to the proof of the conventional MMW algorithm, i.e., $\Theta^{(t)} = \text{Tr}(W^{(t)})$ and $W^{(t)} = e^{\sum_{\tau=1}^{t-1} -\epsilon \sigma_D^{(\tau)}}$. By defining $\epsilon_1 = 1 - e^{-\epsilon}$, we have:

$$\begin{aligned}
\Theta^{(t+1)} &= \text{Tr}\left(W^{(t+1)}\right) = \text{Tr}\left(e^{\sum_{\tau=1}^{t} -\epsilon \sigma_D^{(\tau)}}\right) \\
&\leq \text{Tr}\left(e^{(\sum_{\tau=1}^{t-1} -\epsilon \sigma_D^{(\tau)})} e^{-\epsilon \sigma_D^{(t)}}\right) = \text{Tr}(W^{(t)} e^{-\epsilon \sigma_D^{(t)}}) \leq \text{Tr}(W^{(t)}(\mathbb{I} - \epsilon_1 \sigma_D^{(t)})) \\
&= \text{Tr}\left(W^{(t)}\right)(1 - \epsilon_1 \text{Tr}(\sigma_D^{(t)} \frac{W^{(t)}}{\text{Tr}\left(W^{(t)}\right)})) \leq \Theta^{(t)} e^{-\epsilon_1 \text{Tr}(\sigma_D^{(t)} \frac{W^{(t)}}{\text{Tr}(W^{(t)})})} \\
&= \Theta^{(t)} e^{-\epsilon_1 \text{Tr}\left(\sigma_D^{(t)} \sigma_G^{(t)}\right)} .
\end{aligned} \tag{20}$$

The first inequality comes from the Golden-Thompson inequality, the second inequality employs the conclusion of Corollary 7, the third is supported by $1 - x \leq e^{-x}$, and the last equality arises in $\frac{W^{(t)}}{\text{Tr}\left(W^{(t)}\right)} = \sigma_G^{(t)}$.

By induction, since $\Theta^{(1)} = \text{Tr}(e^{-\epsilon \sigma_D^{(1)}}) \leq \text{Tr}(e^0) \leq 2^N$, we have

$$\Theta^{(T+1)} \leq 2^N e^{-\epsilon_1 \sum_{t=1}^{T} \text{Tr}\left(\sigma_D^{(t)} \sigma_G^{(t)}\right)}. \tag{21}$$

The denotation $A = \sum_{t=1}^{T} (\sigma_D^{(t)})$, $\Theta^{(T+1)}$ yields

$$\Theta^{(T+1)} = \text{Tr}(\exp(-\epsilon A)) = \sum_k \exp\left(-\epsilon\lambda_k(A)\right) \geq \exp\left(-\epsilon\lambda_n(A)\right) ,$$

where $\lambda_n(A)$ refers to the minimum eigenvalue of A. We also have

$$e^{-\epsilon\sum_{t=1}^{T} \text{Tr}(\sigma_D^{(t)}\sigma_G^*)} \leq \exp\left(-\epsilon\lambda_n(A)\right) ,$$

since we always have $\lambda_n(A) \leq \text{Tr}(A\sigma_G^*)$ with $\text{Tr}(\sigma_G^*) = 1$ and then $\exp(-\epsilon\lambda_n(A)) \geq \exp(-\epsilon\text{Tr}(A\sigma_G^*))$. The lower bound of $\Theta^{(T+1)}$ is therefore

$$\Theta^{(T+1)} \geq e^{-\epsilon\sum_{t=1}^{T} \text{Tr}(\sigma_D^{(t)}\sigma_G^*)} . \tag{22}$$

By connecting Eqn. (22) with Eqn. (21), we have

$$e^{-\epsilon\sum_{t=1}^{T} \text{Tr}(\sigma_D^{(t)}\sigma_G^*)} \leq 2^N e^{-\epsilon_1 \sum_{t=1}^{T} \text{Tr}(\sigma_D^{(t)}\sigma_G^{(t)})} . \tag{23}$$

Taking the logarithms of Eqn. (23) and exploiting $\epsilon_1 \geq \epsilon(1-\epsilon)$, we obtain the following inequality,

$$(1-\epsilon)\sum_{t=1}^{T} \text{Tr}(\sigma_G^{(t)}\sigma_D^{(t)}) \leq \sum_{t=1}^{T} \text{Tr}(\sigma_G^*\sigma_D^{(t)}) + \frac{N}{\epsilon}$$

$$\Rightarrow \sum_{t=1}^{T} \text{Tr}(\sigma_G^{(t)}\sigma_D^{(t)}) \leq \sum_{t=1}^{T} \text{Tr}(\sigma_G^*\sigma_D^{(t)}) + \epsilon T + \frac{N}{\epsilon}$$

$$\Rightarrow \sum_{t=1}^{T} \text{Tr}((\rho\sigma_D^{(t)}) - \sum_{t=1}^{T} \text{Tr}(\sigma_G^{(t)}\sigma_D^{(t)}) \geq \sum_{t=1}^{T} \text{Tr}((\rho\sigma_D^{(t)}) - \sum_{t=1}^{T} \text{Tr}(\sigma_G^*\sigma_D^{(t)}) - \epsilon T - \frac{N}{\epsilon}$$

$$\Rightarrow \sum_{t=1}^{T} \mathcal{L}(\sigma_G^{(t)}, \sigma_D^{(t)}) \geq \sum_{t=1}^{T} \mathcal{L}(\sigma_G^*, \sigma_D^{(t)}) - \frac{\epsilon^2 T + N}{2\epsilon} .$$

The first arrow results from the fact that: $0 \leq \epsilon\,\text{Tr}(\sigma_G^{(t)}\sigma_D^{(t)}) \leq \epsilon$ with $\text{Tr}(\sigma_G^{(t)}\sigma_D^{(t)}) \leq \text{Tr}(\sigma_G^{(t)})\text{Tr}(\sigma_D^{(t)}) = 1$ for any $1 \leq t \leq T$. The second arrow comes from adding the term $\sum_{t=1}^{T} \text{Tr}((\rho\sigma_D^{(t)})$ on both sides. The last arrow comes from the definition of the loss function of QMMW. The above equation indicates that $R_T(\sigma_G) \leq \varepsilon_G = \frac{\epsilon^2 T + N}{2\epsilon T}$. □

Proof of Claim 6. This claim can be easily proved by imitating the proof of Claim 5. Conceptually, we hope to derive a bound of the classification error, i.e.,

$$\sum_{t=1}^{T} \mathcal{L}(\sigma_G^{(t)}, \sigma_D^{(t)}) \leq \sum_{t=1}^{T} \mathcal{L}(\sigma_G^{(t)}, \sigma_D^*) + \varepsilon_D T , \tag{24}$$

where σ_D^* refers to the optimal solution. We can then quantify the regret of $R_T(\mathcal{D}_{AB})$.

Following the proof of Claim 5, we define a potential function $\Theta^{(T)} = \text{Tr}(W^T)$ with $W^{(T)} = e^{-\epsilon\sum_{t=1}^{T-1}(\rho-\sigma_G^{(t)})}$. Let $Q^{(t)}$ be $Q^{(t)} = \rho - \sigma_G^{(t)}$ with $Q^{(t)} = Q_+^{(t)} + Q_-^{(t)}$. We decompose $Q(t)$ into two

terms, $Q_+^{(t)}$ and $Q_-^{(t)}$ with $Q(t) = Q_+^{(t)} + Q_-^{(t)}$, where $Q_+^{(t)}$ ($Q_-^{(t)}$) is formed by all non-negative (negative) eigenvalues and eigenvectors of $Q^{(t)}$. Defining that $\epsilon_1 = 1 - e^{-\epsilon}$ and $\epsilon_2 = e^\epsilon - 1$, we have:

$$
\begin{aligned}
\Theta^{(t+1)} &= \mathrm{Tr}\,(W^{(t+1)}) = \mathrm{Tr}\left(e^{\sum_{\tau=1}^{t} -\epsilon(\rho - \sigma_G^{(\tau)})}\right) \\
&\leq \mathrm{Tr}\left(e^{\sum_{\tau=1}^{t-1} -\epsilon(\rho - \sigma_G^{(\tau)})} e^{-\epsilon Q^{(t)}}\right) = \mathrm{Tr}\left(W^{(t)} e^{-\epsilon(Q_+^{(t)} + Q_-^{(t)})}\right) \\
&\leq \mathrm{Tr}\left(W^{(t)} e^{-\epsilon Q_+^{(t)}} e^{-\epsilon Q_-^{(t)}}\right) \\
&\leq \mathrm{Tr}\left(W^{(t)} (\mathbb{I} - \epsilon_1 Q_+^{(t)})(\mathbb{I} - \epsilon_2 Q_-^{(t)})\right) \\
&= \mathrm{Tr}\left(W^{(t)} (\mathbb{I} - \epsilon_1 Q_+^{(t)} - \epsilon_2 Q_-^{(t)})\right), \text{since } Q_+ \perp Q_- \\
&= \mathrm{Tr}\left(W^{(t)}\right)\left(1 - \frac{\mathrm{Tr}\left(W^{(t)}(\epsilon_1 Q_+^{(t)} + \epsilon_2 Q_-^{(t)})\right)}{\mathrm{Tr}\left(W^{(t)}\right)}\right) \\
&= \mathrm{Tr}\left(W^{(t)}\right)\left(1 - \mathrm{Tr}\left(\sigma_D^{(t)}(\epsilon_1 Q_+^{(t)} + \epsilon_2 Q_-^{(t)})\right)\right) \\
&= \mathrm{Tr}\left(W^{(t)}\right)\left(1 - \mathrm{Tr}\left(\sigma_D^{(t)}(\epsilon_1 Q^{(t)} + (\epsilon_2 - \epsilon_1) Q_-^{(t)})\right)\right) \\
&\leq \mathrm{Tr}\left(W^{(t)}\right)\left(1 - \mathrm{Tr}\left(\sigma_D^{(t)}(\epsilon_1 Q_+^{(t)} + \epsilon_1 Q_-^{(t)})\right)\right), \\
&\leq \Theta^{(t)} e^{-\epsilon_1 \mathrm{Tr}\,(\sigma_D^{(t)}(\rho - \sigma_G^{(t)}))},
\end{aligned}
\tag{25}
$$

where the first inequality and the second inequality come from the Golden-Thompson inequality, the third inequality employs the conclusion of Corollary 7, the penultimate inequality is supported by $-\mathrm{Tr}(\sigma_D^{(t)}(\epsilon_2 - \epsilon_1)Q_-^{(t)}) \geq 0$ (To be proved later), and the last inequality is supported by $1 - x \leq e^{-x}$. We now prove that the last second inequality always satisfies $-\mathrm{Tr}(\sigma_D^{(t)}(\epsilon_1 - \epsilon_2)Q_-^{(t)}) \geq 0$. Since $\sigma_D^{(t)} \succeq 0$, it is equivalent to prove $(\epsilon_1 - \epsilon_2)Q_-^{(t)} \preceq 0$. Due to $Q_-^{(t)} \preceq 0$, $(\epsilon_1 - \epsilon_2)Q_-^{(t)} \preceq 0$ is reduced to prove $\epsilon_2 - \epsilon_1 \geq 0$, which is always succeed since $e^\epsilon + e^{-\epsilon} > 2$.

By induction, we have

$$
\Theta^{(T+1)} \leq \Theta^{(1)} e^{\sum_{t=1}^{T} \mathrm{Tr}(\sigma_D^{(t)}(\rho - \sigma_G^{(t)}))}.
\tag{26}
$$

Due to $\Theta^{(1)} = \mathrm{Tr}(e^{-\epsilon(\rho - \sigma_G^{(1)})}) \leq \mathrm{Tr}(e^{\mathbb{I}}) \leq e\,\mathrm{Tr}(\mathbb{I}_{AB}) \leq e^{N+1}$, the above inequality can be reformulated as

$$
\Theta^{(T+1)} \leq e^{N+1} e^{\sum_{t=1}^{T} \mathrm{Tr}(\sigma_D^{(t)}(\rho - \sigma_G^{(t)}))}.
\tag{27}
$$

Concurrently, we have

$$
\Theta^{(T+1)} \geq e^{\sum_{t=1}^{T} \mathrm{Tr}(\sigma_D^*(\rho - \sigma_G^{(t)}))},
\tag{28}
$$

where the proof is analogous to Eqn. (22). We specifically set $A = \sum_{t=1}^{T}(\rho - \sigma_G^{(t)})$. The left term $\Theta^{(T)}$ of Eqn. (28) follows $\Theta^{(T)} = \mathrm{Tr}(\exp(-\epsilon A)) = \sum_k \exp(-\epsilon \lambda_k(A)) \geq \exp(-\epsilon \lambda_n(A))$, where $\lambda_n(A)$

refers to the minimum absolute eigenvalue of A. The right term of Eqn. (28) follows $e^{-\epsilon \sum_{t=1}^{T} \mathrm{Tr}\,(\sigma_D^* A)} \leq \exp(-\epsilon \lambda_n(A))$, since we always have $\lambda_n(A) \leq \mathrm{Tr}(A\sigma_D^*)$ with $0 \preceq \sigma_D^* \preceq \mathbb{I}_{AB}$ and $\mathrm{Tr}\,\sigma_D^* = 1$. This leads to $\exp -\epsilon \lambda_n(A) \geq \exp -\epsilon \mathrm{Tr}(A\sigma_D^*)$. Therefore, we obtain Eqn. (28).

By connecting Eqn. (26) with Eqn. (28), we have

$$e^{-\epsilon \sum_{t=1}^{T} \mathrm{Tr}\,(\sigma_D^*(\rho - \sigma_G^{(t)}))} \leq e^{N+1} e^{-\epsilon_1 \sum_{t=1}^{T} \mathrm{Tr}\,(\sigma_D^{(t)}(\rho - \sigma_G^{(t)}))} \ . \tag{29}$$

Taking logarithms and simplifying Eqn. (29), we obtain the following inequality,

$$-\epsilon \sum_{t=1}^{T} \mathrm{Tr}\,(\sigma_D^*(\rho - \sigma_G^{(t)})) \leq (N+1) - \epsilon_1 \sum_{t=1}^{T} \mathrm{Tr}\,(\sigma_D^{(t)}(\rho - \sigma_G^{(t)}))$$

$$\Rightarrow -\epsilon \sum_{t=1}^{T} \mathrm{Tr}\,(\sigma_D^*(\rho - \sigma_G^{(t)})) \leq (N+1) - \epsilon(1-\epsilon) \sum_{t=1}^{T} \mathrm{Tr}\,(\sigma_D^{(t)}(\rho - \sigma_G^{(t)}))$$

$$\Rightarrow (1-\epsilon) \sum_{t=1}^{T} \mathrm{Tr}\,(\sigma_D^{(t)}(\rho - \sigma_G^{(t)})) \leq \sum_{t=1}^{T} \mathrm{Tr}\,(\sigma_D^*(\rho - \sigma_G^{(t)})) + \frac{N+1}{\epsilon}$$

$$\Rightarrow \sum_{t=1}^{T} \mathrm{Tr}\,(\sigma_D^{(t)}(\rho - \sigma_G^{(t)})) \leq \sum_{t=1}^{T} \mathrm{Tr}\,(\sigma_D^*(\rho - \sigma_G^{(t)})) + \epsilon T + \frac{N+1}{\epsilon}$$

$$\Rightarrow \sum_{t=1}^{T} \mathcal{L}(\sigma_G^{(t)}, \sigma_D^{(t)}) - \frac{\epsilon^2 T + (N+1)}{2\epsilon} \leq \sum_{t=1}^{T} \mathcal{L}(\sigma_G^{(t)}, \sigma_D^*) \ .$$

The first arrow results from the fact that $\epsilon_1 \geq \epsilon(1-\epsilon)$. The second arrow comes from dividing ϵ on both sides. The third arrow employs $\mathrm{Tr}\,(\sigma_D^*(\rho - \sigma_G^{(t)})) \leq 1$, since $\mathrm{Tr}\,(\sigma_D^*(\rho - \sigma_G^{(t)})) = \mathrm{Tr}\,(\sigma_D^* \rho) - \mathrm{Tr}\,(\sigma_D^* \sigma_G^{(t)})) \leq \mathrm{Tr}\,(\sigma_D^* \rho) \leq 1$. The last arrow comes from the definition of the loss function of QMMW. This inequality immediately indicates that the bound for the minimized regret $R(\sigma_D)$ is $R(\sigma_D) \leq \varepsilon_D = (\epsilon^2 T + N + 1)/(2\epsilon T)$. $\qquad\square$

B. SM (B) Instantiation of QMMW

Let us briefly review the Quantum Multiplicative Matrix Weight algorithm (QMMW). QMMW consists of three steps: First, initializing parameters; Second, updating the generated state $\sigma_G^{(t)}$ and the density operator $\sigma_D^{(t)}$ iteratively during T training rounds; Last, calculating the loss $\mathcal{L}(\sigma_G^{(t)}, \sigma_D^{(t)})$ using $\bar{\sigma}_G$ and $\bar{\sigma}_D$ with $\bar{\sigma}_G = \sum_{t=1}^{T} \sigma_G^{(t)}/\sqrt{T}$ and $\bar{\sigma}_D = \sum_{t=1}^{T} \sigma_D^{(t)}/T$. Both the generated state $\sigma_G^{(t)}$ and the discriminator $\sigma_D^{(t)}$ can be treated as Gibbs state. The formal definition of Gibbs state and Gibbs Sampler is:

Definition 4 (Gibbs Sampler [41])**.** *A θ-precise Gibbs-sampler is a unitary that creates as output a purification of a θ-approximation in trace distance of the Gibbs state $\exp\left(\sum_{t=1}^{T} -y_t C^{(t)}\right)/\mathrm{Tr}\left(\exp(\sum_{t=1}^{T} -y_t C^{(t)})\right)$, where $\{C^{(t)}\}_{t=1}^{T}$ is a set of Hermitian matrix. If $\|y\|_1 \leq K$ and the support of y has the size at most d, then we write $\mathcal{T}(K, d, 4\theta)$ for the cost of this unitary.*

We also allow Gibbs-samplers that require a random classical input seed $S \in \{0, 1\}^a$ for some $a = \mathcal{O}(\log(1/\theta))$. In this case the output should be a θ-approximation of the Gibbs state with high probability $(\geq 4/5)$ over a uniformly random input seed S.

For ease of description, we denote the responsible θ-precise Gibbs-sampler for the generated state $\sigma_G^{(t)}$ as $U_{\sigma_G^{(t)}}$. For the discriminator $\sigma_D^{(t)}$, we denote the responsible θ-precise Gibbs-sampler as $U_{\sigma_D^{(t)}}$. Observing the QMMW algorithm, the third step can be efficiently executed using the SWAP test once we have prepared $\{U_{\sigma_G^{(t)}}\}_{t=1}^T$ and $\{U_{\sigma_D^{(t)}}\}_{t=1}^T$ [8], where the query complexity is $\mathcal{O}(1)$. Consequently, preparing Gibbs samplers is the central part of the implementation of QMMW and also dominates the computational cost.

We now elaborate how to accomplish the second step of QMMW, i.e., the construction of $\{U_{\sigma_G^{(t)}}\}_{t=1}^T$ and $\{U_{\sigma_D^{(t)}}\}_{t=1}^T$. This task employs two subroutines $O_{\sigma_G^T}$ and $O_{\sigma_D^T}$, i.e., the subroutine $O_{\sigma_G^T}$ after training t-rounds is

$$\begin{cases} O_{\sigma_G^T} |\tau\rangle |0\rangle^{\otimes N'} := \sum_{\tau \leq t} \alpha_\tau |\tau\rangle U_{\sigma_G^{(\tau)}} |0\rangle^{\otimes N'} = \sum_{\tau \leq t} \alpha_\tau |\tau\rangle |\psi_G^{(\tau)}\rangle, & \tau \leq t \\ O_{\sigma_G^T} |\tau\rangle |0\rangle^{\otimes N'} := \sum_{\tau > t} \alpha_\tau |\tau\rangle |0\rangle^{\otimes N'} = \sum_{\tau > t} \alpha_\tau |\tau\rangle |0\rangle^{\otimes N'}, & \tau > t \end{cases}$$

where $N' = a + N$, a refers to the number of ancillary qubits with $a \sim O(\log N)$ [41], $|\tau\rangle$ refers to the computational basis corresponding to the τ-th training round, $U_{\sigma_G^{(\tau)}}$ prepares the purification $|\psi_G^{(\tau)}\rangle$ of the Gibbs state $\sigma_G^{(\tau)}$, and $\sum_{\tau=1}^t \alpha_\tau^2 = 1$. Note that the Gibbs sampler $U_{\sigma_G^{(\tau)}}$ can only be prepared for $\tau \leq t$. Similarly, the subroutine $O_{\sigma_D^T}$ after training t-rounds is

$$\begin{cases} O_{\sigma_D^T} |\tau\rangle |0\rangle^{\otimes N'} := \sum_{\tau \leq t} \beta_\tau |\tau\rangle U_{\sigma_D^{(\tau)}} |0\rangle^{\otimes N'} = \sum_{\tau \leq t} \gamma_\tau |\tau\rangle |\psi_D^{(\tau)}\rangle, & \tau \leq t \\ O_{\sigma_D^T} |\tau\rangle |0\rangle^{\otimes N'} := \sum_{\tau > t} \gamma_\tau |\tau\rangle |0\rangle^{\otimes N'} = \sum_{\tau > t} \gamma_\tau |\tau\rangle |0\rangle^{\otimes N'}, & \tau > t \end{cases}$$

where $|\psi_D^{(\tau)}\rangle$ refers to the purification of the Gibbs state $\sigma_G^{(\tau)}$. After training T steps with setting $\alpha_\tau = \sqrt{1/T}$ and $\gamma_\tau = \sqrt{1/T}$ for any $\tau \in [T]$, we prepare the purification of Gibbs state $\bar{\sigma}_G^{(t)}$ and $\bar{\sigma}_D^{(t)}$.

In QMMW, we employ the method proposed by [41] to prepare the Gibbs sampler, which has the following result,

Theorem 8 (Theorem 22, [41]). *Suppose that we have query access to the unitaries $U_{\rho^\pm}$ preparing a purification of the subnormal density operators [3] $\mu^\pm$, such that $H = (\mu^+ - \mu^-)/2$. Suppose that $\beta \geq 1$ and $\theta, \delta \in (0, 1]$, there is a quantum algorithm, that using $\mathcal{O}_\theta(\beta^{3.5}/\delta)$ queries to controlled-$U_{\rho^\pm}$ or their inverses, prepares a purification of a quantum state ρ_S such that $\left| \rho_S - \frac{e^{-\beta H}}{\text{Tr}(e^{-\beta H})} \right| \leq \theta$, where S is an $\mathcal{O}(\log(\beta/\delta))$-bit random seed, and the above holds for at least $(1 - \delta)$-fraction of seeds.*

---

[3]A density operator Q is said to be subnormal if it satisfies $\text{Tr} \, Q = 1$.

The update rule of QMMW accompanies with the query access to $U_{\sigma_D^{(1)}}$ (A set of Hadamard gates to prepare the maximally mixed state $\sigma_D^{(1)}$) and U_ρ enables us to use Theorem 8 to construct all Gibbs samplers. Without loss of generality, we consider the preparation of $U_{\sigma_G^{(t)}}$ and $U_{\sigma_D^{(t)}}$. Following the update rule, the density operator μ^+ and μ^- defined in Theorem 8 refers to 0 and the purification of $\sum_{\tau=1}^{t-1} \sigma_D^{(\tau)}/(t-1)$, respectively to prepare $U_{\sigma_G^{(t)}}$. Meanwhile, we have $\beta = 2\epsilon(t-1)$. It is easy to see that $\beta(\mu^+ - \mu^-)/2 = \sum_{\tau=1}^{t-1} -\epsilon\sigma_D^\tau$ is the exponential term for updating $\sigma_G^{(t)}$. The purified state μ^+ can be generated by querying the subroutine $O_{\sigma_D^T}$ once, i.e., with setting $\gamma_\tau = \sqrt{1/(t-1)}$ for any $\tau \in [t-1]$. Likewise, to prepare $U_{\sigma_D^{(t)}}$, the density operator μ^+ and μ^- refers to the purification of ρ and $\sum_{\tau=1}^{t-1} \sigma_G^{(\tau)}/(t-1)$, respectively. Meanwhile, we have $\beta = 2\epsilon(t-1)$. The corresponding purification state of μ^+ and μ^- can be prepared by querying U_ρ and $O_{\sigma_G^T}$ once, with setting $\gamma_\tau = \sqrt{1/(t-1)}$ for any $\tau \in [t-1]$. By induction, we can prepare all Gibbs samplers and build two subroutines after T training rounds.

C. SM (C) Proof of Theorem 2.

Proof. As discussed in the previous subsection, the main computational cost of QMMW is in the preparation of two subroutines, or equivalently a set of Gibbs samplers. We now employ the conclusion of Theorem 8 to characterize the computation cost of building two subroutines. Observing the conclusion of Theorem 8, the computational cost is highly related to the two variables, i.e., β and δ. Notably, the error θ in Theorem 8 is caused by loading classical input into quantum state, which is not required in QMMW. This error can thus be eliminated in our case and the query complexity transformed to $\mathcal{O}(\beta^{3.5}/\delta)$.

At $(t+1)$-th step, we have $\beta = 2\epsilon t$ with $\epsilon = \sqrt{N/T}$, which leads to the cost $\mathcal{O}(\beta^{3.5}/\delta) \leq \mathcal{O}((NT)^3/\delta)$. Since each training round requires at most $\mathcal{O}((NT)^3/\delta)$ query complexity to prepare two Gibbs samplers, the total query complexity for building two subroutines is $\mathcal{O}(N^3T^4/\delta)$ after T training rounds. By setting δ as a small constant, the query complexity of QMMW is $\mathcal{O}(N^3T^4)$, with the first step and the third step requiring only $\mathcal{O}(1)$ query complexity.

The runtime cost of QMMW is also $\mathcal{O}(N^3T^4)$. The first and third steps of QMMW only require $\mathcal{O}(1)$ and $\mathcal{O}(\log N)$ elementary operations. In second step, μ^+ and μ^- can be prepared with $\mathcal{O}(1)$ queries and using $\mathcal{O}(poly \log(\beta))$ elementary operations [41]. Therefore, the total runtime complexity for QMMW is $\mathcal{O}(N^3T^4)$. $\qquad\square$

D. SM(D) Proof of Lemma 3

Proof of Lemma 3. The zero-sum game played by QuGAL is identical to that of QuGAN, since both possess the convex-concave property and have the same equilibrium, supported by the linear property of

trace operations, and ρ is sampled from a convex set. Concretely, we have

$$\max_{\sigma} \min_{\mathcal{D}} \mathcal{L}(\sigma, \mathcal{D})$$

$$= \max_{\sigma} \min_{\mathcal{D}} \text{Tr}((\mathbb{I} - \mathcal{D})\sigma)P(G) + \text{Tr}(\mathcal{D}\rho)P(R) \tag{30}$$

$$= \max_{U_G} \min_{U_D} \text{Tr}((\mathbb{I} \otimes (\mathbb{I} - E_F))U_D(\sigma_G \otimes |0\rangle \langle 0|)U_D^\dagger)P(G) + \text{Tr}((\mathbb{I} \otimes E_F)U_D(\rho \otimes |0\rangle \langle 0|)U_D^\dagger)P(R) \,,$$

where $\sigma_G = \text{Tr}_a(U_G(|0\rangle \langle 0|)^{\otimes N'}U_G^\dagger)$. The first equality follows the definition of loss function of QuGAL, which is

$$\mathcal{L}(\sigma, \mathcal{D}) = \text{Tr}((\mathbb{I} - \mathcal{D})\sigma)P(G) + \text{Tr}(\mathcal{D}\rho)P(R) \,.$$

The second equality employs Naimark's dilation theorem and Stinespring's dilation theorem.

$\square$

E. SM (E) Generative Adversarial Network

The generator G and discriminator D for classical GANs are typically implemented by multi-layer neural networks [18]. The generator can be treated as a function G, which aims to map from a random variable $z \in \mathbb{R}^{|z|}$ sampling from the latent space to the data space $x \in \mathbb{R}^{|x|}$. Mathematically, we have $G : G(z) \to x$, where $|\cdot|$ denotes the number of dimensions with $|z| \ll |x|$, and x refers to the generated data (e.g., an image with $|x|$ pixels). Discriminator D may be similarly characterized as a function that maps from input data to the class distribution: $D : D(x) \to (0, 1)$, where the training data are expected to be 1 (True) and the generated data are expected to be 0 (False). If the distribution learned by the generator is able to match the real data distribution perfectly, the discriminator will be maximally confused, predicting 0.5 for all inputs. This unique solution whereby D can never discriminate between the generated data and the training data is called Nash equilibrium [19].

The training of GANs involves finding the parameters of a discriminator D to maximize classification accuracy, and finding the parameters of a generator G to maximally confuse the discriminator. The performance of GAN is evaluated using a loss function $L(G, D)$ which depends on both the generator and the discriminator. The training procedure can be treated as:

$$\min_{G} \max_{D} L(G, D), \tag{31}$$

where $L(G, D) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z)))]$, $p_{data}(x)$ refers to the distribution of the training dataset, and $p(z)$ is the probability distribution of the latent variable z. During training, the parameters of one model are updated, while the parameters of the other are fixed.