PAPER • OPEN ACCESS

How to enhance quantum generative adversarial learning of noisy information

To cite this article: Paolo Braccia *et al* 2021 *New J. Phys.* **23** 053024

New Journal of Physics
The open access journal at the forefront of physics

Deutsche Physikalische Gesellschaft DPG
IOP Institute of Physics

Published in partnership with: Deutsche Physikalische Gesellschaft and the Institute of Physics

OPEN ACCESS

RECEIVED
21 December 2020

REVISED
5 March 2021

ACCEPTED FOR PUBLICATION
13 April 2021

PUBLISHED
14 May 2021

PAPER

How to enhance quantum generative adversarial learning of noisy information

Paolo Braccia[1,2,3,*], **Filippo Caruso**[1,3,4,*] and **Leonardo Banchi**[1,2,*]

[1] Dipartimento di Fisica e Astronomia, Università di Firenze, I-50019, Sesto Fiorentino (FI), Italy
[2] INFN, Sezione di Firenze, I-50019, Sesto Fiorentino (FI), Italy
[3] LENS and QSTAR, Via N. Carrara 1, I-50019 Sesto Fiorentino, Italy
[4] Istituto Nazionale di Ottica CNR-INO, Firenze, Italy
[*] Authors to whom any correspondence should be addressed.

E-mail: paolo.braccia@unifi.it, filippo.caruso@unifi.it and leonardo.banchi@unifi.it

Keywords: quantum information, quantum machine learning, quantum algorithms, optimization, noise

Abstract

Quantum machine learning is where nowadays machine learning (ML) meets quantum information science. In order to implement this new paradigm for novel quantum technologies, we still need a much deeper understanding of its underlying mechanisms, before proposing new algorithms to feasibly address real problems. In this context, quantum generative adversarial learning is a promising strategy to use quantum devices for quantum estimation or generative ML tasks. However, the convergence behaviours of its training process, which is crucial for its practical implementation on quantum processors, have not been investigated in detail yet. Indeed here we show how different training problems may occur during the optimization process, such as the emergence of limit cycles. The latter may remarkably extend the convergence time in the scenario of mixed quantum states playing a crucial role in the already available noisy intermediate scale quantum devices. Then, we propose new strategies to achieve a faster convergence in any operating regime. Our results pave the way for new experimental demonstrations of such hybrid classical-quantum protocols allowing to evaluate the potential advantages over their classical counterparts.

1. Introduction

Machine learning (ML) techniques, besides transforming the way we approach huge-data processing problems, are starting to permeate even non-computer science research and applied sectors, leading to new data-driven strategies, including also several concrete applications in our everyday life as domotic systems, autonomous cars, face/voice recognition, and medical diagnostics. One of the most outstanding ML results is provided by the generative adversarial networks (GANs) [1], which are models exploiting game-theory theorems [2] to learn how to reproduce some given data distribution as close as possible. More specifically, two agents, named as the generator and the discriminator, compete against each other in a zero-sum game, i.e. they play in turns, each turn trying to improve their own strategy. The generator tries to reproduce the data and fool the discriminator, whose aim is to correctly distinguish real data from generated ones. Under some reasonable assumptions, basically that the spaces of the strategies are compact and convex, the game has a unique Nash equilibrium point, where the generator is able to exactly reproduce the wanted real data distribution.

In the last few decades quantum information science [3] has focused on how to store and process information with quantum systems, and how they can be exploited to implement more efficient protocols than classical ones. This field is currently leading to the first prototypes of quantum devices, with some of them already reaching the commercial market, especially in the context of quantum communication and quantum sensing protocols. Over the last few years, quantum machine learning (QML) [4, 5], combining ML with quantum information tools, has emerged as one of the most promising applications for near-term

quantum devices. Nowadays quantum processors belong to the so-called noisy-intermediate-scale-quantum (NISQ) era [6]. Circuits running on such devices are characterized by limited size and depth, and the absence of exact error correction protocols makes them still unsuitable for general-purpose computation. However, they can already be employed for variational *hybrid quantum–classical algorithms* [7–11], where the NISQ hardware is paired with a classical device performing the optimization of some classical parameters. The noise resilience and short depth of these algorithms is what makes them suitable for NISQ hardwares. Most efforts in QML research are devoted to exploit quantum resources, such as superposition and entanglement, to achieve quantum speedups over classical ML tasks. In fact, once loaded on a quantum computer, data can be processed exponentially faster than what is possible on classical computers [12, 13]. Clearly, working in a fully quantum landscape is interesting as well, particularly from the perspective of improving quantum simulations [14], control [15], metrology [16], and communication [17] strategies.

In this context, GANs can be successfully generalized to the quantum domain leading to the so-called *quantum* generative adversarial networks (QGANs) [18, 19]. The aim of QGANs is to learn reproducing the state of a quantum system, usually a register of qubits. A way to achieve this goal, as well as implementing learning algorithms over quantum computers, is to leverage the representation power of parametrized quantum circuits (PQCs), which are routinely implemented in current superconducting [20] or ion-trap [21] devices. Examples of learning algorithms realized via PQCs are the quantum approximation optimization algorithm [22], and the variational autoencoders [23] and eigensolvers [24]. A detailed review on PQCs and their features, as well as applications, can be found in reference [25]. Since PQCs are circuits composed of quantum gates controlled by real tunable parameters, they allow us to steer the output state at our will, by adapting the gate parameters to the measurement outcomes. As for classical neural networks, we can set up a gradient-based strategy to optimally update the parameters. Moreover, QGANs have been exploited to learn classical distributions of data [26, 27] and to provide a new tool in learning pure states [19, 28], whereas mixed states (i.e. noisy information) have been addressed only as ensembles of pure data [20]. However, mixed states play a crucial role in the coming NISQ technologies, since the environmental noise is unavoidable and usually partially destroys the quantum features, such as entanglement, that do not have a classical counterpart and that are instead mainly responsible for the predicted quantum speedups.

For these reasons we strongly believe that, in order to more deeply understand the performance of QGANs on real hardware, it is remarkably relevant to investigate the scenario of learning mixed quantum states. This is the main focus of this work. The paper is organized as follows. In section 2 we review the mathematical formalism for QGANs, hence showing why learning mixed states could be an issue. In section 3 we discuss how to implement a QGAN game on PQCs, where we find the emergence of limit-cycles around the target, slowing down the optimization process. Then, we explore a so-called *optimistic* algorithm that allows one to achieve convergence, removing the limit cycles. After a discussion about convex optimization over PQCs (section 4), conclusions and outlooks are drawn in section 5.

2. Quantum adversarial game

In GANs the goal of the discriminator (D) is indeed to discriminate *real* (R) data from the fake ones generated by the generator (G), while the goal of the latter is to fool the discriminator as much as possible by generating *fake* data. Here both real and generated data are modelled as quantum states, respectively described by their density operators ρ_{R} and ρ_{G}. To generate a mixed quantum state, one can begin by creating a generic pure state in a larger Hilbert space by applying a quantum circuit to a qubit register initialized in a pure state (e.g. $|00\ldots0\rangle$), and then tracing out half of the qubit register. Notice that, in practice, the tracing out operation corresponds to ignoring the state of the auxiliary half of the register. The same procedure can be exploited to generate the fake data, but in terms of a PQC where the gate parameters can be tuned. Besides, D applies another PQC to the real or fake state at hand, and entangles it to an additional (ancilla) qubit that later is measured in order to perform the discrimination—see figure 1.

As in the classical case, without any restriction on the operations performed by both agents, the game *should* end when G is able to perfectly reproduce the real data and, accordingly, D is unable to correctly discriminate fake data from real ones. Even in the quantum domain, this corresponds to the unique Nash equilibrium of the underlying game [18]. Let us point out that, before reaching the equilibrium point, both D and G try to iteratively update their strategy to win the game. The D action is modelled via a two-outcome positive operator-valued measure (POVM) Π_i^{D} whose outcome $i \in \{\mathrm{R},\ \mathrm{G}\}$ judges whether the state is real or fake, generated by G. Therefore, at each iteration, D has to solve a binary quantum state discrimination task. The error in such discrimination process is given by the conditional probability of judging real a generated state, i.e. $p(\mathrm{R}|\mathrm{G}) = \mathrm{Tr}[\Pi_{\mathrm{R}}^{\mathrm{D}}\rho_{\mathrm{G}}]$, and by that of judging fake a real state, $p(\mathrm{G}|\mathrm{R}) = \mathrm{Tr}[\Pi_{\mathrm{G}}^{\mathrm{D}}\rho_{\mathrm{R}}] = 1 - \mathrm{Tr}[\Pi_{\mathrm{R}}^{\mathrm{D}}\rho_{\mathrm{R}}]$. Assuming equal *a priori* probabilities, and that G's current state is known by D, we may define the *discrimination error* as $[p(\mathrm{R}|\mathrm{G}) + p(\mathrm{G}|\mathrm{R})]/2$. The discriminator strategy

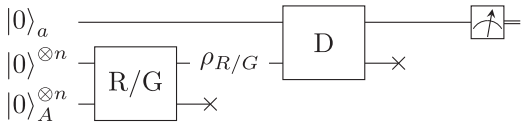


Figure 1. QGAN circuit structure for generic n-qubit mixed states. The R/G/D blocks are PQCs that are exploited to create real/generated data, and to implement the discrimination process, respectively. The discriminator has access to an ancilla qubit a, while the auxiliary qubits A are used by R/G to create mixed states. The $\times$ symbol over a wire means tracing the degrees of freedom associated to it, but it does not correspond to any actual physical operation.

during their turn can be formalized as a minimization of the discrimination error that, without loss of generality, can be written as

$$\text{Discriminator}: \max_{\Pi_D} \text{Tr}[\Pi_D(\rho_R - \rho_G)], \quad \text{with } \rho_R, \rho_G \text{ fixed}, \tag{1}$$

where we set $\Pi_D \equiv \Pi_R^D$ to simplify the notation. An analytic solution to the above optimization is provided by Helstrom's theorem [29, 30], which states that the optimum POVM $\{\Pi_i^D\}$ is formed by projectors onto the positive (Π_R^D) and negative (Π_G^D) subspaces of the operator $\rho_R - \rho_G$. On the other hand, the generator's strategy is to fool the discriminator as much as possible by reducing their ability to distinguish the real and generated states. This results in the following optimal strategy for G

$$\text{Generator}: \min_{\rho_G} \text{Tr}[\Pi_D(\rho_R - \rho_G)], \quad \text{with } \rho_R, \Pi_D \text{ fixed}. \tag{2}$$

This strategy has a formal analytic solution as $\rho_G = |\pi_{max}\rangle\langle\pi_{max}|$, where $|\pi_{max}\rangle$ is the eigenvector of Π_D with maximum eigenvalue. If D is always playing with the optimal Helstrom measurement, then ρ_G is a projection onto an eigenstate of $\rho_R - \rho_G$ with positive eigenvalue.

However, it is simple to show that D and G cannot and *should not* solve the optimization problems in equations (1) and (2) at each iteration. Firstly, they *cannot* find the optimal solution without perfectly knowing, at each iteration, ρ_R and the other player's strategy, which contradicts the original scope of the game. Secondly, they *should not* perform such difficult optimization at each round: if D and G iteratively play using the solution of equations (1) and (2), then they never reach the equilibrium for mixed states ρ_R. This is summarized by the following:

Remark 1. For mixed states ρ_R, the adversarial game fails to converge when D and G iteratively use the strategies in equations (1) and (2).

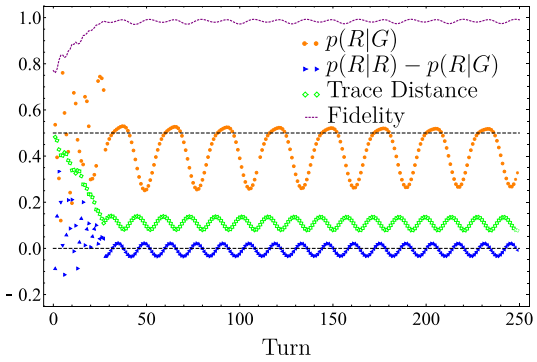
Indeed, as discussed above, the solution of equation (2) is always a pure state $\rho_G = |\pi_{max}\rangle\langle\pi_{max}|$, and as such $\rho_G \neq \rho_R$ in general, thus proving the claim of remark 1. To achieve convergence, each player must *slightly* update their strategy at each operation [18], rather than using equations (1) and (2). Moreover, in the language of Nash equilibria, each player is unaware of the adversary's move, and the best they can do is to assume that the opponent is playing with the optimal strategy and fight against it. Setting the *score* as the bilinear function

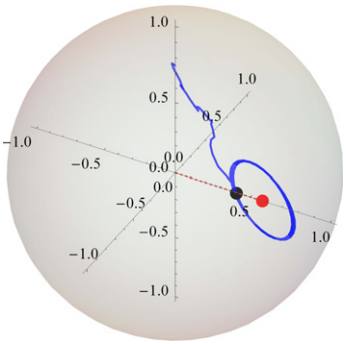
$$S(\rho_G, \Pi_D) = \text{Tr}[\Pi_D(\rho_R - \rho_G)], \tag{3}$$

we see that G increases its score whenever D loses the same amount, making this a zero-sum game. Both the states ρ_G and the measurement operators Π_D form a convex set in their respective spaces.[5] Therefore, the Nash equilibrium is the result of the minimax problem $\min_{\rho_G} \max_{\Pi_D} S(\rho_G, \Pi_D) = \frac{1}{2} \min_{\rho_G} \|\rho_R - \rho_G\|_1 = 0$ where the first equality follows from the Helstrom theorem and the definition of the one-norm [3]. As a result, the Nash equilibrium is when the generator is able to perfectly reproduce ρ_R, as originally shown in [18]. However, how to achieve in practice this equilibrium configuration is far from being trivial.

Inspired by the success of gradient-based training of GANs [31], the most natural approach to play the quantum adversarial game is to use a suitable parametrizations of ρ_G and Π_D, see e.g. figure 1, and then iteratively update these parameters, e.g. via gradient descent [18]. Using these methods, convergence with pure target states $\rho_R = |\psi_R\rangle\langle\psi_R|$ has been obtained in several scenarios [19, 28], while for mixed states convergence was observed with a few extra steps, e.g. by setting ρ_G as a random superposition of pure states [20]. In spite of these successful examples, gradient-based training may be problematic, as we numerically investigate in the next section. This is due to the bilinear nature of the score function in equation (3). Indeed it has been shown that the adversarial optimization of bilinear score functions may display limit

[5] Since Π_D is part of a POVM it is a positive operator with $\|\Pi_D\|_\infty < 1$.





a) Figures of merit. **b)** Trajectory of ρ_G in the Bloch ball.

Figure 2. Emergence of a limit cycle when the score in equation (4) is optimized via GDA, as described in the main text. Here the learning rate of both agents is $\eta = 0.1$, and one training turn consists of five discriminator's steps followed by a single generator's one.

cycles when trained with standard gradient descent rules, or even a 'chaotic' behaviour, see e.g. [32, 33] and references therein.

More precisely, let us consider the simplest case where ρ_R is a single-qubit mixed state. The most natural parametrization of ρ_R is via the Bloch vector $\vec{r}$, namely $\rho_R = [\mathbb{1} + \vec{r}\cdot\vec{\sigma}]/2$ where $\vec{\sigma}$ is the vector of Pauli matrices and $|\vec{r}| \leqslant 1$. Similarly we parametrize ρ_G with the Bloch vector $\vec{g}$ and $\Pi_D = [d^0\mathbb{1} + \vec{d}\cdot\vec{\sigma}]/2$, where $d^0 = \mathrm{Tr}\,\Pi_D$ and $d^0 \geqslant |\vec{d}|$ (because $\Pi_D \geqslant 0$). With this simple parametrization equation (3) becomes a bilinear form in the Bloch vectors

$$S(\Pi_D, \rho_G) = \frac{\vec{d}\cdot(\vec{r} - \vec{g})}{2}. \tag{4}$$

The above score function has been extensively investigated in references [34, 35] where the emergence of limit cycles in classical GANs training was shown. Nonetheless, references [34, 35] focus on bilinear problems with linear constraints, while Bloch vectors satisfy a non-linear constraint since they live in the Bloch ball. This difference may be the reason behind the good performance of quantum adversarial learning for pure states [19, 28], as pure states lie at the boundary of the Bloch sphere where such non-linear constraints are important. However, when dealing with the optimization of highly mixed states, which lie well inside the Bloch ball, the presence of the boundary may not affect the optimization, and limit cycles may emerge. We summarise this aspect in the following remark, whose proof, adapted from [34], can be found in appendix.

Remark 2. Gradient descent/ascent (GDA) applied to the problem $\min_{\rho_G}\max_{\Pi_D} S(\Pi_D, \rho_G)$ diverges for states far from the surface of the Bloch sphere.

We bring evidence to the previous statement by running a QGAN game in a single qubit scenario where both D and G are parametrized via their Bloch vectors. We employ GDA—namely gradient descent for $\vec{g}$ and gradient ascent for $\vec{d}$—on the score function in equation (4) with an added penalty term to enforce the constraints on the Bloch vectors, i.e. $\|\vec{g}\| \leqslant 1$ and $\|\vec{d}\| \leqslant d^0 \leqslant 2 - \|\vec{d}\|$. Results are shown in figure 2, where the limit cycle behaviour in the trajectory of $\vec{g}$ is evident.

An algorithm dubbed 'optimistic mirror descent' (OMD) has been proposed in reference [34] to escape from the limit cycles that emerge in the minimax optimization of bilinear cost functions in equation (4). In the next section we show that, although perfect limit cycles may not exist for more complex parametrizations of ρ_G and Π_D, a simple GDA update may produce a 'chaotic' behaviour, where convergence is not observed. We find instead that convergence is obtained via OMD.

3. Training with parametric quantum circuits

Motivated by the capabilities of current noisy intermediate-scale quantum hardwares [6], common PQCs are based on single qubit gates controlled by tunable real parameters, e.g. qubit rotations around a fixed axis with variable angle, and non-parametric two-qubit gates, typically CNOTs.[6] Sequences of single and

[6] The controlled-NOT (CNOT) operation applies Pauli's X to the target qubit if the control one is found in $|1\rangle$ and does nothing otherwise.

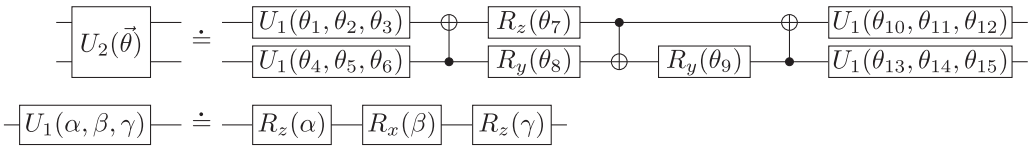


Figure 3. The building block of G and D parametric circuits with three CNOTs and 15 single-qubit rotations. U_1 and U_2 implement elements of SU(2), controlled by three angles $\{\alpha,\beta,\gamma\}$, and SU(4), controlled by 15 parameters $\vec{\theta}$, respectively. Here $R_i(\theta)=\exp\left[-\mathrm{i}\theta/2\cdot\sigma_i\right]$ are rotations around the ith axis.

two-qubit gates are then stacked in a layered fashion. When sketched down, it is easy to see a certain resemblance with classical neural networks, with the parameters playing the role of the weights and biases of the latter. Indeed, it turns out that, just as a neural network with enough parameters and proper activations can represent any function, a PQC with suitable depth and structure can approximate any unitary mapping over the input quantum register. Indeed, CNOT gates and single-qubit rotations are universal for quantum computation, i.e. they can be composed to simulate any unitary evolution to the desired accuracy [36]. Notice that, however, finding the best circuital scheme to achieve such accuracy from case to case is still a matter of debate [37], and scalability of QML is also an hot topic [38, 39]. PQCs can be designed to comply with nowadays NISQ hardware, by adapting the two-qubit (entangling) gates to the connectivity of the experimental realization of the quantum processor, and by limiting the circuit's depth to fight decoherence. Since learning tasks ultimately boil down to the problem of minimizing a certain loss/score function of the model parameters, we can employ PQCs as quantum learning models and tune them through a feedback loop with a classical optimizer. This is the standard framework of hybrid quantum–classical variational approaches [8], and we will use this scheme in our analysis.

Here, we will ultimately be concerned with the problem of learning a mixed state via a QGAN game. Since every mixed state can be written as a pure state in a larger Hilbert space (figure 1), we build the generator via the following PQC with classical parameters $\boldsymbol{\theta}_\mathrm{G}$

$$\rho_\mathrm{G}=\mathrm{Tr}_\mathrm{A}\left[|\psi_\mathrm{GA}(\boldsymbol{\theta}_\mathrm{G})\rangle\langle\psi_\mathrm{GA}(\boldsymbol{\theta}_\mathrm{G})|\right],\qquad |\psi_\mathrm{GA}(\boldsymbol{\theta}_\mathrm{G})\rangle=U(\boldsymbol{\theta}_\mathrm{G})|0\rangle^{\otimes 2n}, \tag{5}$$

where both A and ρ_G contain n qubits, and $U(\boldsymbol{\theta}_\mathrm{G})$ is the unitary operator corresponding to the PQC. Similarly, since every measurement operator can be written as a projective measurement onto a larger Hilbert space (figure 1), we define the discriminator's POVM with classical parameters $\boldsymbol{\theta}_\mathrm{D}$ as

$$\Pi_\mathrm{D}=\mathrm{Tr}_\mathrm{a}\left[U(\boldsymbol{\theta}_\mathrm{D})^\dagger(\mathbb{1}_\mathrm{D}\otimes|0\rangle_\mathrm{a}\langle 0|)U(\boldsymbol{\theta}_\mathrm{D})(\mathbb{1}_\mathrm{D}\otimes|0\rangle_\mathrm{a}\langle 0|)\right], \tag{6}$$

where a is a single auxiliary qubit. This measurement can be interpreted as follows: first apply a PQC $U(\boldsymbol{\theta}_\mathrm{D})$ entangling the system with an auxiliary qubit a, initially in $|0\rangle$, then measure the qubit a in the computational basis. If the outcome 0 is detected, then we guess that the state is the real state, otherwise (outcome 1) the state is judged as fake.

3.1. Circuits ansatz

Following references [28, 40], G and D circuits are built by repeating a two-qubit block which implements a generic unitary $U\in\mathrm{SU}(4)$. As shown in figure 3, this building block is composed of 15 single-qubit rotations and three CNOT gates. One block allows to generate every two-qubit pure state. For larger registers, we apply this block to each pair of consecutive qubits, thus obtaining a layer. Layers are then concatenated in a staggered pattern.

Throughout this manuscript, the gradients needed to train the QGAN are analytically computed through the *parameter shift rule* [41–43]. Since the parametrized part of our circuit consists of single-qubit rotations, the gradient of a function $f(\vec{\theta})=\langle O(\vec{\theta})\rangle$ is obtained as

$$\frac{\partial f}{\partial\theta_i}=\frac{1}{2}\left[f\left(\vec{\theta}+\frac{\pi}{2}\vec{e}_i\right)-f\left(\vec{\theta}-\frac{\pi}{2}\vec{e}_i\right)\right], \tag{7}$$

where $\vec{e}_i$ is the unit vector in the ith direction. Notice that, since the estimation of each derivative involves running the circuit twice, the parameter-shift rule makes it possible to run gradient-based hybrid quantum–classical algorithms on NISQ devices, provided that the circuit itself is device-friendly.

We first focus on learning pure states, for which it is known that QGANs converge. Indeed, figure 4 does further confirm it in terms of the relevant figures of merit, as the score function value, the probability $p(\mathrm{R}|\mathrm{G})$ of D labelling fake data as true, the trace distance $d=\frac{1}{2}\|\rho_\mathrm{G}-\rho_\mathrm{R}\|_1$ between the generated state and the target one, and their fidelity $F=\left[\mathrm{Tr}\sqrt{\sqrt{\rho_\mathrm{G}}\rho_\mathrm{R}\sqrt{\rho_\mathrm{G}}}\right]^2$. In our simulations the target *real* data are

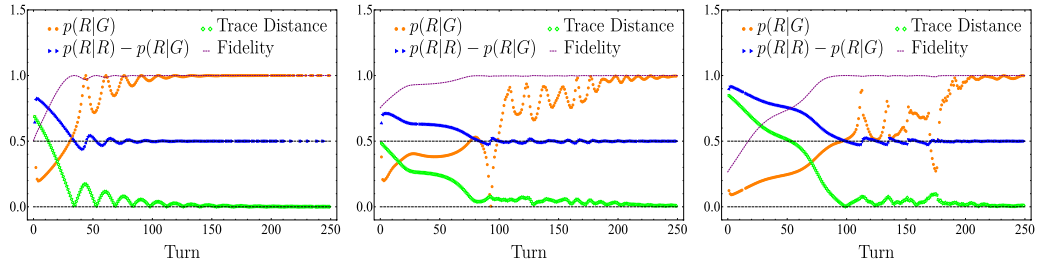


Figure 4. Convergence of QGANs for learning pure states of n qubits. Both agents rely on standard GDA optimization. The target and the initial fake states are randomly chosen on the Bloch sphere. The orange-dotted and the blue-triangle-shaped curves are the figures of merit of G and D, respectively, whereas the green-square-shaped and purple-dashed ones are accuracy metrics for the training, respectively the trace distance between target and generated state and their fidelity.

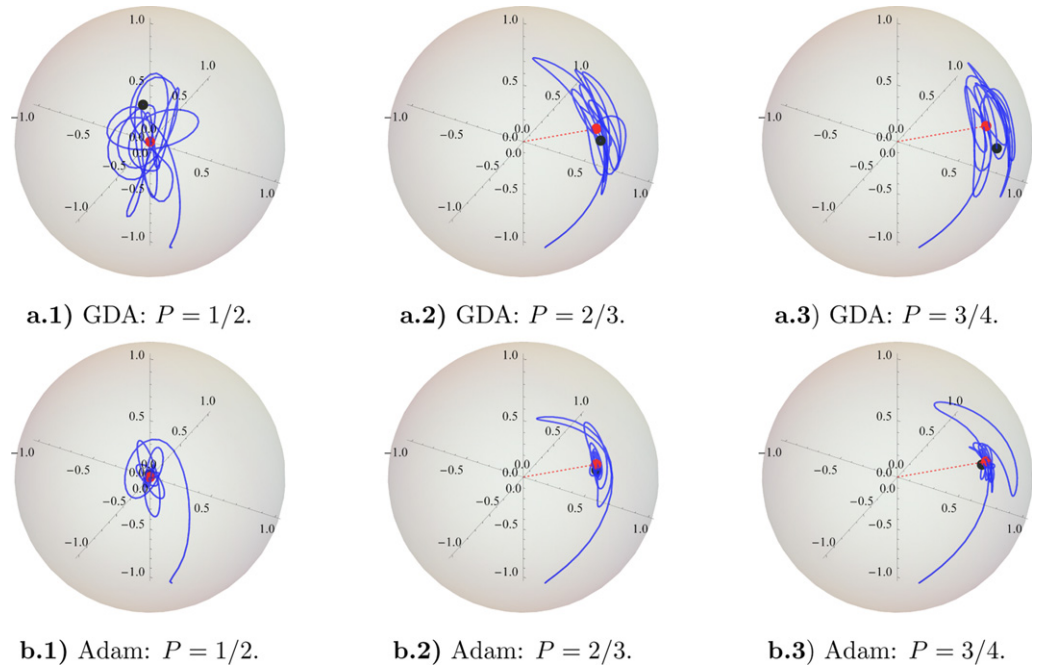


Figure 5. Limit-cycle-like behaviour of QGANs when learning single-qubit mixed states for different values of the purity P. As optimizers, we use GDA (top row) and Adam (bottom). None of them display convergence, although the latter has a less pronounced oscillating behaviour. In all these cases the initial configurations of G and D correspond to the same random parameters. These Bloch ball trajectories have been obtained by running the QGAN for 250 total turns, where each turn comprises ten optimization steps for D followed by 1 for G. Lastly, the learning rate of GDA is set to 0.1, whereas that of Adam is 0.05.

random pure states of n qubits, with $n = 1, 2, 3$, obtained via a PQC with the same structure of the one used for G, but with random fixed parameters. Here, training is carried out via alternately updating D and G via a single GDA step. We have tried different optimizers, always observing a qualitatevely similar convergence behaviour[7].

3.2. Emergence of limit cycles

We now generalize the above analysis to the more interesting case of learning mixed states. They have been so far addressed as an ensemble of orthogonal pure states [20], while here they are created by tracing out half of the qubits register. Our results are summarized in figure 5, where we show the learning process for mixed states of the form $\rho_{\mathrm{R}} = \frac{I}{2} + \frac{a}{2\sqrt{2}}(\sigma_x + \sigma_y)$ with purity $P = \mathrm{tr}\,\rho^2 = \frac{1+a^2}{2}$, ranging from the completely mixed one $P = 1/2$ to $P = 3/4$. The selected optimizers are the previously defined GDA and *Adam*, i.e. one of the best performing optimization algorithm for ML [44]. As we can see in figure 5, none of the chosen

[7] Hyperparameters such as the learning rates might be further fine-tuned in order to improve convergence speed, but this is beyond the aim of this manuscript.

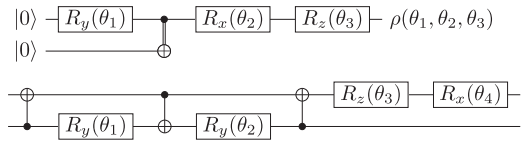


Figure 6. The minimal circuits for G (top), and D (bottom). The circuital elements are defined in figure 3.

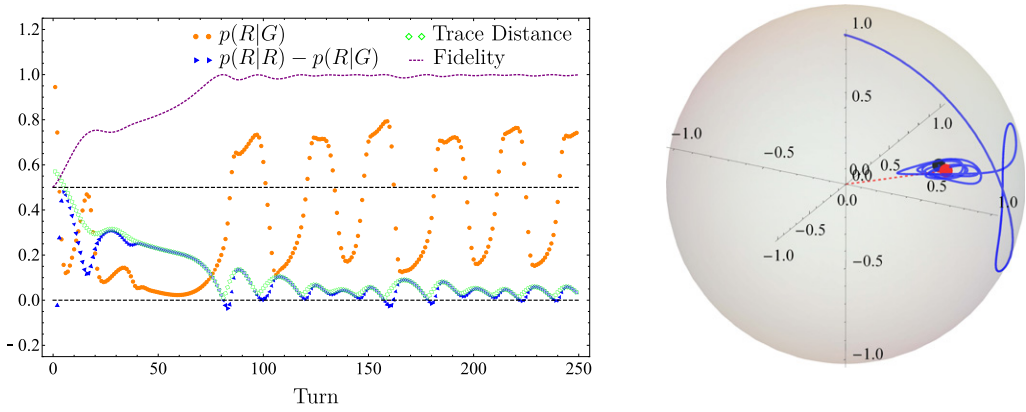


Figure 7. Training behaviour for learning a mixed state with $P = 3/4$ under the same conditions of figure 5 with Adam and the tailored agents, but a different initial configuration. The curves shown in (a) are explained in figure 4, whereas (b) shows the trajectory of the generated state in the Bloch ball.

optimizers allows to reach convergence, even by changing the values of the optimization hyperparameters. However, comparing these results with the ones in figure 2, we can see that for PQCs the limit-cycle behaviour disappears because the score function is no longer bilinear. Let us point out that in figure 5 we have an overparametrization because D and G use 15 parameters each, whereas a general single-qubit POVM has four real degrees of freedom only, and a single qubit mixed state has three. For this reason we devise two tailored circuits in order to achieve a minimal parametrization for both D and G (see figure 6), as in the following:

$$\rho_{\rm G}(\boldsymbol{\theta}) = \frac{1}{2}\begin{pmatrix} 1 + c(\theta_1)c(\theta_2) & c(\theta_1)s(\theta_2)\,(s(\theta_3) + ic(\theta_3)) \\ c(\theta_1)s(\theta_2)\,(s(\theta_3) - ic(\theta_3)) & 1 - c(\theta_1)c(\theta_2) \end{pmatrix}, \tag{8}$$

and

$$\Pi_{\rm D}(\boldsymbol{\theta}) = \frac{1}{2}\begin{pmatrix} 1 + c\,(\theta_1 + \theta_2)\,c\,(\theta_4) & s(\theta_4)\,(c(\theta_1)s(\theta_3) - ic(\theta_2)c(\theta_3)) \\ s(\theta_4)\,(c(\theta_1)s(\theta_3) + ic(\theta_2)c(\theta_3)) & 1 - c\,(\theta_1 - \theta_2)\,c\,(\theta_4) \end{pmatrix}, \tag{9}$$

with $\cos(\theta) \equiv c(\theta)$ and $\sin(\theta) \equiv s(\theta)$. Even with these tailored circuits, convergence is not achieved as numerically reported in figure 7. Moreover, by using simple gradient descent method we still observed limit cycles. We do not report such figure, as it displays even worse convergence than figure 7.

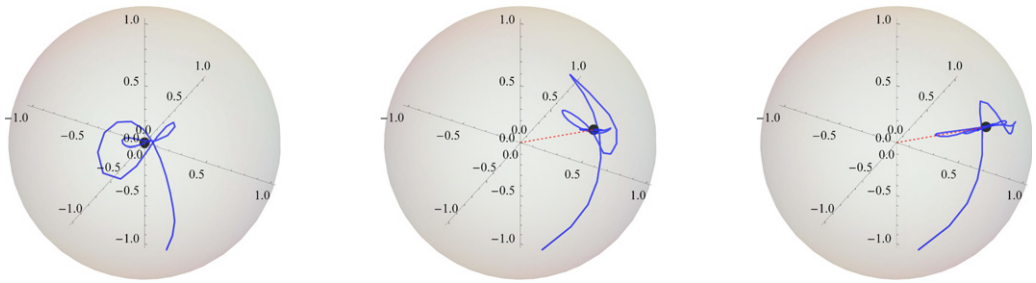
3.3. Training with optimism

In standard GANs competing players are usually unaware of the opponent's strategy. However, each player may try to guess the opponent's move in order to improve its strategy. This is the building concept of the OMD optimization algorithm [45], which was shown to fix convergence issues, namely limit cycles, of classical GANs with bilinear score functions—see reference [34]. However, there it has been used to enhance convergence also in the case of non-bilinear score functions. Motivated by these results, we now show that OMD works successfully also for our QGANs—see figure 8. More specifically, the OMD-based update rule for the score function of equation (3), $S(\boldsymbol{\theta}_{\rm D}, \boldsymbol{\theta}_{\rm G}) := S(\Pi_{\rm D}(\boldsymbol{\theta}_{\rm D}), \rho_{\rm G}(\boldsymbol{\theta}_{\rm G}))$, reads

$$\boldsymbol{\theta}_{\rm D}^{t+1} = \boldsymbol{\theta}_{\rm D}^{t} + 2\eta_{\rm D}\nabla_{\boldsymbol{\theta}_{\rm D}} S(\boldsymbol{\theta}_{t}^{\rm D}, \boldsymbol{\theta}_{t}^{\rm G}) - \eta_{\rm D}\nabla_{\boldsymbol{\theta}_{\rm D}} S(\boldsymbol{\theta}_{t-1}^{\rm D}, \boldsymbol{\theta}_{t-1}^{\rm G}), \tag{10}$$

$$\boldsymbol{\theta}_{\rm G}^{t+1} = \boldsymbol{\theta}_{\rm G}^{t} - 2\eta_{\rm G}\nabla_{\boldsymbol{\theta}_{\rm G}} S(\boldsymbol{\theta}_{t+1}^{\rm D}, \boldsymbol{\theta}_{t}^{\rm G}) + \eta_{\rm G}\nabla_{\boldsymbol{\theta}_{\rm G}} S(\boldsymbol{\theta}_{t}^{\rm D}, \boldsymbol{\theta}_{t-1}^{\rm G}), \tag{11}$$

where $\eta_{\rm D/G}$ are the learning rates for D and G. Notice that this rule corresponds to the scenario where D is optimized first.



a) OMD: $P = 1/2$. **b)** OMD: $P = 2/3$. **c)** OMD: $P = 3/4$.

Figure 8. Bloch ball trajectories showing the convergence of QGANs for learning mixed states of different purities via OMD, with the same number of turns and agents' steps per turn, and the same target states, used in figure 5. The learning rate was set to $\eta = 0.25$ for both D and G.

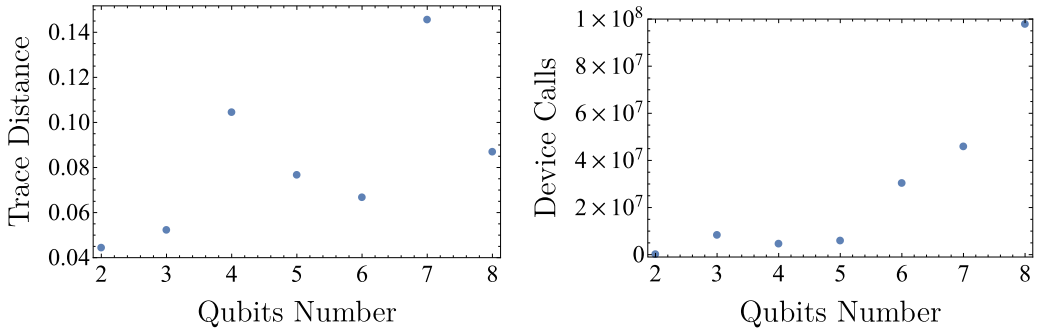


Figure 9. Left: final values of the trace distance between target and generated states after a fixed number of training turns $I = 5000$. Right: number of circuits' evaluations needed in order for the generated state to reach a trace distance from the target that is less than 0.1. In both cases we studied mixed states of two to eight qubits. During each QGAN turn D took 20 steps and G a single one. We used OMD as optimizer and the learning rates for both agents were set to 0.1. The parameters of the circuit generating the target states were chosen so that the latter had purities $P \in [0.5, 0.6]$.

As a proof of principle of the extendibility of this training scheme to higher dimensional states, we have tested the QGAN against a simple class of multi-qubit mixed states. Particularly, we have generated a set of target states of n-qubits by acting on an $n+1$ dimensional qubits register with a train of general $SU(4)$ blocks, again generated via the circuits of figure 3, and then ignoring the last qubit. We have used the same ansatz for G. More precisely, this architecture is such that the first $SU(4)$ block acts on qubits $(1, 2)$, the second on qubits $(2, 3)$ and so on until the $(n, n+1)$ pair. Analogously, we designed the discriminator as an opposite sequence of blocks, starting on the $(n-1, n)$ pair of qubits and ending on the $(0, 1)$ one, with 0 labelling the ancilla qubit used to implement the discriminator's POVM. We chose such a simple scheme for the sake of numerical simulations, a deeper analysis on the best circuital ansatze is left for future work. Results are shown in figure 9. Our findings are encouraging as far as the scalability of the method is concerned, but a full analysis on the possible emergence of barren plateaus [46] and how to tweak the QGAN's hyperparameters in order to escape them is still needed.

4. Convex optimization

Since PQCs are not the only way of modelling quantum states, here we present a non-parametric method, hereafter dubbed ConvexQGAN, to solve the minimax problem $\min_{\rho_G} \max_{\Pi_D} S(\rho_G, \Pi_D)$ using the formalism of convex optimization presented in reference [47]. Since $\{\rho_G\}$, $\{\Pi_D\}$ are both convex sets and $S(\rho_G, \Pi_D)$ is bilinear, when we iteratively fix either ρ_G or Π_D we always obtain a convex function over a convex set. Therefore, by adapting the Frank–Wolfe algorithm from reference [47], we may write the following update rules at the kth step

$$\Pi_D^{k+1} = (1 - \alpha_k)\Pi_D^k + \alpha_k |D_k\rangle\langle D_k|, \tag{12}$$

$$\rho_G^{k+1} = (1 - \beta_k)\rho_G^k + \beta_k |G_k\rangle\langle G_k|, \tag{13}$$

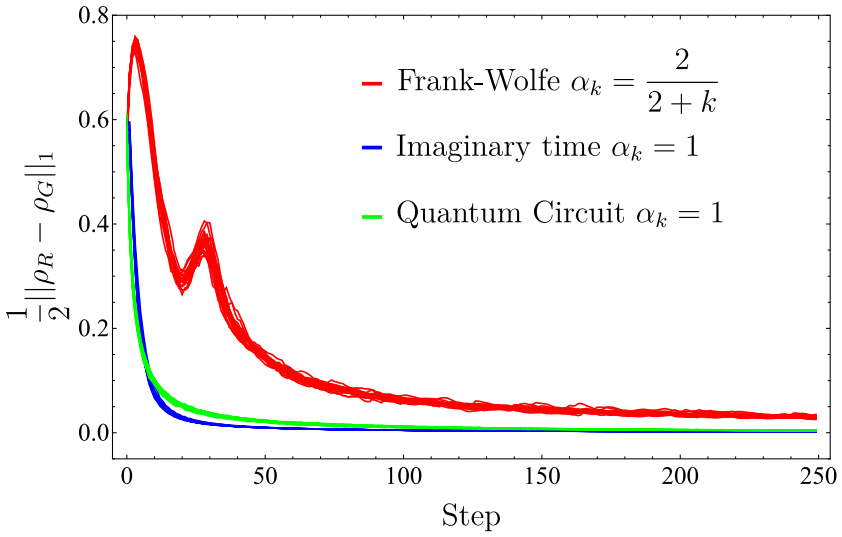


Figure 10. ConvexQGAN: learning different random mixed states using either the Frank–Wolfe iteration equations (12) and (13), the imaginary time evolution equation (14) or the quantum circuit update from equation (17). The cases with $\alpha_k = 1$ in equation (13) correspond to Helstrom measurements. For each algorithm, 20 lines are plotted for different random initial states. All simulations are for five-qubit states.

where α_k and β_k are decaying learning rates, e.g. typically $\alpha_k = \beta_k = \frac{2}{k+2}$, the state $|\mathrm{G}_k\rangle$ is the eigenvector with smallest eigenvalue of $\nabla_{\rho_\mathrm{G}} S(\rho_\mathrm{G}^k, \Pi_\mathrm{D}^k) = -\Pi_\mathrm{D}^k$, while $|\mathrm{D}_k\rangle$ is the eigenvector with smallest eigenvalue of $-\nabla_{\Pi_\mathrm{D}} S(\rho_\mathrm{G}^k, \Pi_\mathrm{D}^k) = -(\rho_\mathrm{R} - \rho_\mathrm{G}^k)$. Although the update rules directly follow from the Frank–Wolfe algorithm, we highlight here an interesting result from the physics points of view. The states $|\mathrm{D}_k\rangle$ are elements of Helstrom measurement to optimally distinguish the real state ρ_R from the current fake state ρ_G^k. As such, it is tempting to consider a different strategy with $\alpha_k = 1$ at each iteration step. The downside of the latter approach is that the measurement operator highly fluctuates between different steps. However, for $\alpha_k = 1$ we get $|\mathrm{D}_k\rangle = |\mathrm{G}_k\rangle$ so equation (13) gets a clear operational meaning. The generator's state is iteratively updated with one of the states entering in the Helstrom optimal measurement. This reminds us the original optimization from equation (2), but without its convergence issues for mixed states. Indeed, the update rule of equation (13) allows the generation of mixed states, unlike in equation (2).

Finally we propose a physics inspired alternative by observing that, for small β_k, equation (13) can be interpreted as an imaginary time evolution

$$\rho_\mathrm{G}^{k+1} \propto \mathrm{e}^{\beta_k H_k} \rho_\mathrm{G}^k \, \mathrm{e}^{\beta_k H_k}, \qquad H_k = |G_k\rangle\langle G_k|, \tag{14}$$

where after the imaginary evolution we need to normalize the state such as $\mathrm{Tr}[\rho_\mathrm{G}^{k+1}] = 1$. The gradient-based Frank–Wolfe algorithm equations (12) and (13) and the imaginary time iteration equation (14) are numerically studied in figure 10 for random five-qubit states with full-rank. For the imaginary time iteration we use $\alpha_k = 1$, so $|\mathrm{G}_k\rangle = |\mathrm{D}_k\rangle$ in equation (14). We observe in figure 10 that the imaginary time evolution, together with the optimal Helstrom measurement at each step, significantly outperforms the Frank–Wolfe iteration, both in terms of speed and accuracy.

ConvexQGAN methods show fast convergence towards the equilibrium configuration, but they require eigendecompositions of the state at each step. This operation is simple for classical computers as long as the Hilbert space is sufficiently small. To extend this operation to larger systems, we now discuss how to write an update like in equation (14), but using a quantum circuit. For this purpose, we use the following quantum map, which is at the heart of the quantum density matrix exponentiation algorithm [48],

$$\mathcal{E}_\sigma^{\pm t}[\rho] = \mathrm{Tr}_2[\mathrm{e}^{\pm it\ \mathrm{SWAP}} \rho \otimes \sigma \mathrm{e}^{\mp it\ \mathrm{SWAP}}] = \cos(t)^2 \rho + \sin(t)^2 \sigma \pm \frac{i}{2} \sin(2t)[\rho, \sigma], \tag{15}$$

where SWAP is the swap operator. Applying this map twice with different signs, one has

$$\mathcal{E}_\sigma^{-t} \circ \mathcal{E}_\sigma^{+t}[\rho] = \cos(t)^4 \rho + \sin(t)^2(1 + \cos^2 t)\sigma + \frac{1}{4} \sin(2t)^2 [[\rho, \sigma], \sigma]. \tag{16}$$

Therefore, setting $I_\sigma^t[\rho] = \mathcal{E}_\sigma^{-t} \circ \mathcal{E}_\sigma^{+t}[\rho]$ and t_k such that $\cos^4 t_k = 1 - \beta_k$, namely $\beta_k \approx 2t_k^2$, we get

$$\rho_\mathrm{G}^{k+1} = I_{H_k}^{t_k}[\rho_\mathrm{G}^k] = (1 - \beta_k)\rho_\mathrm{G}^k + \beta_k(H_k + H_k\rho_\mathrm{G}^k + \rho_\mathrm{G}^k H_k - 2H_k\rho H_k) + \mathcal{O}(\beta_k^2), \tag{17}$$

where H_k was defined in equation (14). The latter update rule is akin to a mixture of equations (13) and (14), but it has the advantage that it can be explicitly evaluated as a quantum circuit applied to ρ_k and two copies of the state $|G_k\rangle$. As shown in figure 10, the performance obtained with the update rule equation (17) is similar to that of imaginary time evolution. Therefore, if the states $|G_k\rangle$ can be efficiently prepared, for instance via strategies like the Helstrom classifier circuit from [49], then the above update rule can be used for QGAN training in a quantum computer.

5. Conclusions

In this work we have studied the convergence of quantum generative adversarial learning for mixed states. We have first showed that 'learning' via simple GDA updates, or even via more advanced methods such as the Adam optimizer, may be problematic when the target state is mixed. We have attributed such convergence issues to the bilinear nature of QGAN's score function. Indeed, it is known from classical GAN literature that the optimization of such score functions leads to limit cycles, where the generator gets stuck into cycling around the target solution without ever reaching it. We have observed that states obtained via PQCs, such as those commonly implemented in nowadays available NISQ devices, are less affected by the emergence of limit cycles, but may nonetheless display a 'chaotic' behaviour during training, without achieving convergence.

We have then proposed new algorithms for reliable training of QGANs, which always achieve convergence in our numerical simulations. The first algorithm, suitable for PQCs, is based on the adaptation of OMD, i.e. a gradient-based technique allowing provable convergence with bilinear score functions. The second algorithm is based on convex optimization techniques, and is especially suited for non-parametric states that are iteratively updated via a suitably designed, yet non-parametric, quantum circuit.

Thanks to our theoretical and numerical analysis, we believe that the proposed algorithms should work better than previously used techniques for QGAN training, especially when highly mixed states are involved. Having good training heuristics for learning mixed states will help in leveraging their higher representation power, as well as in providing us with a way to study noisy quantum maps. Indeed, a next necessary step for the classification of QGANs capabilities is the analysis of their performance against noise. This path has been paved in [50], where it was shown that adversarial schemes share the noise robustness of other known hybrid quantum–classical variational algorithms [51–54]. Lastly, from the physics point of view, since QGANs perform an implicit state tomography, we believe that, by further endowing our scheme with the ability to process entangled copies of the target state, performance will be enhanced. It is an open question whether an adversarial strategy may take over some current metrology scheme [16], by providing faster and more efficient strategies for sensing and system certification. Our results shed new light on how hybrid classical-quantum QML protocols might be exploited in already available experimental platforms with potential promising applications in quantum computing and noise sensing.

Acknowledgments

LB acknowledges support by the program 'Rita Levi Montalcini' for young researchers, Grant No. PGR15V3JYH, funded by 'Ministero dell'Istruzione, dell'Università e della Ricerca (MIUR)'. FC was financially supported by the Fondazione CR Firenze through the projects QUANTUM-AI, the PATHOS EU H2020 FET-OPEN Grant No. 828946, and the Florence University Grant Q-CODYCES.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Appendix. Proof of remark 2

The following proof is adapted from [34]. We use the Bloch parametrization equation (4), which we rewrite for simplicity as $S(\vec{d},\vec{g}) = \vec{d}\cdot(\vec{r}-\vec{g})$ ignoring the constant factors. GDA applied to $\min_{\vec{g}}\max_{\vec{d}} S(\vec{d},\vec{g})$ results in the update rule

$$\vec{d}_{t+1} = \vec{d}_t + \eta(\vec{r} - \vec{g}_t), \tag{A.1}$$

$$\vec{g}_{t+1} = \vec{g}_t - \eta(-\vec{d}_t), \tag{A.2}$$

where η is a suitably small 'learning rate' and t is the iteration step. The unique fixed point is with $\vec{g} = \vec{r}$ and $\vec{d} = 0$, which physically corresponds to perfect generation $\rho_G = \rho_R$ and impossibility to distinguish real from generated data $\Pi_D \propto \mathbb{1}$. We evaluate the distance from this fixed point as $\Delta_t = \|d_t\|_2^2 + \|r - g_t\|_2^2$, where $\|\cdot\|_2$ is the ℓ_2-norm. From equations (A.1) and (A.2) we get

$$\|\vec{d}_{t+1}\|_2^2 = \|\vec{d}_{t+1}\|_2^2 + 2\eta\, \vec{d}_t \cdot (\vec{r} - \vec{g}_t) + \eta^2 \|\vec{r} - \vec{g}_t\|_2^2, \tag{A.3}$$

$$\|\vec{r} - \vec{g}_{t+1}\|_2^2 = \|\vec{r} - \vec{g}_t\|_2^2 - 2\eta\, \vec{d}_t \cdot (\vec{r} - \vec{g}_t) + \eta^2 \|\vec{d}_{t+1}\|_2^2, \tag{A.4}$$

and accordingly

$$\Delta_{t+1} = (1 + \eta^2)\Delta_t, \tag{A.5}$$

namely the distance from the equilibrium point increases at each iteration. We point out that the above proof is valid only when we neglect the physical constraints on the Bloch vectors. The latter are however important for pure states or for states near the surface of the Bloch sphere.

ORCID iDs

Paolo Braccia https://orcid.org/0000-0003-0158-0943
Filippo Caruso https://orcid.org/0000-0002-8366-4296
Leonardo Banchi https://orcid.org/0000-0002-6324-8754

References

[1] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y 2014 Generative adversarial nets *Advances in Neural Information Processing Systems* 2672–80
[2] Kakutani S *et al* 1941 *Duke Math. J.* **8** 457–9
[3] Watrous J 2018 *The Theory of Quantum Information* (Cambridge: Cambridge University Press)
[4] Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N and Lloyd S 2017 *Nature* **549** 195–202
[5] Lamata L 2020 *Mach. Learn.: Sci. Technol.* **1** 033002
[6] Preskill J 2018 *Quantum* **2** 79
[7] Zhu D *et al* 2019 *Sci. Adv.* **5** eaaw9918
[8] McClean J R, Romero J, Babbush R and Aspuru-Guzik A 2016 *New J. Phys.* **18** 023023
[9] McClean J R, Kimchi-Schwartz M E, Carter J and De Jong W A 2017 *Phys. Rev.* A **95** 042308
[10] Cerezo M *et al* 2020 arXiv:2012.09265
[11] Bharti K *et al* 2021 arXiv:2101.08448
[12] Harrow A W, Hassidim A and Lloyd S 2009 *Phys. Rev. Lett.* **103** 150502
[13] Liu Y, Arunachalam S and Temme K 2020 arXiv:2010.02174
[14] Georgescu I M, Ashhab S and Nori F 2014 *Rev. Mod. Phys.* **86** 153
[15] Dong D and Petersen I R 2010 *IET Control Theory & Appl.* **4** 2651–71
[16] Giovannetti V, Lloyd S and Maccone L 2011 *Nat. Photon.* **5** 222
[17] Gisin N and Thew R 2007 *Nat. Photon.* **1** 165–71
[18] Lloyd S and Weedbrook C 2018 *Phys. Rev. Lett.* **121** 040502
[19] Dallaire-Demers P L and Killoran N 2018 *Phys. Rev.* A **98** 012324
[20] Hu L *et al* 2019 *Sci. Adv.* **5** eaav2761
[21] Sriarunothai T, Wölk S, Giri G S, Friis N, Dunjko V, Briegel H J and Wunderlich C 2018 *Quantum Sci. Technol.* **4** 015014
[22] Farhi E, Goldstone J and Gutmann S 2014 arXiv:1411.4028
[23] Pepper A, Tischler N and Pryde G J 2019 *Phys. Rev. Lett.* **122** 060501
[24] Peruzzo A, McClean J, Shadbolt P, Yung M H, Zhou X Q, Love P J, Aspuru-Guzik A and O'brien J L 2014 *Nat. Commun.* **5** 4213
[25] Benedetti M, Lloyd E, Sack S and Fiorentini M 2019 *Quantum Sci. Technol.* **4** 043001
[26] Romero J and Aspuru-Guzik A 2019 arXiv:1901.00848
[27] Zoufal C, Lucchi A and Woerner S 2019 *npj Quantum Information* **5** 1–9
[28] Benedetti M, Grant E, Wossnig L and Severini S 2019 *New J. Phys.* **21** 043023
[29] Helstrom C W 1976 *Quantum Detection and Estimation Theory* vol 3 (New York: Academic)
[30] Holevo A S 1973 Statistical problems in quantum physics *Proc. of the Second Japan–USSR Symp. on Probability Theory* (Springer) 104–19
[31] Goodfellow I 2016 arXiv:1701.00160
[32] Mertikopoulos P, Papadimitriou C and Piliouras G 2018 Cycles in adversarial regularized learning *Proc. of the Twenty-Ninth Annual ACM-SIAM Symp. on Discrete Algorithms* (Philadelphia, PA: SIAM) 2703–17
[33] Flokas L, Vlatakis-Gkaragkounis E V and Piliouras G 2019 arXiv:1910.13010
[34] Daskalakis C, Ilyas A, Syrgkanis V and Zeng H 2017 arXiv:1711.00141
[35] Zhang G and Yu Y 2019 Convergence of gradient methods on bilinear zero-sum games *Int. Conf. on Learning Representations*
[36] Nielsen M A and Chuang I L 2010 *Quantum Computation and Quantum Information* (Cambridge: Cambridge University Press)
[37] Sim S, Johnson P D and Aspuru-Guzik A 2019 *Adv. Quantum Technol.* **2** 1900070
[38] Pérez-Salinas A, Cervera-Lierta A, Gil-Fuster E and Latorre J I 2020 *Quantum* **4** 226

[39] Schuld M, Sweke R and Meyer J J 2020 arXiv:2008.08605
[40] Shende V V, Markov I L and Bullock S S 2004 *Phys. Rev.* A **69** 062321
[41] Mitarai K, Negoro M, Kitagawa M and Fujii K 2018 *Phys. Rev.* A **98** 032309
[42] Schuld M, Bergholm V, Gogolin C, Izaac J and Killoran N 2019 *Phys. Rev.* A **99** 032331
[43] Banchi L and Crooks G E 2020 arXiv:2005.10299
[44] Kingma D P and Ba J 2014 arXiv:1412.6980
[45] Rakhlin A and Sridharan K 2013 Online learning with predictable sequences *Conf. on Learning Theory* 993–1019
[46] McClean J R, Boixo S, Smelyanskiy V N, Babbush R and Neven H 2018 *Nat. Commun.* **9** 1–6
[47] Banchi L, Pereira J, Lloyd S and Pirandola S 2020 *npj Quantum Information* **6** 1–10
[48] Lloyd S, Mohseni M and Rebentrost P 2014 *Nat. Phys.* **10** 631–3
[49] Lloyd S, Schuld M, Ijaz A, Izaac J and Killoran N 2020 arXiv:2001.03622
[50] Anand A, Romero J, Degroote M and Aspuru-Guzik A 2020 arXiv:2006.01976
[51] Kandala A, Mezzacapo A, Temme K, Takita M, Brink M, Chow J M and Gambetta J M 2017 *Nature* **549** 242–6
[52] Dong Y, Meng X, Lin L, Kosut R and Whaley K B 2019 arXiv:1911.00789
[53] Barkoutsos P K, Nannicini G, Robert A, Tavernelli I and Woerner S 2020 *Quantum* **4** 256
[54] Gentini L, Cuccoli A, Pirandola S, Verrucchi P and Banchi L 2020 *Phys. Rev.* A **102** 052414