Quantum Generative Adversarial Networks: Improving Dynamics Simulation with an Ancilla Qubit

Master's thesis, September 2025

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Physics

Guillermo Abad López

Supervisor:
Dr. Some Sankar Bhattacharya

Co-Supervisor:
Dr. Ayaka Usui

Acknowledgments

First, I would like to thank my supervisors, Dr. Ayaka Usui and Dr. Some Sankar Bhattacharya for their mentoring and guidance, and also Dr. Anna Sanpera for introducing me to this project, and team, which I ended up enjoying very much.

Thanks should also go to the all coworkers at GIQ that helped me, and made my days at GIQ very enjoyable.

Finally, I would also like to mention the support of my other co-workers at Qilimanjaro, which have endured a not very present version of me these past months, and have not complained one bit, even when they had reasons to.

Abstract

Simulating complex quantum systems remains a critical challenge, as conventional quantum techniques–such as those based on the Suzuki–Trotter decomposition—often result in deep circuits that demand substantial computational resources. Quantum Generative Adversarial Networks (QGANs) offer a promising alternative by learning the time evolution of target Hamiltonian using significantly fewer gates. However, standard QGAN architectures commonly suffer from unstable convergence and learning plateaus in the loss landscape, which hinder training and prevent the generator from achieving high-fidelity solutions.

To address these limitations, we propose augmenting the generator with an ancilla qubit, expanding the learning space, and providing additional degrees of freedom that enable training to progress when the model becomes trapped in certain regions of the loss landscape. In this work, we investigate the effect of incorporating an ancilla under various connectivity topologies and at different stages of training, in order to perturb the optimization landscape and aid the generator overcome problematic training cases

Simulation results demonstrate that ancilla-assisted QGANs successfully escape learning plateaus and other non-convergent behaviours, particularly when the ancilla's connectivity links distant regions of the ansatz. Notably, the optimized fidelity overall improves when the ancilla is introduced mid-way through the training.

Contents

1 Introduction

In recent years, quantum computing [1]–[3] has emerged and drawn considerable attention from researchers. This new computation paradigm exploits the properties of quantum mechanics to perform certain tasks substantially faster than classical computers [4], for example, simulating the dynamics of complex quantum systems, which requires computing the time evolution from its Hamiltonian complex interactions. This is inefficient on classical computers, because the degrees of freedom grow exponentially with system size. As Feynman said [1]; "Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical".

Despite this interest, simulating complex quantum systems remains a critical challenge. Conventional quantum techniques are based on the Suzuki-Trotter decomposition [5], [6], which breaks the system Hamiltonian H into its different terms applied individually several times, in short Trotter steps. In some cases, an accurate simulation demands many Trotter steps, inflating the gate counts and circuit depth, an acute limitation for the current *noisy intermediate-scale quantum* (NISQ) devices [7], [8], where gate errors and decoherence with time are non-negligible.

While several methods exist, we spotlight a way to *learn the time evolution* $U_H(t) = e^{-iHt}$ directly with Quantum Generative Adversarial Networks (QGANs) [9]–[11]. Instead of synthesizing $U_H(t)$ by long sequences of two-body gates, QGANs train a generator parametrized circuit to reproduce the effect of $U_H(t)$ while a discriminator tries to distinguish between them, driving both gradients. In practice, this can compress Hamiltonian evolutions into much shallower ansatzes. A striking example is shown in Ref. [12], where a 3-qubit Heisenberg time-evolution circuit that requires $\sim 1.19 \times 10^4$ gates by standard decomposition-based methods was well-approximated by 52 gates in a QGAN.

Despite these advantages, QGANs also inherit some variational circuits pathologies: unstable convergence, vanishing gradients, and extended *learning plateaus* where parameters stagnate. These effects are well documented in QGAN surveys [11], and are worsened by the expressivity–trainability trade-off: very deep models become susceptible to *learning plateaus* due to a large number of redundant parameters, but shallow models become untrainable and cannot reach high fidelities [13].

Another limitation of current QGAN schemes is that the generator has the same system size as the target Hamiltonian to learn, and therefore needs to explore its entire Hilbert space to find a single solution. However, it has been shown that two distinct Hamiltonians can generate identical dynamics in a smaller subspace [14]. Therefore, expanding the generator learning space, for example, with an ancilla qubit, leaves more room to learn the now reduced Hamiltonian time evolution.

This thesis, therefore, aims to improve the training of QGANs when learning the time evolution of target Hamiltonians by expanding the generator's space. More concretely, we ask whether a minimal architectural change to the generator—adding an *ancilla qubit*—can expand the learning space in a way to overcome *learning plateaus* and thereby improve convergence and final fidelity?

Organization of the thesis. The structure of the next chapters will be as follows: Chapter 2, reviews the necessary background: simulation of Hamiltonians [13], [15]; quantum machine learning[16], [17]; QGANs[9], [10], its known issues and some solutions [11]; like the quantum Wasserstein GANs (qWGANs) [12], and finally introduce the Choi states [18], [19] for learning unitaries. Chapter 3 then specifies the ancilla-assisted proposal from this thesis. Chapter 4 contains the details of how we implemented the QGANs training simulations and our ancilla proposal, and introduces the metrics we will use for the results evaluation. Chapter 5 presents the simulation results, showing how the ancilla does not improve fidelity when added from the start, whereas adding it to plateaus or other non-convergent cases helps training progress, while highlighting the importance of this ancilla connectivity and characteristics. Finally, Chapter 6 concludes our work, summarizing our findings, outlining the limitations of the study, as well as the future possible research directions initiated by this work.

2 Background

2.1 Simulating Complex Hamiltonian Dynamics

To simulate the dynamics of a quantum system, we must reproduce the time evolution $U_H(t) = e^{-iHt}$ generated by its Hamiltonian H. While classical computers can work with arbitrary operators, Hamiltonian time evolution included, their cost scales exponentially with system size (Hilbert-space dimension 2^n for n qubits), making dynamics beyond ∼50 components effectively intractable [15]. On the other hand, quantum devices can work with such Hilbert spaces in an efficient native way, since their minimal processing units, the qubits, are made to work and exploit the properties of quantum mechanics. Quantum hardware is advancing rapidly, with several companies claiming to be close to demonstrating practically useful speed-ups, for some very specific tasks [15], [20]–[23].

On quantum devices, especially in the NISQ era [7], [8], the available gates are typically limited to one- and two-qubit rotations, generated by Pauli matrices σ_i^k acting on qubit i,

$$\begin{aligned} &R_{X_i}(\theta) = e^{-i\sigma_i^x\theta/2}, \quad R_{Y_i}(\theta) = e^{-i\sigma_i^y\theta/2}, \quad R_{Z_i}(\theta) = e^{-i\sigma_i^z\theta/2} \\ &R_{X_iX_j}(\theta) = e^{-i\sigma_i^x\sigma_j^x\theta/2}, \quad R_{Y_iY_j}(\theta) = e^{-i\sigma_i^y\sigma_j^y\theta/2}, \quad R_{Z_iZ_j}(\theta) = e^{-i\sigma_i^z\sigma_j^z\theta/2}. \end{aligned} \tag{2.1}$$

Consider the dynamics of an N-qubit model with nearest-neighbor Ising and transverse-field terms,

$$U_{ZZ+X}(t) = e^{-iH_{ZZ+X}t} \quad \text{with} \quad H_{ZZ+X} = a\sum_{i=1}^{N-1}\sigma_i^z\sigma_{i+1}^z \; + \; b\sum_{i=1}^{N}\sigma_i^x \tag{2.2}$$

The parts $H_X = \sum_i^N \sigma_i^x$ and $H_{ZZ} = \sum_i^{N-1}\sigma_i^z\sigma_{i+1}^z$, each one by itself, decompose into parallel one- and two-qubit gates which can be directly applied,

$$\begin{aligned} U_X(t) = e^{-iH_Xt} &= \prod_{i=1}^{N} e^{-i\sigma_i^x t} = \prod_{i=1}^{N} R_{Xi}(2t) \ , \\ U_{ZZ}(t) = e^{-iH_{ZZ}t} &= \prod_{i=1}^{N-1} e^{-i\sigma_i^z\sigma_{i+1}^z t} = \prod_{i=1}^{N-1} R_{Z_iZ_{i+1}}(2t) \ . \end{aligned} \tag{2.3}$$

However, when such terms are combined, $e^{-it(A+B)} \neq e^{-itA}e^{-itB}$ for noncommuting A, B, and the Baker–Campbell–Hausdorff [24] or Zassenhaus [25] expansions produce an infinite series of nested commutators. A standard workaround is the Suzuki–Trotter product formula [5], [6],

$$e^{-i(A+B)t} = \lim_{n\to\infty}\left(e^{-iA\frac{t}{n}}e^{-iB\frac{t}{n}}\right)^n = e^{-iA\frac{t}{n}}e^{-iB\frac{t}{n}} \overset{(n)}{\cdots} e^{-iA\frac{t}{n}}e^{-iB\frac{t}{n}} + O(t^{n+1}), \tag{2.4}$$

where each repeated term is called a "Trotter step", containing the different terms for shorter times, yielding implementable gate sequences at the price of a circuit depth and gate count overhead, that grows with the required number n of Trotter steps, which for acceptable approximations in complex Hamiltonians can become large. Although several tailored variants exist, all share the same fundamental problem. For our $H_{ZZ+X} = aH_{ZZ} + bH_X$ example this gives

$$U_{ZZ+X}(t) = \lim_{n\to\infty}\left(e^{-iH_{ZZ}\frac{at}{n}}\, e^{-iH_X\frac{bt}{n}}\right)^n = \lim_{n\to\infty}\left(U_{ZZ}\left(at/n\right)\; U_X\left(bt/n\right)\right)^n, \tag{2.5}$$

i.e., many short "Trotter steps" alternating two-qubit and one-qubit layers. On NISQ devices, each gate incurs a fixed non-negligible noise independent of its effective simulation time; thus, large n accumulates errors and reduces fidelity. These limitations motivate the study of alternatives that avoid such deep decompositions for simulating complex Hamiltonians (systems).

One such approach is quantum generative adversarial networks (QGANs) [9]–[11]; a *quantum machine learning* (QML) framework, which can compress dynamics into much shallower ansatzes [12].

2.2 Quantum Machine Learning as a solution

QML studies how quantum devices can process and learn patterns from data more efficiently than classical models in specific tasks [16], [17]. In the near term, most practical approaches are *hybrid*: a parameterized quantum circuit (PQC) prepares a state; classical optimization updates circuit parameters based on a cost computed from quantum measurements. This exploits native quantum resources (superposition, entanglement, interference) while relying on mature classical optimizers [7], [8].

(i) *Parameterized ansatz.* PQCs are built from one- and two-qubit rotations arranged to respect hardware connectivity; problem-inspired or hardware-efficient ansatzes are common. (ii) *Objectives and gradients.* Costs are expectation values of observables or distances between states. Gradients are evaluated with parameter-shift rules [26], enabling end-to-end training on NISQ hardware.

As with classical deep models, PQCs face optimization challenges: noise, limited connectivity, and *barren plateaus* (vanishing gradients) can hinder convergence. Robust and smooth objectives that supply more informative gradients can mitigate this over long runs [8], [13].

2.3 QGANs: formulation and training

QGANs [9]–[11] are a two–player (adversarial) game [27] between a parameterized quantum circuit, the *generator* $G(\theta)$, and a parametrized measurement operator, the *discriminator* $D(\phi)$. The generator aims to prepare a state ψ_G/ρ_G that approximates a target state ψ_T/ρ_T produced by a fixed (or unknown) process T, while the discriminator attempts to tell them apart, providing learning signals for both. Using the "vanilla" total variation objective, leads to the minimax program [28]:

$$\min_{G/\theta} \max_{D/\phi} \quad \mathrm{Tr}[D_\phi(\rho_T - \rho_G)] = \mathbb{E}_{\rho_T}[D_\phi] - \mathbb{E}_{\rho_G}[D_\phi] \equiv \mathcal{L}(\rho_T, \rho_G, D_\phi) \tag{2.6}$$

where we abused the notation of $\mathbb{E}_\rho[D] := \mathrm{Tr}(D\rho)$, which refers to the expectation of the outcome of measuring operator D, on quantum state ρ. Training alternates updates of D/ϕ and G/θ, moving on the loss landscape $\mathcal{L}(\rho_T, \rho_G, D_\phi)$ via ascent and descent, respectively; Algorithm 1 outlines the steps. In idealized settings, this procedure targets a Nash equilibrium where the generator's state is indistinguishable from the target ($\mathcal{L} = 0$), with a learning that always converges. Intuitively, $G(\theta)$ increases $\mathbb{E}_{\rho_G}[D_\phi]$, i.e., steers ρ_G toward the discriminator's direction, while D_ϕ maximizes $\mathbb{E}_{\rho_T}[D_\phi] - \mathbb{E}_{\rho_G}[D_\phi]$, approximating the optimal discriminator between ρ_T and ρ_G, trying to return their trace distance $\mathrm{Tr}\,|\rho_T - \rho_G|$, which is equivalent to steering away from ρ_G, at the same time that tries approaching ρ_T. The process ends when both the generated state ρ_G and the discriminator D_ϕ converge into the target state ρ_T direction.

In practice, the generator parametrization and finite capacity make the optimization nontrivial, and runs may encounter local minima, barren plateaus, long oscillations, or other non-convergent behaviors that stagnate the training. Figure 2.1a shows an example of a basic QGAN scheme.

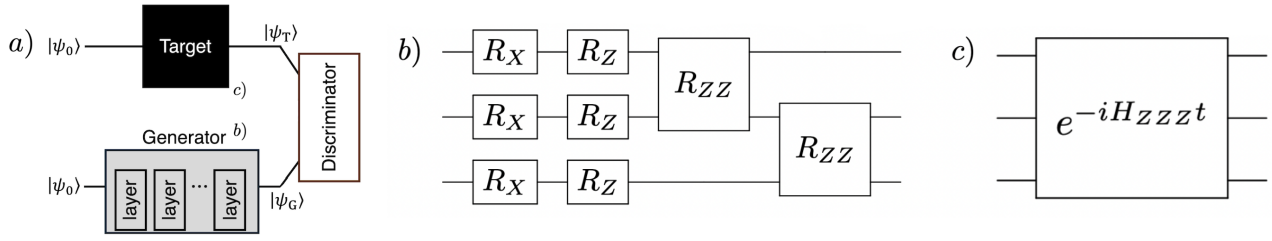

Figure 2.1: a) Basic QGAN scheme, depicting the target time evolution acting on the initial state, the generator, which is composed of repeated layers, also acting in the initial state, and the discriminator receiving the outputs of both. b) Example of 3-qubit, generator layer, with single-qubit rotations R_X and R_Z, followed by two-qubit rotations R_{ZZ}, each of them parametrized by an angle determining the magnitude of the rotation, and c) Time evolution of the target Hamiltonian H_{ZZZ}, where the magnitude is determined by t.

In dynamics simulation, both T and $G(\theta)$ are implemented as quantum circuits (Fig. 2.1b-c): $U_T(t) = e^{-iHt}$ realizes the time evolution of a target Hamiltonian H, and $U_G(\theta)$ is the generator ansatz. Given an initial state $|\psi_0\rangle$, they transform it into the following states:

$$\rho_T = |\psi_T\rangle\langle\psi_T| \text{ with } |\psi_T\rangle = U_T(t)|\psi_0\rangle, \quad \text{and} \quad \rho_G = |\psi_G\rangle\langle\psi_G| \text{ with } |\psi_G\rangle = U_G(\theta)|\psi_0\rangle, \tag{2.7}$$

to discriminate by measuring some observables on them, $\mathbb{E}_{\rho_T}[D_\phi]$ and $\mathbb{E}_{\rho_G}[D_\phi]$, producing the scalar loss $\mathcal{L}(\rho_T, \rho_G, D_\phi)$ used to update ϕ and θ via gradient methods. In practice, $U_G(\theta)$ is typically a shallow, hardware-efficient ansatz, and measurements are restricted to device-compatible forms.

Algorithm 1 Basic QGAN training pseudocode

Require: iterations $Iters$; dis steps n_{dis}; gen steps n_{gen}; fidelity threshold $F^\star$; time evolution t; target Hamiltonian T, generator ansatz G; discriminator measure basis D; initial state ψ_0; learnin rate η

Initialize random generator parameters θ (of the hardware-efficient ansatz)
Initialize random discriminator parameters ϕ (of the operator to measure)
Initialize target channel $U_T(t)$ (from the Hamiltonian time evolution)
Prepare generator state $\psi_G \leftarrow U_G(\theta)(\psi_0)$
Prepare target state $\psi_T \leftarrow U_T(t)(\psi_0)$
for $i \leftarrow 1$ **to** $Iters$ **do**
  for $c \leftarrow 1$ **to** n_{dis} **do** ▷ **Discriminator update(s)**
    Compute the Loss gradient: $\hat{\nabla}_\phi \mathcal{L}(\psi_G, \psi_T; D_\phi) \leftarrow$ Eq. (2.6)
    Update Discriminator going up the gradient $\phi \leftarrow \phi + \eta \hat{\nabla}_\phi \mathcal{L}$
  end for
  for $c \leftarrow 1$ **to** n_{gen} **do** ▷ **Generator update(s)**
    Compute the Loss gradient: $\hat{\nabla}_\theta \mathcal{L}(\psi_G, \psi_T; D_\phi) \leftarrow$ Eq. (2.6)
    Update Generator going down the gradient $\theta \leftarrow \theta - \eta \hat{\nabla}_\theta \mathcal{L}$
  end for
  Estimate fidelity F_i; **if** $F_i \geq F^\star$ **then break**
end for
return Final trained parameters and training curves: θ, ϕ, $\{F_j\}_{j=1}^{Iters}$

2.3.1 Training pathologies and mitigations

Although QGANs can achieve high fidelities, are theoretically very robust with ensured convergences, and enjoy bounded maximum loss (trace distance), which allows training the discriminator until optimality, adversarial training remains challenging. Typical pathologies include: (i) *vanishing gradients* [29] when one component (generated state, discriminator, or target) is far from the others; (ii) *mode collapse* [30], where G reproduces only a subset of the target modes, (iii) *non-convergence*/oscillations between the two players; (iv) the need for *common support* between ψ_T and ψ_G ($\langle\psi_T|\psi_G\rangle \neq 0$) to define meaningful distances; and (v) limited generator capacity [9], leading to local Nash equilibria far from the desired solution. These issues are compounded by noise, restricted connectivity, and *barren plateaus* arising from parameter redundancy in deep models [11], [13].

Mitigations include (i) well-conditioned objectives that yield informative gradients even when supports are disjoint or one component is weak, and (ii) gradient penalties [31] or Lipschitz constraints on the discriminator to ensure differentiability. Such techniques have been studied and applied to characterization of analog pulses, learning random distributions, and learning pure/mixed states [28], [32]–[41]. A particularly effective approach that includes both are quantum Wasserstein GANs (qWGANs).

2.3.2 Quantum Wasserstein GANs

Replacing total variation in Eq. (2.6) with the Earth-Mover/Wasserstein-1 distance [42] yields (i) smoother optimization with more stable, informative gradients; (ii) tolerance to disjoint supports; (iii)

empirically reduced mode collapse; and (iv) a non-negative objective that encourages the generated state ρ_G to directly reduce the distance to ρ_T, instead of only pursuing the discriminator D_ϕ. Moreover, qWGANs admit a dual formulation with efficiently measurable observables and a regularizer term ensuring differentiability, addressing many issues of the original QGANs [11], [12], [43].

The classical Wasserstein-1 loss encodes the minimum "cost" of transporting one distribution into another (amount of "dirt" moved, times the moving distance) [43], [44], the quantum version is more general, although the same intuition holds. Using it in our model, the objective becomes:

$$\min_{G/\theta} \min_{\pi \in \mathcal{H}_T \otimes \mathcal{H}_G} \mathrm{Tr}(\pi C) = \mathbb{E}_\pi[C] \qquad s.t.\ \ \mathrm{Tr}_{\mathcal{H}_T}(\pi) = \rho_G,\ \ \mathrm{Tr}_{\mathcal{H}_G}(\pi) = \rho_T \tag{2.8}$$

with a cost matrix C acting on $\mathcal{H}_T \otimes \mathcal{H}_G$. A common choice is $C = \frac{1}{2}(I \otimes I - \mathrm{SWAP})$, where SWAP is the swap operator that interchanges states: $\mathrm{SWAP}(\psi_x \otimes \psi_y) = \psi_y \otimes \psi_x$, which leads to a quantum semimetric $qW(\rho_T, \rho_G)$ [12] satisfying:

$$1.\ qW(\rho_T, \rho_G) \geq 0 \qquad 2.\ qW(\rho_T, \rho_G) = qW(\rho_G, \rho_T) \qquad 3.\ qW(\rho_T, \rho_G) = 0 \text{ iff } \rho_T = \rho_G. \tag{2.9}$$

In this view, training also alternates: the discriminator step seeks a joint state π that minimizes $\mathbb{E}_\pi[C]$. With $C = \frac{1}{2}(I \otimes I - \mathrm{SWAP})$, the objective is (up to constants) $-\mathrm{Tr}[\pi\,\mathrm{SWAP}]$, so the discriminator effectively searches for a *purification* of (ρ_T, ρ_G) within a larger joint state π whose two marginals resemble each other as much as possible (maximizing the SWAP overlap). The generator step then updates θ so that ρ_G approaches ρ_T while remaining consistent with the current joint state. Although this intuition is useful, directly optimizing over purifications is inconvenient to implement as a measurement-based discriminator. A more practical—and common—approach is the equivalent dual (min–max) formulation, which also resembles our previous QGAN expression more [12]:

$$\begin{aligned} \min_{G} \max_{\phi,\psi} \quad & \mathrm{Tr}(\rho_T\, D_\psi) - \mathrm{Tr}(\rho_G\, D_\phi) \equiv \mathbb{E}_{\rho_T}[D_\psi] - \mathbb{E}_{\rho_G}[D_\phi] \\ \text{s.t.} \quad & I_T \otimes D_\phi - D_\psi \otimes I_G \preceq C, \qquad D_\psi \in \mathcal{H}_T,\ D_\phi \in \mathcal{H}_G, \end{aligned} \tag{2.10}$$

where two measurement operators D_ψ (on the target space) and D_ϕ (on the generator space) are constrained jointly by C. This dual objective, although less intuitive, is "just" a difference of expectations, easy to implement, and the constraint ensuring differentiability.

Regularized qWGAN. The constraint can be enforced by a gradient penalty [31] via an additive regularizer ξ_R [12], yielding the final objective used in our study:

$$\begin{aligned} \min_{G} \max_{\phi,\psi} \quad & \mathbb{E}_{\rho_T}[D_\psi] - \mathbb{E}_{\rho_G}[D_\phi] - \mathbb{E}_{\rho_T \otimes \rho_G}[\xi_R] \ \equiv\ \mathcal{L}_W(\rho_T, \rho_G, D_\psi, D_\phi) \\ \text{with} \quad & \xi_R = \frac{\lambda}{e} \exp\left(\frac{-C - I_T \otimes D_\phi + D_\psi \otimes I_G}{\lambda}\right) \approx e^{\frac{-C}{2\lambda}} e^{\frac{-D_\psi}{\lambda}} \otimes e^{\frac{D_\phi}{\lambda}} e^{\frac{-C}{2\lambda}} \end{aligned} \tag{2.11}$$

where in the last step, we have slightly relaxed the regularization, for an easier implementation [12].

Direct estimation of gradients. To be efficient in implementing this loss function, we can directly implement its gradient in the QGAN process. To do so, first we show precisely how our components depend on the optimization parameters, splitting most of them into tensors of Pauli matrices:

$$\rho_0 = |\psi_0\rangle \langle\psi_0|, \qquad \rho_G = U_G(\theta)\rho_0 U_G^\dagger(\theta), \qquad U_G(\theta) = \prod_j e^{i\theta_j g_j/2} \tag{2.12}$$

$$D_\phi = \sum_k \alpha_k A_k, \qquad D_\psi = \sum_l \beta_l B_l, \qquad \xi_R = \sum_q r_q R_q \tag{2.13}$$

where g_j, A_k and B_l are Pauli matrices, and R_q is a combination of SWAPs and/or Pauli matrices [12]. The parameters of the generator are all the gates θ_j's (with a quantity given by the number of

layers times gates per layer) and the parameters of the discriminator α_k and β_l are the coefficients defining each measurement basis (four each, one for each possible Pauli matrix). Leaving the directly applicable gradients of the loss function $\mathcal{L}_W$ in Eq. (2.11), respect the optimization parameters, as:

$$\frac{\partial\,\mathcal{L}_W}{\partial\alpha_k} = \mathrm{Tr}[\rho_G A_k] - \mathrm{Tr}\left[(\rho_g \otimes \rho_T)\frac{(I_T \otimes A_k)R}{\lambda}\right] \tag{2.14}$$

$$\frac{\partial\,\mathcal{L}_W}{\partial\beta_l} = \mathrm{Tr}[\rho_T B_l] - \mathrm{Tr}\left[(\rho_g \otimes \rho_T)\frac{(B_l \otimes I_G)R}{\lambda}\right] \tag{2.15}$$

$$\frac{\partial\,\mathcal{L}_W}{\partial\theta_j} = \frac{\partial\,\mathrm{Tr}\left[D_\phi U_G(\theta)\rho_0 U_G^\dagger(\theta)\right]}{\partial\theta_j} - \frac{\partial\,\mathrm{Tr}\left[\xi_R(U_G(\theta)\rho_0 U_G^\dagger(\theta)\otimes\rho_T)\right]}{\partial\theta_j} \tag{2.16}$$

where the pending gates derivatives can be done with well-known techniques, like parameter-shift [26].

2.3.3 Simulating the full unitary via Choi states

Training on a single initial state $|\psi_0\rangle$ learns only one trajectory of the entire unitary evolution. To learn the *entire* unitary channel U_T, one may augment the initial state (e.g., noise, batch sampling) [45], but such schemes do not guarantee full-channel learning. A principled alternative employs the Choi–Jamiołkowski isomorphism [18], [19]. We consider the maximally entangled state

$$|\Omega\rangle = \frac{1}{\sqrt{d}}\sum_{i=0}^{d-1}|i\rangle\,|i\rangle\,, \qquad d = 2^n, \tag{2.17}$$

and construct the Choi state, which encodes the channel effect on all elements of a complete basis, therefore containing the same information as the full unitary (as its Choi matrix):

$$|\Phi_T\rangle = (I\otimes U_T)\,|\Omega\rangle = \frac{1}{\sqrt{d}}\sum_{i=0}^{d-1}|i\rangle\otimes U_T\,|i\rangle\,, \qquad |\Phi_G\rangle = (I\otimes U_G)\,|\Omega\rangle = \frac{1}{\sqrt{d}}\sum_{i=0}^{d-1}|i\rangle\otimes U_G\,|i\rangle\,.$$

Where the overlap between these Choi states equals the (squared, normalized) process fidelity:

$$F_{\text{proc}} = \left|\langle\Omega|\,(I\otimes U_G^\dagger U_T)\,|\Omega\rangle\right|^2 = \frac{1}{d^2}\left|\sum_{i,j=0}^{d-1}\langle i|j\rangle\,\langle i|U_G^\dagger U_T|j\rangle\right|^2 = \frac{1}{d^2}\left|\mathrm{Tr}[U_G^\dagger U_T]\right|^2,$$

which equals 1 iff U_G and U_T differ by a global phase. Thus training on Choi states is equivalent to learning the full unitary channel, at the cost of doubling the qubit count. This approach is natural for Hamiltonian simulation and has been successfully used in qWGANs for circuit approximation [12]. Figure 2.2b-c sketches the standard and Choi-state QGAN set-up used here, and then introduces this thesis proposal, adding an ancilla to aid the generator, with what we continue in the next chapter.

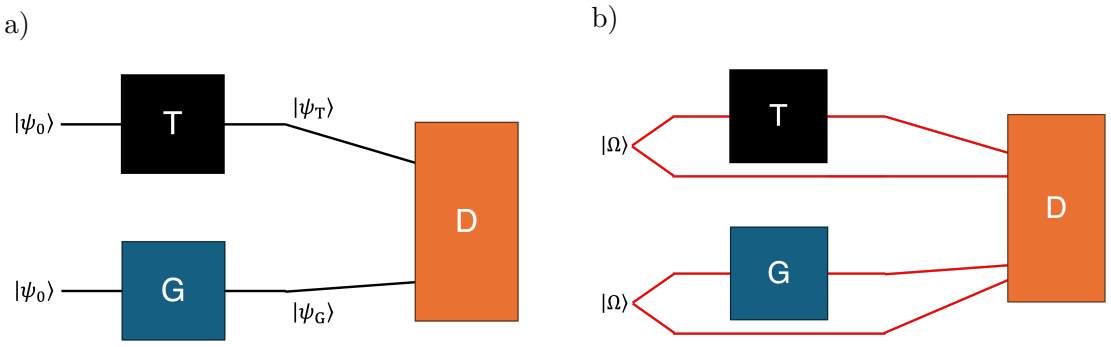

Figure 2.2: a) Standard QGAN, which trains to simulate the target state $|\psi_T\rangle$ (time evolution applied to a single initial state $|\psi_0\rangle$), with the generator's state $|\psi_G\rangle$, and b) QGAN's Choi state ($(U\otimes I)\,|\Omega\rangle$) setup, which learns the entire time-evolution unitary, where red represents lines that are originally entangled.

3 Ancilla-assisted QGANs

3.1 Motivation

Although qWGANs alleviate many issues of the original QGAN formulation, several pathologies persist in practice—most notably *learning plateaus*, *oscillatory behaviours*, and *limited generator capacity*.

Standard QGAN generators typically match the system size of the target and must explore the full system Hilbert space to recover a single solution. Yet distinct Hamiltonians can induce identical dynamics within a restricted subspace [14]. This suggests that extending the generator's Hilbert space with an *ancilla* qubit can introduce degrees of freedom that ease optimization while ultimately reproducing the same system-level unitary. Related ideas—introducing auxiliary degrees of freedom—are common in quantum simulation and algorithm design (e.g., gadget construction [46]).

Motivated by these observations, we propose augmenting the QGAN generator with a single ancilla. Intuitively, the ancilla (i) enlarges the trainable space and (ii) can *bridge* distant regions of a hardware-efficient ansatz, perturbing otherwise flat loss landscapes and helping resume progress from plateaus (Fig. 3.1). We study several ancilla–system connectivity patterns of varying complexity and assess *when* to introduce the ancilla (from the start, after detecting stagnation, or at a fixed mid-training iteration), with an eye toward near-term realizations on NISQ devices with constrained connectivity (e.g., heavy-hex, square lattices) [7], [8].

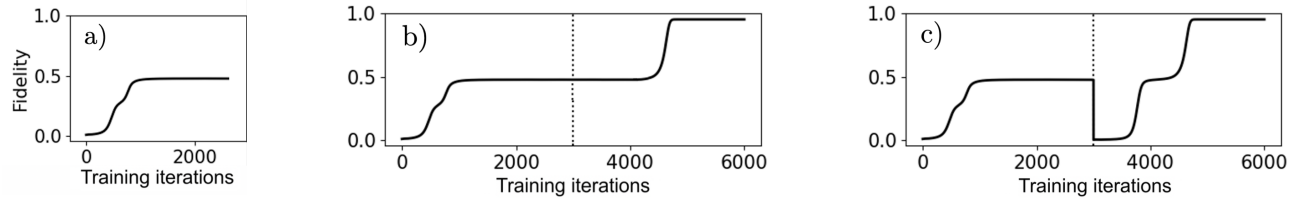

Figure 3.1: a) Example of fidelity versus iterations in a stuck learning plateau, b) example of escaping a plateau, by adding an ancilla qubit with starting gate angles set to 0, where the fidelity changes slowly, and c) example of escaping a plateau by adding an ancilla with randomized gate angles, where at first instance the fidelity drops, since the ancilla gates, has moved from its initial state, to which is being compared against.

3.2 Model

Let S denote the n-qubit system (so $\dim \mathcal{H}_S = N$) and let A be a single-qubit ancilla initialized in $|0\rangle_A$. The *target* channel U_T acts unitarily on S, while the *generator with ancilla*, $U_{GA}(\theta)$, acts unitarily on $S \otimes A$ instead, with its *effective* action on the system S alone, obtained by tracing out A, being described by a completely positive and trace-preserving (CPTP) map $\mathcal{E}_G$, rather than a unitary:

$$\mathcal{E}_G(\,\cdot\,) = \mathrm{Tr}_A\Big[U_{GA}\Big((\,\cdot\,) \otimes |0\rangle\langle 0|_A\Big)U_{GA}^\dagger\Big] = \tilde{U}_G(\cdot)\tilde{U}_G^\dagger + \tilde{U}_{mix}(\cdot)\tilde{U}_{mix}^\dagger. \tag{3.1}$$

where $\tilde{U}_G = \langle 0|\, U_{GA}\, |0\rangle_A$ is the subspace block in which the ancilla starts and remains in $|0\rangle_A$, and $\tilde{U}_{mix} = \langle 1|\, U_{GA}\, |0\rangle_A$ collects the off-diagonal block (leakage), where the ancilla transitions to $|1\rangle_A$.

Proposal: Ancilla through generator, compared at discriminator. We propose adding an ancilla to the generator, initialized at $|0\rangle_A$, coupling it to the original system qubits, via (one- and) two-qubit gates appended after each original layer, yielding enlarged layers for U_{GA}. Then pass the coupled generator output state $|\psi_{GA}\rangle$, to the discriminator, where it will be compared against the original target state and an idle reference ancilla accompanying it, originating from $U_T \otimes I_A$ (Fig. 3.2b). Training thus encourages the generator's ancilla to *disentangle*, while driving $|\psi_{GA}\rangle$ toward $\sim |\psi_T\rangle \otimes |0\rangle_A$,

steering with it $\tilde{U}_G$ and $\tilde{U}_{mix}$ towards U_T and 0 respectively (Fig. 3.2a). In this way, $\mathcal{E}_G$ can initially exploit a larger non-unitary parameter manifold, while training guides it towards disentangling the ancilla and reducing the CPTP map back to a unitary that implements U_T in S.

To formalize this, consider the Choi-style initial state (Section 2.3.3), augmented with an extra ancilla:

$$|\Omega'\rangle = \frac{1}{\sqrt{d}} \sum_{i=0}^{d-1} |i\rangle\,|i\rangle\,|0\rangle_A\,. \tag{3.2}$$

Let T' (resp. G') denote the Choi copy space for the target (resp. generator) half. The fidelity between the target state $(I_A \otimes U_T \otimes I_{T'})\,|\Omega'\rangle$ and the generator state $(U_{GA} \otimes I_{G'})\,|\Omega'\rangle$ is

$$\left|\langle\Omega'|\,(U_{GA}\otimes I_{G'})^\dagger (I_A\otimes U_T\otimes I_{T'})\,|\Omega'\rangle\right|^2 = \frac{1}{d^2}\left|\sum_{i,j=0}^{d-1} \langle i|j\rangle\,\langle i,0|\,U_{GA}^\dagger (I_A\otimes U_T)\,|j,0\rangle\right|^2 = \tag{3.3}$$

$$= \frac{1}{d^2}\left|\sum_{i=0}^{d-1} \langle i,0|\,U_{GA}^\dagger (I_A\otimes U_T)\,|i,0\rangle\right|^2 = \frac{1}{d^2}\left|\sum_{i=0}^{d-1} \langle i|\,\tilde{U}_G^\dagger U_T\,|i\rangle\right|^2 = \frac{1}{d^2}\left|\mathrm{Tr}[\tilde{U}_G^\dagger U_T]\right|^2,$$

This fidelity equals 1 iff $\tilde{U}_G = e^{i\varphi} U_T$ for some global phase φ, and there is no leakage, i.e., $\tilde{U}_{mix} = 0$. Equivalently, U_{GA} is block-diagonal in the ancilla basis with the $|0\rangle\langle 0|_A$ block equal (up to a phase) to U_T (Fig. 3.2a). Consequently, our proposal leverages the enlarged parameter space of the generator–ancilla unitary U_{GA} early in training, but then is explicitly driven towards returning a disentangled $|0\rangle_A$ ancilla, and correctly simulating the desired U_T in the corresponding system S.

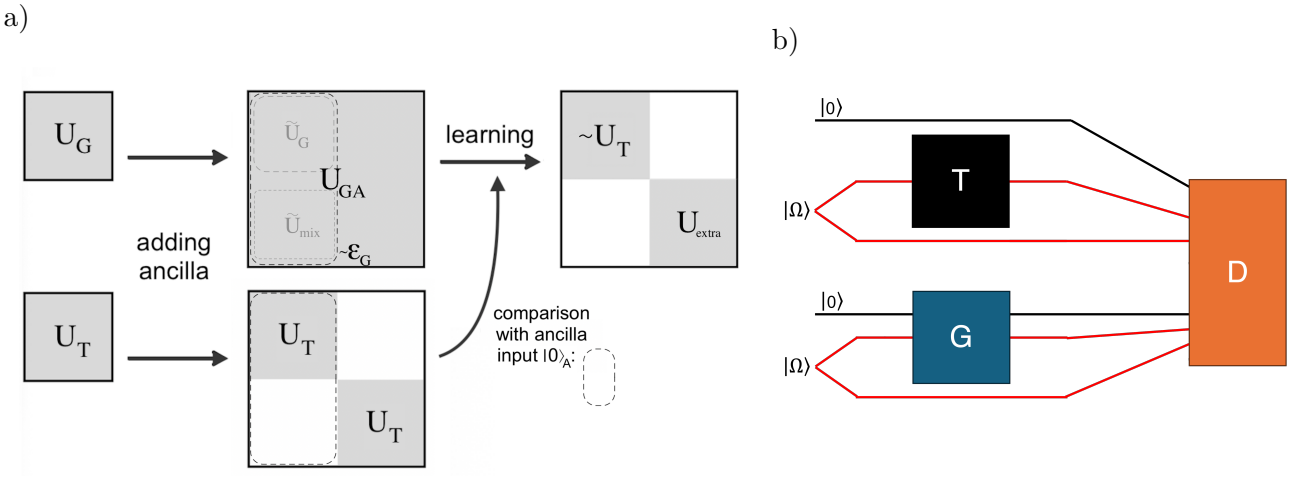

Figure 3.2: a) Schematic learning process: the enlarged generator U_{GA} is driven so that its $|0\rangle\langle 0|_A$ subspace matches U_T and its off-diagonal (leakage) terms vanish (white marks the zero-valued blocks); a residual non-trained unitary U_{extra} remains in the $|1\rangle\langle 1|_A$ subspace. b) Setup sketch, adding the ancilla on top of the Choi state: the ancilla traverses only the generator, while the target carries an idle reference ancilla; then both outputs are compared at the discriminator.

3.3 Connectivity, NISQ considerations and costs

We evaluate several ancilla–system couplings, together with other characteristics, such as whether its angles should be randomized at the start, if it should contain one-qubit gates as well, etc. Some of those connectivities will provide non-local shortcuts in the ansatz original connectivity, which should aid plateaus escape. The goal would be to achieve that with a connectivity that at the same time, is efficiently implementable on common NISQ topologies (e.g., heavy-hex), where topologies are not only a line, but have qubits with > 2 connections every certain qubits. Resource-wise, adding one qubit doubles the system size and simulation cost, and also adds degrees of freedom, which might not always be positive, due to possible parameter redundancies in deep ansataz. So the *timing* of ancilla insertion (e.g., at start, only after stagnation is detected, or at a fixed mid-run iteration) is critical.

4 Methodology

4.1 Simulations

The simulations here are based on the Python implementation used in Ref. [12] for training qWGANs. The code has been fully refactored and tested to (i) support the addition of an ancilla qubit, (ii) generalize to arbitrary target Hamiltonians and generator ansatzes, and (iii) improve performance, organization, and data generation, so that larger-scale studies of ancilla insertion are feasible.

The generator $U_G(\theta)$ is treated as a parametrized quantum circuit (given an ansatz), whose parameters θ are updated during training, the target $U_T(t)$ is implemented as a fixed circuit obtained by exponentiating the specified Hamiltonian H; and the discriminator is represented by two measurement directions D_ϕ, D_ψ (Pauli-vector axes ϕ, ψ) used to measure the generated and target states respectively. We compute Eq. (2.14) explicit gradients of the dual quantum Wasserstein loss in Eq. (2.11), including the regularization term that ensures differentiability. All the matrix operations are implemented with `NumPy` [47] and `SciPy` [48]. An overview of the workflow appears in Algorithm 2.

Algorithm 2 qWGAN complete training with Choi states (parameters details on sec. 4.3)

Require: epochs *Epochs*, iterations per epoch *Iter*; dis steps n_{dis}; gen steps n_{gen}; fidelity threshold $F^\star$; time evolution t; target Hamiltonian H, generator ansatz G, generator layers n_G; discriminator measure basis D; learning rates η_G, η_D; momentum coeff β_G, β_D; log and plots path *dir*

Initialize random generator parameters θ (of the hardware-efficient ansatz)
Initialize random discriminators parameters ϕ, ψ (of the operators axis to measure)
Initialize generator and discriminator optimzers $v_G, v_{D,\phi}, v_{D,\psi} \leftarrow 0, 0, 0$
Prepare generator state $\psi_G \leftarrow$ Choi state of $(U_G(\theta) = \prod_j^{n_G} G_j(\theta))$
Prepare target state $\psi_T \leftarrow$ Choi state of $(U_T(t) = e^{-iHt})$
for $e \leftarrow 1$ **to** *Epochs* **do**
  for $i \leftarrow 1$ **to** *Iters* **do**
    for 1 **to** n_{dis} **do** ▷ **Discriminator update(s)**
      for Ω **in** ψ, ϕ **do** ▷ Update both measure axis
        Compute discriminator loss gradient: $\widehat{\nabla}_\Omega \mathcal{L}_W(\psi_T, \psi_G; D_\psi, D_\phi) \leftarrow$ Eq. (2.14)
        Compute and add regularizer gradient: $\widehat{\nabla}_\Omega \mathcal{L}'_W \leftarrow \widehat{\nabla}_\Omega \mathcal{L}_W + \widehat{\nabla}_\Omega \xi_r$
        Update optimizer momentum: $v_{D,\Omega} \leftarrow \beta_D v_{D,\Omega} + (1-\beta_D)\widehat{\nabla}_\Omega \mathcal{L}'_W$
        Update discriminator parameters: $\Omega \leftarrow \Omega - \eta_D v_{D,\Omega}$
      end for
      Regenerate discriminator operators D_ψ, D_ϕ
    end for
    for 1 **to** n_{gen} **do** ▷ **Generator update(s)**
      Compute generator loss gradient: $\widehat{\nabla}_\Omega \mathcal{L}_W(\psi_T, \psi_G; D_\psi, D_\phi) \leftarrow$ Eq. (2.16)
      Compute and add regularizer gradient: $\widehat{\nabla}_\theta \mathcal{L}'_W \leftarrow \widehat{\nabla}_\theta \mathcal{L}_W + \widehat{\nabla}_\theta \xi_r$
      Update optimizer momentum: $v_G \leftarrow \beta_G v_G + (1-\beta_G)\widehat{\nabla}_\theta \mathcal{L}'_W$
      Update generator parameters: $\theta \leftarrow \theta - \eta_G v_G$
      Regenerate generator state: $\psi_G \leftarrow$ Choi state of $U_G(\theta)$
    end for
    Compute and store fidelity and cost: $F'_{e\cdot Iters+i}, \mathcal{L}'_{W\,e\cdot Iters+i} \leftarrow F, \mathcal{L}_W$
  end for
  Plot current training curves $(dir, \{F'_j, \mathcal{L}'_{W\,j}\}_{j=1}^{e\cdot Iters})$ **if** $F \geq F^\star$ **then break**
end for
return Final trained parameters and training curves: θ , ψ, ϕ, $\{F_j, \mathcal{L}_{W\,j}\}_{j=1}^{Epochs\cdot Iters}$

4.2 Metrics to evaluate results

In this section, we introduce the three main metrics that we will use during Chapter 5, for analyzing the simulation results. Then we evaluate and show how the metrics perform.

The first metric to introduce, from where we will build the other two, is the best fidelity (F_{best}), which gives us information about a single training repetition, considering its fidelity $F = |\langle\psi_G|\psi_T\rangle|^2$ at each i iteration F_i, and returns the maximum one at any point. The second metric is the average best fidelity ($F_{avg,best}$), for when we have run multiple training repetitions (e.g., N), for which we just average the $F_{best,j}$ of each repetition j. Finally, the last one, the success rate (S_{rate}), basically counts the number of runs that have passed a certain fidelity threshold (F^*):

$$F_{best} = \max_i(F_i) \ , \qquad F_{avg,best} = \frac{1}{N}\sum_{j=0}^{N} F_{best,j} \ , \qquad S_{rate} = \operatorname*{count}_j(F_{best,j} > F^*). \tag{4.1}$$

It is important to recognize that the last metric (success rate) works as a binary classification (passes the fidelity threshold or no), where all unsuccessful cases contribute equally, no matter the best fidelity they have reached (e.g., a 10% best fidelity, counts equally towards the success rate, as a 95% one, since none passed the 99% threshold), so we must exercise caution in its use, as it can be misleading, returning very low vales, in cases, where almost all the results are just below the threshold (e.g., a case where 100 repetitions end up with a best fidelity of 98%, would return a success rate of 0%, and another case, where 95 repetitions end up with a best fidelity of 20%, but 5 have a 99%, would return a higher success rate). It is also sensitive to statistical deviations; since for cases where only a few successful cases are obtained, it can fluctuate significantly (a few repetitions passing the threshold, versus not passing it, change its value a lot). The second metric, the average best fidelity, doesn't have these problems, since each repetition contributes with its weight, making it a more robust indicator.

To see if these metrics correctly reflect how good or bad a training has been, let us check if the training, and therefore the metrics we defined, get better results the longer (more epochs) we train. In Fig. 4.1, we train a qWGAN against the ZZZ target Hamiltonian, defined in Eq. (4.2), for different numbers of epochs, seeing that the average best fidelity ($F_{avg,best}$) increases with them, and even recognizing how it gets bounded towards the end. Finally, mention how the success rate (S_{rate}) becomes non-zero and increases with the epochs, having a noticeable jump at 20 epochs (of 300 iterations each).

a) b)

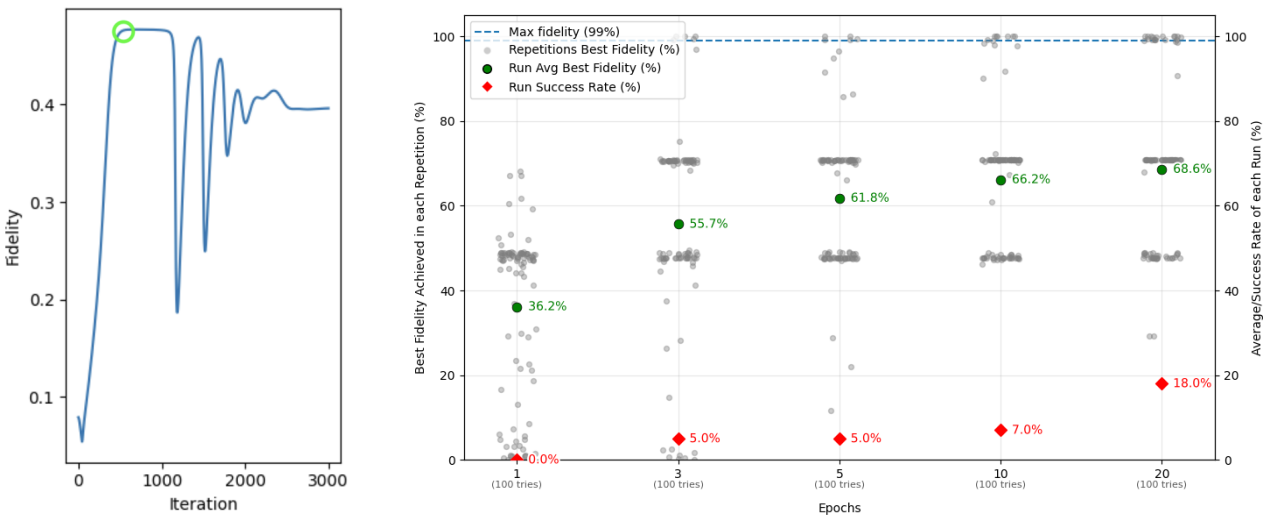

Figure 4.1: a) Evolution of the fidelity through time in a training instance. The green circle indicates the value of the F_{best}, b) Metrics in the case where H_{ZZZ} is targeted for different values of training epochs. Showing in gray dots, precisely the F_{best} for each of the repetitions. Then in green and red, the $F_{avg,best}$ and S_{rate} respectively.

4.3 Simulation Parameters

We now summarize the key simulation parameters used for the results in Chapter 5, what they mean, their default values (unless stated otherwise), and where those values came from:

Training parameters:

- *epochs* (*Epochs*): number of training epochs (default: 20).
- *iterations_epoch* (*Iters*): iterations per epoch (default: 300).
- *max_fidelity* (F^*): early-stop fidelity threshold (default: 0.99).
- *steps_gen/dis* (n): discriminator/generator updates per iteration (default: 1).
- *system_size* (N): number of system qubits, excluding Choi state and ancilla (default: 3).

Generator/Target parameters

- *gen_layers* (n_G): number of generator-ansatz layers (default: 3).
- *gen_ansatz* (G): generator ansatz unitary (default: nearest-neighbour $G_{ZZ,X,Z}(\theta)$).
- *target_hamiltonian* (H): target Hamiltonian terms (default: three-body $H_{ZZZ} \equiv ZZZ$).
- *target_evolution_time* (t): time to evolve with the target Hamiltonian (default: 1).

Optimizer parameters

- *l_rate* (η): coefficient controlling how much the gradient modifies our parameters (default: 0.01).
- *momentum_coeff* (β): coefficient about resistance to change learning directions (default: 0.9).

For our study, the generator ansatz $G_{ZZ,X,Z}$, and the Hamiltonian ZZZ (represented in Figure 2.1) dynamics are given by:

$$\begin{aligned} U_G(\theta) &= \prod_j^{layers} G_{ZZ,X,Z}(\theta) = \prod_j^{layers} \left(\prod_i^{N-1} R_{Z_i Z_{i+1}}(\theta) \prod_i^{N} R_Z(\theta)_i \prod_i^{N} R_X(\theta)_i \right) \\ U_T(t) &= e^{-iZZZt} = e^{-i\left(\sum_i^{N-2} \sigma_{z_i}\sigma_{z_{i+1}}\sigma_{z_{i+2}}\right)t} = \prod_i^{N-2} R_{Z_i Z_{i+1} Z_{i+2}}(2t). \end{aligned} \tag{4.2}$$

Therefore, unless stated otherwise, our simulations will use a fixed-number of total training iterations (6000), training once the discriminator and the generator in each, with an early stop criterion at target fidelity > 0.99, focusing primarily on learning dynamics generated by a three-body ZZZ target Hamiltonian using a two-body hardware-efficient ansatz shown in Fig. 2.1, with three layers.

The previous default values, were choosen after evaluating the defined metrics ($F_{avg,best}, S_{rate}$) against sweeps of each of the parameters. (i) Increasing epochs improves both metrics, with a marked gain between 10 and 20 epochs; corresponding to simulation times which start to be at the limit of what allows us to collect statistically significant data, we therefore set *epochs* $= 20$. (ii) Varying the generator/discriminator step ratio (correcting the iterations, for comparing the same number of updates) produced minor differences, so we use the 1–1 setting of Ref. [12]. (iii) Each generator layer added increases the average best fidelity ($F_{avg,best}$) by 6% approximately, and doubles the success rate (S_{rate}), as seen in Fig. 4.2, but it also increases the training time; therefore we use three layers where the success rate (S_{rate}) already is non-zero. (iv) Learning-rate and momentum sweeps of the optimizer showed only mild improvements towards larger learning rates; we therefore retain the values of Ref. [12]. Under these settings and on a 2024 Mac mini with an M4 Pro chip and 64GB of RAM, each training run (repetition) required approximately 7 minutes with the ancilla enabled and about 3 minutes without it, with the results from the thesis taking around 800 hours of computational time, split across several computers.

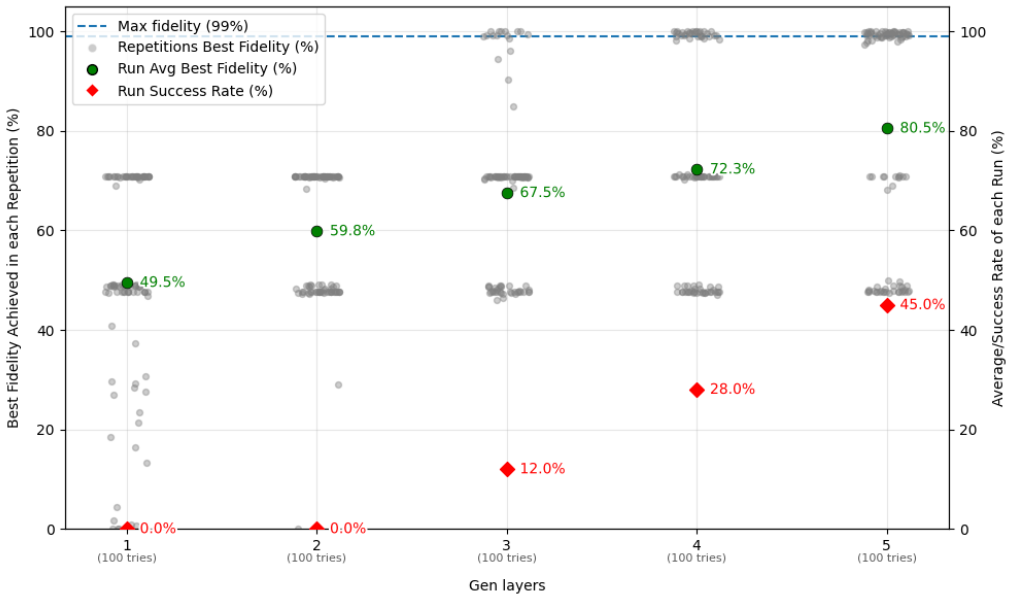

Figure 4.2: Results of QGAn training against the ZZZ target Hamiltonian, for different numbers of layers in the generator. Showing in gray dots the best fidelity (F_{best}) achieved for each repetition (100 in each Hamiltonian), in green the average ($F_{avg,best}$) of each case, and in red the success rate (S_{rate}).

4.4 Adding the ancilla, methodology

To add an ancilla, we extend the generator circuit with one additional qubit and couple it to the system according to the chosen topology (Fig. 4.3), leaving existing parameters unchanged. The discriminator is extended by two qubits: one to receive the generator's ancilla output and one reference qubit (kept idle alongside the target) so that the discriminator can compare the ancilla's final state to its initial state, encouraging the ancilla to return to its input state while the system learns U_T autonomously. Upon ancilla insertion, discriminator parameters are reinitialized.

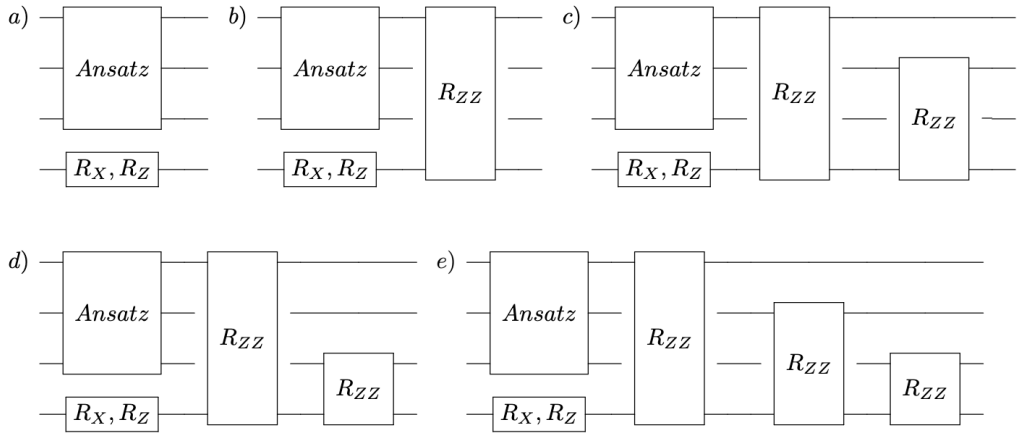

Figure 4.3: New layers in the generator, after adding the different ancilla topologies that will be employed during the Section 5. a) "Disconnected" or "Diconn.", where the ancilla is not coupled anywhere, b) "Ansatz" where the ancilla connects to a single qubit, following the ansatz pattern, c) "Short Bridge", where the ancilla connects to the first and second qubits, d) "Bridge", where the ancilla connects the first and last qubit, and e) "Total", where the ancilla connects to all the rest of qubits.

Besides the topologies defined here (Fig. 4.3), we will also use "NoRand" and "No1Q" variations, which mean, respectively, to start with the ancilla gates angles at zero (not randomized), and for the ancilla to not contain 1-qubit gates, only the 2-qubit connections.

5 Results

This chapter reports simulation results obtained by systematically varying (i) the ancilla–system connectivity (topology) and (ii) the timing of ancilla insertion. Unless otherwise noted, experiments use a fixed training budget of 6000 iterations with an early stop at target fidelity $F > 0.99$. The default setting learns the dynamics of a three-body ZZZ target Hamiltonian with a three-layer, two-local hardware-efficient generator ansatz. We will start analyzing how the different ancilla topologies help when added from the start of the training. Then we move to study whether adding the ancilla in problematic cases that do not converge in the designated number of iterations, helps them converge. Finally, given the two previous results we move into studying the addition of the ancilla midway through the training at a fixed moment, and see if this strategy helps us achieve better performance, with the metrics average best fidelitiy ($F_{avg,best}$) and success rate (S_{rate}), described in Eq. (4.1).

5.1 Ancilla present from the start

We first assess adding a single ancilla qubit from the beginning of training, giving the extra degrees of freedom directly from the start. For targets of the form $H_{aXZX+bZZ} \equiv aXZX + bZZ$:

$$aXZX + bZZ \equiv a \sum_{i}^{N-2} \sigma_i^x \sigma_{i+1}^z \sigma_{i+2}^x \; + \; b \sum_{i}^{N-1} \sigma_i^z \sigma_{i+1}^z, \; \rightarrow \; U_T(t) = e^{-i(aXZX+bZZ)t} \; ,$$

we compare "bridge" and "total" ancilla topologies (Fig. 4.3d-e). As shown in Fig. 5.1, adding the ancilla from the outset does not improve performance and can degrade it for simpler targets.

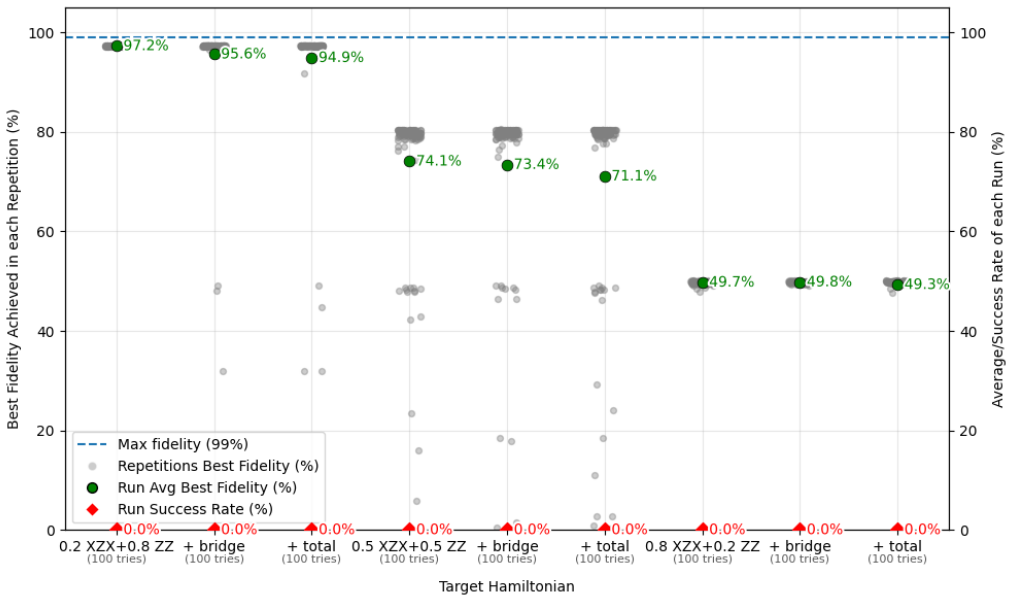

Figure 5.1: Results of adding the ancilla qubit in "bridge" and "total" topology to three target Hamiltonians of the form $a\ XZX + b\ ZZ$, for different a, b's. Showing in gray dots the best fidelity (F_{best}) achieved for each repetition (100 in each Hamiltonian), and in green the average ($F_{avg,best}$) of all those repetitions for each Hamiltonian.

We then fix the target to ZZZ and survey a broader set of topologies. Figure 5.2 shows that average gains are small; the best cases ("bridge"/"total") yield only a modest increase in average best fidelity ($F_{avg,best}$) (about +3.5%), while changes in success rate (S_{rate}) are noisy owing to threshold effects when the number of successful runs is small. This lack of improvements could be explained by

the fact that, even though in principle expanding the parameters can be used for escaping plateaus, when unnecessary, it also might make the appearances of plateaus easier due to parameter redundancy. Overall, early inclusion of the ancilla adds extra parameters without clear benefit, thereby increasing resource demands (simulation time and qubit count) with limited payoff.

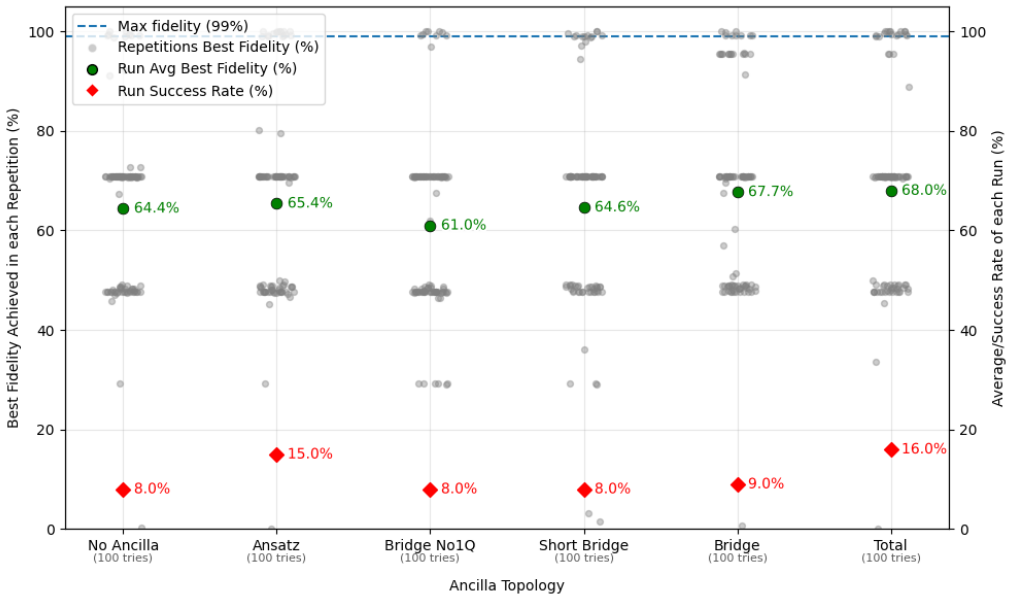

Figure 5.2: Results of adding the ancilla qubit in distinct topologies to the ZZZ target Hamiltonians. Showing in gray dots the best fidelity (F_{best}) achieved in each repetition (100 for each case), in green the average ($F_{avg,best}$) of all the repetitions, and in red the success rate (S_{rate}, fid>0.99) of each ancilla topology.

5.2 Ancilla insertion in non-convergent runs

We next introduce the ancilla only in difficult runs that fail to reach $F > 0.99$ within the budget, not penalizing (adding redundancy) the ones that will be successful already, and using fewer resources at the same time. Two distinct failure cases are considered: (i) clear learning plateaus (flat fidelity traces) and (ii) other non-convergent (e.g., oscillatory) behaviour. For each problematic case, we insert an ancilla with randomized angles and repeat 15 times per instance to estimate variability.

5.2.1 Escaping learning plateaus

Twelve plateaued trajectories (examples in Fig. 5.3) were used as seeds.

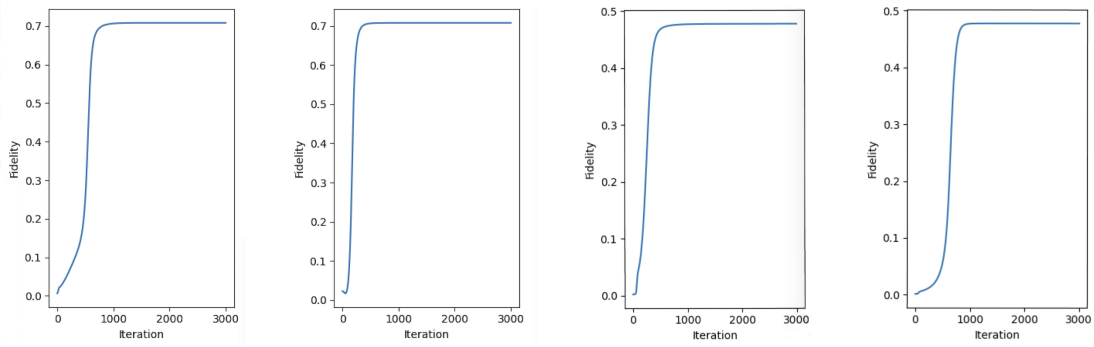

Figure 5.3: Examples of learning plateaus after 3000 total iterations.

As summarized in Fig. 5.4, simple couplings (such as; “disconected”, “disconnected No1Q”, “ansatz”,

"bridge no1Q" and "short bridge") do not lift the plateau; by contrast, topologies that connect distant parts of the ansatz ("bridge" and "total") raise the *average best fidelity* ($F_{avg,best}$) by approximately +6%, and only the "total" topology also increases the success rate (S_{rate}) by about +13%. These improvements also outperform a trivial discard-and-restart strategy tried, where we start from scratch with a new training, which only achieves a 65.7% average best fidelity and a 10.7% success rate.

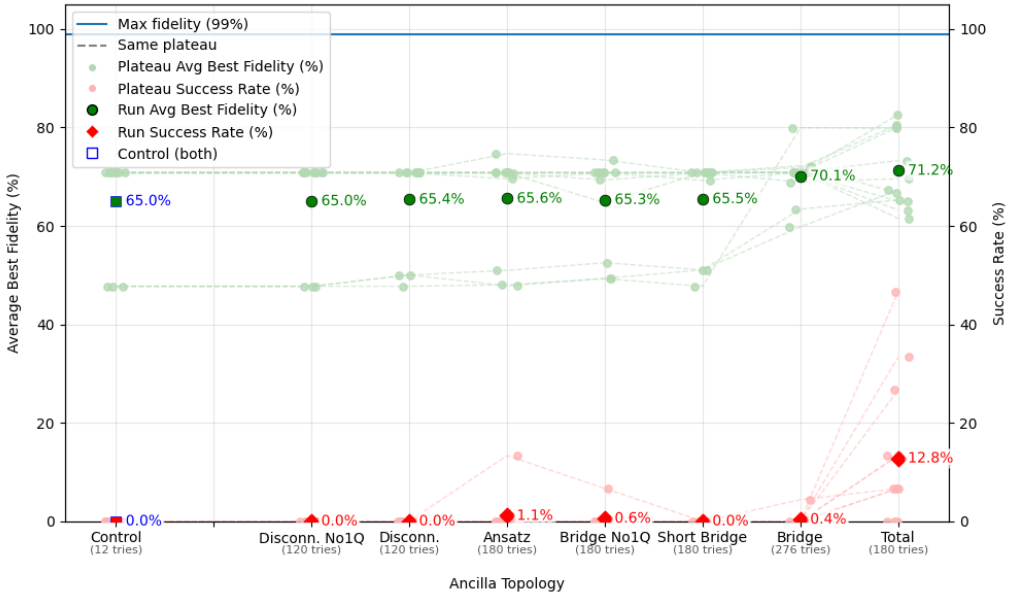

Figure 5.4: Results of adding the ancilla qubit, for twelve stuck plateaus, in distinct topologies to the ZZZ target Hamiltonian. The light green dots indicate the average best fidelity for each plateau (> 10 repetitions in each), the darker green dots show the total average of each ancilla topology, the light red dots show the average success rate (fid> 0.99) of each plateau, and the dark red dots show the total success rate of each ancilla topology. The control case, where nothing is changed in the run, is indicated in blue.

Detailed views of the two best performing plateaus, which both start near 47.7% fidelity, show that, after "bridge" insertion, runs typically jump to a small set of higher plateaus, improving average best fidelity ($F_{avg,best}$) by roughly +18.5% and +30%, respectively (Fig. 5.5).

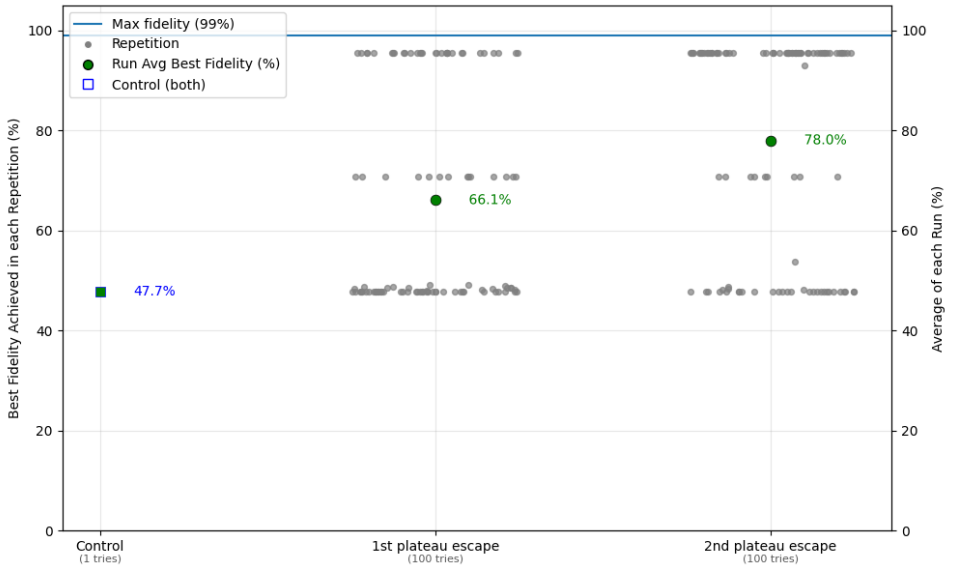

Figure 5.5: Two trapped plateaus at 47.7% fidelity escape when adding a randomly initialized ancilla qubit in "bridge" topology, showing in gray the best fidelity F_{best} in each repetition (100 each), and in green the average $F_{avg,best}$.

5.2.2 Escaping other non-convergent runs

We repeat the protocol on eight other non-convergent cases, but which have not plateaued (e.g., oscillatory loss/fidelity; Fig. 5.6).

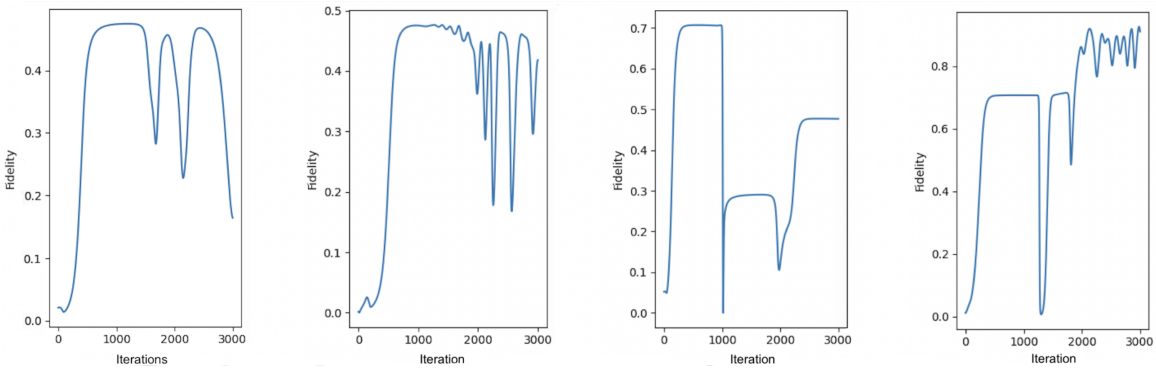

Figure 5.6: Example of non-convergent trainings after 3000 total iterations, which are not learning plateaus.

Results in Fig. 5.7 mirror the plateau findings but with larger margins: "bridge"/"total" topologies yield an average best fidelity ($F_{avg,best}$) gain of about +16%. Success-rate (S_{rate}) is higher in general due to some controls eventually converging with additional iterations, but the "total" topology still improves over the baseline (+6.7%). These improvements again outperform the same trivial discard-and-restart strategy from before, but now by a much bigger margin. These results indicate that the ancilla, rather than just helping in the plateau stuck cases, also aids oscillations and restores progress in otherwise unstable trainings.

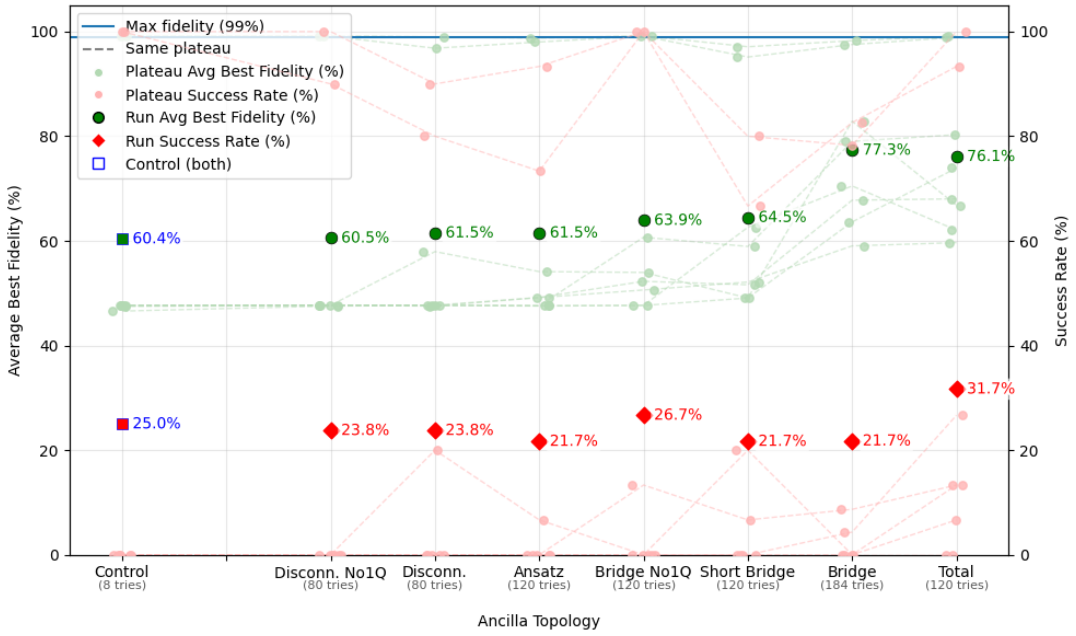

Figure 5.7: Results of adding the ancilla qubit, for eight non-convergent non-plateaud cases, in distinct topologies to the ZZZ target Hamiltonian. Showing in gray dots the best fidelity (F_{best}) achieved for all the repetitions (> 100 for each case), in green the average ($F_{avg,best}$) of all those repetitions for each ancilla topology, in red the success rate (S_{rate}, fid> 0.99) of each ancilla topology. Finally, in blue, the control case where nothing is changed to the run, and it just continues to run for the same epochs as the rest.

5.3 Midrun ancilla insertion

The ancilla does not help when added from the start, but it does help when manually added to the non-convergent cases (stuck, oscillations, etc). Here, we consider adding the ancilla at a fixed moment mid-run, when the cases that converge fast won't get (negatively) affected, but the non-convergent cases remain, and can be aided.

We evaluate the following unconditional protocol: insert automatically the ancilla at the midpoint of training, without manually checking its state. As shown in Fig. 5.8, mid-run insertions of the ancilla in "bridge" and "total" topologies, with randomized gate angles, improves both the average best fidelity ($F_{avg,best}$) and the success rate (S_{rate}) for the ZZZ target. Notably, the "bridge" topology already offers better results, while remaining practical on common NISQ devices layouts (e.g., heavy-hex), avoiding the gates overhead that a fully connected ancilla would induce in the logical-to-physical qubit mapping and routing [49], [50] quantum devices need before execution, to fit more complex quantum circuits into their limited connectivities. Conversely, initializing ancilla gates at zero ("NoRand") yields little to no benefit.

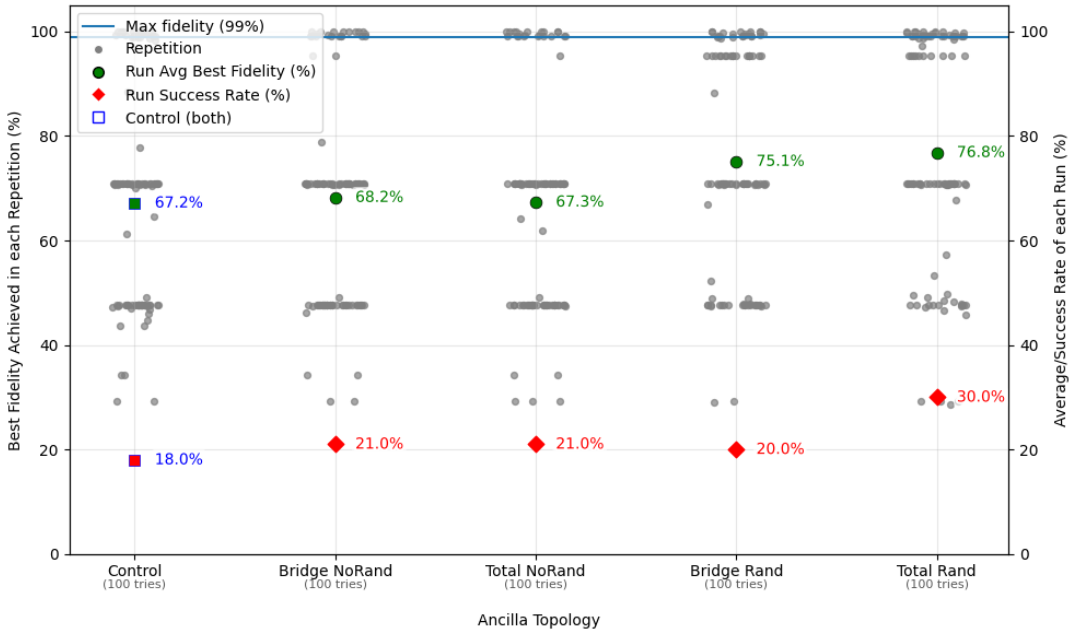

Figure 5.8: Results of adding the ancilla qubit mid-run, for a ZZZ target Hamiltonian, and different ancilla topologies. Two cases start with ancilla qubit gates angle set to zero ("NoRand") and the other two initialize them randomly ("Rand"). In gray we show the best fidelity (F_{best}) reached in each repetition (100 each case), in green the average ($F_{avg,best}$) of each case, and in red its success rate (S_{rate}).

6 Discussion

This thesis investigated whether *ancilla-assisted* QGANs can improve the learning of Hamiltonian dynamics compared with standard QGAN baselines. Previously, it has been shown that QGANs, and, in particular, Wasserstein QGANs (qWGANs), can *learn* to simulate a target nontrivial Hamiltonian time-evolution with far fewer gates than Suzuki-Trotter decompositions based methods, often resulting in substantial circuit overhead and resource demands that are unacceptable for NISQ devices. At the same time, the QGAN literature highlights practical training issues, including instability and plateaus [11], which motivate architectural innovations.

Therefore, we asked: can a minimal architectural change, adding a single *ancilla* qubit to the QGAN, expand the learning space sufficiently to overcome *learning plateaus* and thereby improve convergence and final fidelity when learning the time evolution generated by a target Hamiltonian?

Short answer: yes. The ancilla helps primarily in runs that show non-convergent behaviours (e.g., plateaus, oscillations), especially when inserted midway through training and connected in a *"bridge"* or *"total"* topology with its own single-qubit rotations and a randomized initialization of its angles; adding it from the start yields, at best, marginal average gains.

In our ancilla-assisted QGAN study, we systematically varied (i) ancilla connectivity topologies (e.g., *bridge* connections that link two distant regions of the ansatz, or *total* connections that link everything), and (ii) *when* the ancilla is introduced (from the start, or only after a non-convergent case has been identified, or at a fixed mid-training point), to study the learning of the full dynamics generated by three-body target Hamiltonians via their Choi state, employing a parameterized general-axis expectation-value as discriminator, and using three layers of two-body hardware-efficient ansatzes as a generator, both of which we added the ancilla when required.

6.1 Conclusions

We observed that the ancilla-assisted variants consistently helped in non-convergent runs, improving convergence and final fidelity. Importantly, they outperformed a naïve discard-and-restart strategy, which discards runs and restarts from scratch:

- **Escaping learning plateaus.** The ancilla-assisted variant reliably helped escape plateaus, resuming progress to higher fidelities, confirming that the ancilla beneficially perturbs the landscape, although escaped runs sometimes became trapped in subsequent plateaus.
- **Stabilizing non-convergent runs.** In runs characterized by non-convergent behaviours of the loss/fidelity, like oscillations, or up and down steps, etc, inserting the ancilla damped these dynamics, and led to more straightforward convergences and higher fidelities.

Another important finding is that not all the ways to use the ancilla helped equally (or at all), and since the ancilla can be a costly resource, doubling simulation time and memory usage, and consuming scarce qubits on NISQ devices, we should be mindful about how and when to use it:

- **Connectivity matters.** When the ancilla *connects* otherwise distant regions of the hardware-efficient generator ansatz (*bridge* and *total* topologies), training results improved; for simpler topologies, they did not. This fact indicates that the ancilla qubit acts as a mediator, allowing distant qubits to communicate in fewer steps than through the original hardware-efficient ansatz. Interestingly, in the end, the ancilla must return to its initial state (to achieve a fidelity ~ 1), so it cannot have a direct effect on the final achieved generator, only as an intermediary of the other original qubits during the intermediate training steps. This effect is expected to be more influential for larger systems, where the separation of qubits at the hardware-efficient ansatzes

is larger, and the bridge structure may be more influential. Notably, the simple *bridge* topology, which can be efficiently implemented in many NISQ devices with limited connectivity (e.g., heavy-hex), we found helps considerably. Thus, it is more efficient than using the more complex "total" connectivity, which might add a big overhead of SWAP gates [49], [50].

- **Shape and initialization matters.** Other conditions were also necessary to see improvements, specifically, the ancilla required its own single-qubit gates in addition to the two-qubit couplings. Moreover, a zero-angle initialization of its gates did not help, whereas a randomized one did.

- **Timing matters.** Adding the ancilla from the beginning produced only small average gains; by contrast, introducing it midway through training consistently improved the average best fidelity and success rate. We conjecture that this works because it aids the non-convergent cases (plateaus or odd oscillations) that remain training, while not affecting the ones that converge fast by themselves before the addition.

Taken together, these results suggest that training with the baseline generator first, and once stagnation is detected, or after a preset fraction of the total iterations, *injecting* an ancilla with a *total/bridge* topology (*bridge* for a more NISQ-friendly approach), with single-qubit gates and randomized angles, improves the training of QGANs on Hamiltonian dynamics in problematic cases. This strategy is simple to implement, compatible with standard QGANs, and specifically targets the empirically challenging regimes (plateaus and non-convergent oscillations).

Viewed more broadly, ancilla-assisted QGANs provide a simple, low-cost lever to reshape the optimization landscape in a favorable way, without the need to redesign the entire model. Acting as a training *catalyst* for the difficult cases when learning complex Hamiltonian dynamics.

6.2 Limitations of the study

Our conclusions must be interpreted in light of the study's scope:

- **Size limitations.** Experiments were only conducted on small systems (e.g., three-qubit targets). Although we expect that an ancilla joining even further sections in bigger systems should perform better, the actual scaling behaviour for larger Hilbert spaces, more complex Hamiltonians (k-body with $k \geq 4$), and deeper ansatzes remains to be tested.

- **Limited targets and generators.** We have only worked with 6 different target Hamiltonians, from which only the ZZZ case was studied extensively. We have limited ourselves to a single ansatz for the generator. Under this limitation, it remains to be seen whether the conclusions drawn from the present simulation results can be generalized to different Hamiltonian systems.

- **Fixed training parameters.** After analysing the parameters for shallow circuits, we fixed the total training iterations, the fidelity stopping criterion, the type of optimizer, and its parameter values. With improved computational power in the future, it remains to be tested whether it works equally well, worse, or better for deeper circuits.

- **Fixed loss function.** We have to consider that we only used a single loss function during all the results analysis, the Wasserstein distance of order 1. This choice is based on the knowledge of QGAN pathologies from previous studies. It is not known whether our results may differ for other loss functions introduced to address other problems of QGAN, which may handle multi-qubit distances differently.

6.3 Further Developments

In light of the previous limitations, several improvements are natural:

- **Improve code efficiency.** Develop a more efficient code implementation for matrix operations and gradient descent, to handle larger system sizes, memory, and speed-wise. This would enable studying how much the ancilla help scales with the system size (qubit count), and the target complexity (k-body terms with larger k). It is interesting, since we expect that an ancilla connecting further regions of the ansatz, would get better results. This would also allow us to broaden the sets of models/configurations to study in a realistic timeframe, partially solving the first three limitations of the study. We are already working on migrating the code to use PennyLane [51], for this reason.
- **Hyper-tune the optimizer.** Explore alternative optimizers, hyperparameter tuning, and adaptive schedulers to reduce the number of iterations required for high fidelity, further sharpening comparisons and accelerating data generation across models.

Other directions that arise directly from this work include:

- **Optimal ancilla insertion.** Finding the optimal ancilla insertion time (e.g., at a different percentage of the total training iterations, or triggered by a slope threshold that detects stagnation), is a natural continuation of the final obtained results. Since we only tried at exactly the middle of the run, and already got an improvement.
- **Comparison against other techniques.** Another extension would be to benchmark the gains against other classical or quantum heuristics, such as random parameter perturbations when stuck, multiple parallel starts, or improved initial ansatz connectivity.
- **Study plateau distributions.** For some targets (e.g., the ZZZ Hamiltonian), stuck runs clustered around discrete fidelities (e.g., 47.7%, 72%, 95%), whereas the Ising case showed a more homogeneous pattern. Understanding why these structures arise may be key to further improving final fidelities.

Finally, another potentially impactful direction is:

- **Projecting the ancilla qubit.** Post-selecting on an ancilla measurement implements a non-unitary map while learning a unitary evolution. This enlarges the effective solution space and can ease convergence, at the cost of a success probability $p < 1$ when deploying the learned generator (the ancilla outcome certifies whether a run "succeeds"). Interleaving projections during training (a Zeno-like guidance) could also constrain learning to favourable subspaces. In addition, excluding the ancilla from the discriminator's input reduces its dimension, which in our implementation nearly halves the training runtime.

Code is publicly available on: https://github.com/ayaka-usui/qgan_subspace

References

[1] R. P. Feynman, "Simulating physics with computers," *International Journal of Theoretical Physics*, vol. 21, no. 6-7, pp. 467–488, 1982. DOI: 10.1007/BF02650179.

[2] D. D., "Quantum theory, the church-turing principle and the universal quantum computer," *Proceedings of the Royal Society of London A*, vol. 400, pp. 97–117, 1985.

[3] S. Lloyd, "Universal quantum simulators," *Science*, vol. 273, no. 5278, pp. 1073–1078, Aug. 23, 1996. DOI: 10.1126/science.273.5278.1073.

[4] A. W. Harrow and A. Montanaro, "Quantum computational supremacy," *Nature*, vol. 549, no. 7671, pp. 203–209, Sep. 2017, ISSN: 1476-4687. DOI: 10.1038/nature23458. [Online]. Available: http://dx.doi.org/10.1038/nature23458.

[5] H. F. Trotter, "On the product of semi-groups of operators," *Proceedings of the American Mathematical Society*, vol. 10, no. 4, pp. 545–551, 1959. DOI: 10.1090/S0002-9939-1959-0108732-6.

[6] M. Suzuki, "Generalized trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems," *Communications in Mathematical Physics*, vol. 51, no. 2, pp. 183–190, 1976. DOI: 10.1007/BF01609348.

[7] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum*, vol. 2, p. 79, 2018. DOI: 10.22331/q-2018-08-06-79. arXiv: 1801.00862 [quant-ph].

[8] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, *et al.*, "Noisy intermediate-scale quantum (nisq) algorithms," *Reviews of Modern Physics*, vol. 94, no. 1, p. 015 004, Feb. 15, 2022. DOI: 10.1103/RevModPhys.94.015004. arXiv: 2101.08448 [quant-ph].

[9] S. Lloyd and C. Weedbrook, "Quantum generative adversarial learning," *Phys. Rev. Lett.*, vol. 121, p. 040 502, 4 Jul. 2018. DOI: 10.1103/PhysRevLett.121.040502. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.121.040502.

[10] P.-L. Dallaire-Demers and N. Killoran, "Quantum generative adversarial networks," *Phys. Rev. A*, vol. 98, p. 012 324, 1 Jul. 2018. DOI: 10.1103/PhysRevA.98.012324. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.98.012324.

[11] T. A. Ngo, T. Nguyen, and T. C. Thang, "A survey of recent advances in quantum generative adversarial networks," *Electronics*, vol. 12, no. 4, 2023, ISSN: 2079-9292. DOI: 10.3390/electronics12040856. [Online]. Available: https://www.mdpi.com/2079-9292/12/4/856.

[12] S. Chakrabarti, Y. Huang, T. Li, S. Feizi, and X. Wu, *Quantum wasserstein generative adversarial networks*, 2019. arXiv: 1911.00111 [quant-ph]. [Online]. Available: https://arxiv.org/abs/1911.00111.

[13] E. R. Anschuetz and B. T. Kiani, "Quantum variational algorithms are swamped with traps," *Nature Communications*, vol. 13, no. 1, Dec. 2022, ISSN: 2041-1723. DOI: 10.1038/s41467-022-35364-5. [Online]. Available: http://dx.doi.org/10.1038/s41467-022-35364-5.

[14] K. Gietka, A. Usui, J. Deng, and T. Busch, "Simulating the same physics with two distinct hamiltonians," *Physical Review Letters*, vol. 126, no. 16, Apr. 2021, ISSN: 1079-7114. DOI: 10.1103/physrevlett.126.160402. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.126.160402.

[15] A. J. Daley, I. Bloch, C. Kokail, *et al.*, "Practical quantum advantage in quantum simulation," *Nature*, vol. 607, pp. 667–676, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:251132664.

[16] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum algorithms for supervised and unsupervised machine learning," *arXiv*, 2013. DOI: 10.48550/arXiv.1307.0411. arXiv: 1307.0411 [quant-ph].

[17] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, pp. 195–202, Sep. 14, 2017. DOI: 10.1038/nature23474.

[18] A. Jamiołkowski, "Linear transformations which preserve trace and positive semidefiniteness of operators," *Reports on Mathematical Physics*, vol. 3, no. 4, pp. 275–278, 1972. DOI: 10.1016/0034-4877(72)90011-0.

[19] M.-D. Choi, "Completely positive linear maps on complex matrices," *Linear Algebra and its Applications*, vol. 10, no. 3, pp. 285–290, 1975. DOI: 10.1016/0024-3795(75)90075-0.

[20] F. Arute, K. Arya, R. Babbush, *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, pp. 505–510, 2019. [Online]. Available: https://www.nature.com/articles/s41586-019-1666-5.

[21] O. Lanes, M. Beji, A. D. Corcoles, *et al.*, *A framework for quantum advantage*, 2025. arXiv: 2506.20658 [quant-ph]. [Online]. Available: https://arxiv.org/abs/2506.20658.

[22] H.-Y. Huang, M. Broughton, J. Cotler, *et al.*, "Quantum advantage in learning from experiments," *Science*, vol. 376, no. 6598, pp. 1182–1186, Jun. 2022, ISSN: 1095-9203. DOI: 10.1126/science.abn7293. [Online]. Available: http://dx.doi.org/10.1126/science.abn7293.

[23] A. D. King, A. Nocera, M. M. Rams, *et al.*, *Computational supremacy in quantum simulation*, 2024. arXiv: 2403.00910 [quant-ph]. [Online]. Available: https://arxiv.org/abs/2403.00910.

[24] H. F., "Die symbolische exponentialformel in der gruppentheorie," *Ber. Verh. Kgl. SÃ¤chs. Ges. Wiss. Leipzig., Math.-phys. Kl. 58 (1906) pp.19-48*, 1906.

[25] Z. H., "Über lie'sche ringe mit primzahlcharakteristik," *Abh.Math.Semin.Univ.Hambg. 13, 1–100*, 1939. [Online]. Available: https://doi.org/10.1007/BF02940753.

[26] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, "Evaluating analytic gradients on quantum hardware," *Physical Review A*, vol. 99, no. 3, Mar. 2019, ISSN: 2469-9934. DOI: 10.1103/physreva.99.032331. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.99.032331.

[27] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020, ISSN: 0001-0782. DOI: 10.1145/3422622. [Online]. Available: https://doi.org/10.1145/3422622.

[28] L. Kim, S. Lloyd, and M. Marvian, "Hamiltonian quantum generative adversarial networks," *Phys. Rev. Res.*, vol. 6, p. 033 019, 3 Jul. 2024. DOI: 10.1103/PhysRevResearch.6.033019. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevResearch.6.033019.

[29] M. Arjovsky and L. Bottou, *Towards principled methods for training generative adversarial networks*, 2017. arXiv: 1701.04862 [stat.ML]. [Online]. Available: https://arxiv.org/abs/1701.04862.

[30] Y. Kossale, M. Airaj, and A. Darouichi, "Mode collapse in generative adversarial networks: An overview," in *2022 8th International Conference on Optimization and Applications (ICOA)*, 2022, pp. 1–6. DOI: 10.1109/ICOA55659.2022.9934291.

[31] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, *Improved training of wasserstein gans*, 2017. arXiv: 1704.00028 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1704.00028.

[32] K. Borras, S. Y. Chang, L. Funcke, *et al.*, "Impact of quantum noise on the training of quantum generative adversarial networks," *Journal of Physics: Conference Series*, vol. 2438, no. 1, p. 012 093, Feb. 2023, ISSN: 1742-6596. DOI: 10.1088/1742-6596/2438/1/012093. [Online]. Available: http://dx.doi.org/10.1088/1742-6596/2438/1/012093.

[33] S. Nokhwal, S. Nokhwal, S. Pahune, and A. Chaudhary, *Quantum generative adversarial networks: Bridging classical and quantum realms*, 2023. arXiv: 2312.09939 [quant-ph]. [Online]. Available: https://arxiv.org/abs/2312.09939.

[34] B. T. Kiani, G. De Palma, M. Marvian, Z.-W. Liu, and S. Lloyd, "Learning quantum data with the quantum earth mover's distance," *Quantum Science and Technology*, vol. 7, no. 4, p. 045 002, Jul. 2022, ISSN: 2058-9565. DOI: 10.1088/2058-9565/ac79c9. [Online]. Available: http://dx.doi.org/10.1088/2058-9565/ac79c9.

[35] L. Hu, S.-H. Wu, W. Cai, *et al.*, "Quantum generative adversarial learning in a superconducting quantum circuit," *Science Advances*, vol. 5, no. 1, Jan. 2019, ISSN: 2375-2548. DOI: 10.1126/sciadv.aav2761. [Online]. Available: http://dx.doi.org/10.1126/sciadv.aav2761.

[36] K. Huang, Z.-A. Wang, C. Song, *et al.*, "Quantum generative adversarial networks with multiple superconducting qubits," *npj Quantum Information*, vol. 7, no. 1, 165, p. 165, Dec. 2021. DOI: 10.1038/s41534-021-00503-1.

[37] S. Ahmed, C. Sánchez Muñoz, F. Nori, and A. F. Kockum, "Quantum state tomography with conditional generative adversarial networks," *Physical Review Letters*, vol. 127, no. 14, Sep. 2021, ISSN: 1079-7114. DOI: 10.1103/physrevlett.127.140502. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.127.140502.

[38] C. Zoufal, A. Lucchi, and S. Woerner, "Quantum generative adversarial networks for learning and loading random distributions," *npj Quantum Information*, vol. 5, no. 1, Nov. 2019, ISSN: 2056-6387. DOI: 10.1038/s41534-019-0223-2. [Online]. Available: http://dx.doi.org/10.1038/s41534-019-0223-2.

[39] M. Benedetti, E. Grant, L. Wossnig, and S. Severini, "Adversarial quantum circuit learning for pure state approximation," *New Journal of Physics*, vol. 21, no. 4, p. 043 023, Apr. 2019, ISSN: 1367-2630. DOI: 10.1088/1367-2630/ab14b5. [Online]. Available: http://dx.doi.org/10.1088/1367-2630/ab14b5.

[40] S. A. Stein, B. Baheri, D. Chen, *et al.*, "Qugan: A quantum state fidelity based generative adversarial network," in *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, IEEE, Oct. 2021, pp. 71–81. DOI: 10.1109/qce52317.2021.00023. [Online]. Available: http://dx.doi.org/10.1109/QCE52317.2021.00023.

[41] M. Y. Niu, A. Zlokapa, M. Broughton, *et al.*, *Entangling quantum generative adversarial networks*, 2021. arXiv: 2105.00080 [quant-ph]. [Online]. Available: https://arxiv.org/abs/2105.00080.

[42] G. De Palma, M. Marvian, D. Trevisan, and S. Lloyd, "The quantum wasserstein distance of order 1," *IEEE Transactions on Information Theory*, vol. 67, no. 10, pp. 6627–6643, Oct. 2021, ISSN: 1557-9654. DOI: 10.1109/tit.2021.3076442. [Online]. Available: http://dx.doi.org/10.1109/TIT.2021.3076442.

[43] L. Weng, *From gan to wgan*, 2019. arXiv: 1904.08994 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1904.08994.

[44] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML'17, Sydney, NSW, Australia: JMLR.org, 2017, pp. 214–223.

[45] H.-L. Huang, Y. Du, M. Gong, *et al.*, "Experimental quantum generative adversarial networks for image generation," *Physical Review Applied*, vol. 16, no. 2, Aug. 2021, ISSN: 2331-7019. DOI: 10.1103/physrevapplied.16.024051. [Online]. Available: http://dx.doi.org/10.1103/PhysRevApplied.16.024051.

[46] D. Harley, I. Datta, F. R. Klausen, *et al.*, "Going beyond gadgets: The importance of scalability for analogue quantum simulators," *Nature Communications*, vol. 15, no. 1, Aug. 2024, ISSN: 2041-1723. DOI: 10.1038/s41467-024-50744-9. [Online]. Available: http://dx.doi.org/10.1038/s41467-024-50744-9.

[47] C. R. Harris, K. J. Millman, S. J. van der Walt, *et al.*, "Array programming with numpy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020, ISSN: 1476-4687. DOI: 10.1038/s41586-020-2649-2. [Online]. Available: http://dx.doi.org/10.1038/s41586-020-2649-2.

[48] P. Virtanen, R. Gommers, T. E. Oliphant, *et al.*, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: 10.1038/s41592-019-0686-2.

[49] A. Zulehner, A. Paler, and R. Wille, *An efficient methodology for mapping quantum circuits to the ibm qx architectures*, 2018. arXiv: 1712.04722 [quant-ph]. [Online]. Available: https://arxiv.org/abs/1712.04722.

[50] G. Li, Y. Ding, and Y. Xie, *Tackling the qubit mapping problem for nisq-era quantum devices*, 2019. arXiv: 1809.02573 [cs.ET]. [Online]. Available: https://arxiv.org/abs/1809.02573.

[51] V. Bergholm, J. Izaac, M. Schuld, *et al.*, *Pennylane: Automatic differentiation of hybrid quantum-classical computations*, 2022. arXiv: 1811.04968 [quant-ph]. [Online]. Available: https://arxiv.org/abs/1811.04968.