

Latent Style-based Quantum GAN for high-quality Image Generation

Su Yeon Chang^{1,2,*} Supanut Thanasilp^{3,4} Bertrand Le Saux,⁵ Sofia Vallecorsa,¹ and Michele Grossi^{1,†}

¹European Organization for Nuclear Research (CERN), Geneva, Switzerland

²Laboratory of Theoretical Physics of Nanosystems (LTPN),

Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

³Institute of Physics, Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland

⁴Chula Intelligent and Complex Systems, Department of Physics,
Faculty of Science, Chulalongkorn University, Bangkok, Thailand, 10330

⁵Φ-lab, European Space Agency, Frascati (RM), Italy

Quantum generative modeling is among the promising candidates for achieving a practical advantage in data analysis. Nevertheless, one key challenge is to generate large-size images comparable to those generated by their classical counterparts. In this work, we take an initial step in this direction and introduce the *Latent Style-based Quantum GAN (LaSt-QGAN)*, which employs a hybrid classical-quantum approach in training Generative Adversarial Networks (GANs) for arbitrary complex data generation. This novel approach relies on powerful classical auto-encoders to map a high-dimensional original image dataset into a latent representation. The hybrid classical-quantum GAN operates in this latent space to generate an arbitrary number of fake features, which are then passed back to the auto-encoder to reconstruct the original data. Our LaSt-QGAN can be successfully trained on realistic computer vision datasets beyond the standard MNIST, namely Fashion MNIST (fashion products) and SAT4 (Earth Observation images) with 10 qubits, resulting in a comparable performance (and even better in some metrics) with the classical GANs. Moreover, we analyze the barren plateau phenomena within this context of the continuous quantum generative model using a polynomial depth circuit and propose a method to mitigate the detrimental effect during the training of deep-depth networks. Through empirical experiments and theoretical analysis, we demonstrate the potential of LaSt-QGAN for the practical usage in the context of image generation and open the possibility of applying it to a larger dataset in the future.

I. INTRODUCTION

Over the past few decades, generative modeling has stood as one of the main pillars in machine learning (ML), revolutionizing not only academia but also industries and everyday life [1–4]. These models aim to generate synthetic data that closely resembles the original data by learning the underlying probability distribution. While operating on high-dimensional data manifolds poses some key challenges, it also inspires researchers to propose diverse architectures [5–9] and training strategies [10, 11].

Among those various architectures, generative adversarial networks (GANs) [6, 10, 12] and diffusion models (DMs) [7, 13, 14] have emerged as two of the most developed and widely used. On one hand, GANs learn the implicit data distribution of an arbitrary dataset by simultaneously training two distinct neural networks in an adversarial minimax game, successfully being used for a wide range of applications such as image generation [15, 16], text-to-image synthesis [17, 18], image-to-image translation [3, 19, 20] and high-energy physics particle shower simulation [4, 21]. On the other hand, DMs rely on iteratively learning to reconstruct data which are intentionally perturbed by noise. Of particular interest, one DM variant known as a latent diffusion model (LDM) [22] incorporates a strength of pre-trained autoencoders to embed original data into a low-dimensional latent space and learns data generation at this level, directly circumventing the issue of operating in a high dimensional space. This approach significantly reduces

the computational resources while retaining high fidelity of generated data [22].

Meanwhile, due to the rise of quantum computers, quantum machine learning (QML) has emerged as a new paradigm for data analysis, harnessing the power of quantum mechanics in the hope of achieving a practical advantage over conventional classical ML [23–32]. Such growing interest has also spurred efforts to extend QML to the context of generative models by employing parametrized quantum circuits to learn either discrete or continuous distributions (see Figure 1 for a visual summary). Discrete generative models (including quantum Born machines [31, 33–39], quantum GANs [40–44] and quantum Boltzmann machines [45, 46]) employ a parametrized n -qubit quantum state to represent a discrete distribution of 2^n bit-strings with generated samples efficiently obtained as measurements in a computational basis. On the other hand, in the case of continuous models such as variational quantum generator [47] or style-based quantum GANs [48, 49], a quantum circuit acts as a feature map and takes classical random input to produce expectation values as new samples. Despite less sampling efficiency, this approach by design naturally handles continuous data generation and is expected to have a wide range of applications, such as image synthesis, where each pixel takes a continuous value.

Compared to discrete models where there exists a relatively larger body of literature [31–40, 43–46, 50], studies of the continuous quantum generative modeling are much less explored (see the Section A for details of the recent research advancements). The proof of concept on small-size data generation was demonstrated in the context of 3-dimensional Monte Carlo event generation [48]. Recently, an expressivity of the continuous model has been investigated and universality is

* su.yeon.chang@cern.ch

† michele.grossi@cern.ch

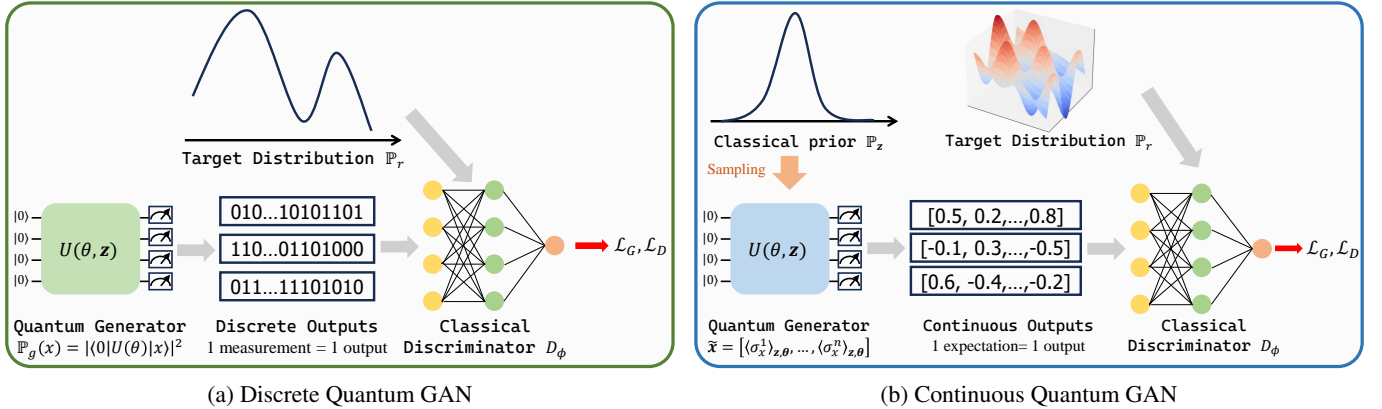


Figure 1. **Schematic diagram summarizing the general training framework of discrete and continuous quantum GANs.** Frequently, we use a hybrid approach with a quantum generator and classical discriminator [40, 48], although an alternative option exists where a quantum discriminator is employed [47].

shown to be achievable under some sufficient conditions [49]. Yet, there remain many open questions, both fundamentally and practically. One of which is *how to achieve the capability of producing an arbitrary number of high-quality images with large sizes*. Given large-size images being generated by classical ML today, resolving this particular problem is one of the key pieces for a practical quantum advantage or utility.

In this work, we propose a hybrid quantum-classical GAN approach, which we call *Latent Style-based Quantum GAN (LaSt-QGAN)*, capable of generating large-size images. We leverage the idea of LDMs by first embedding the complex high-dimensional data into a lower-dimensional latent space using a pre-trained classical autoencoder and then training a style-based quantum GAN directly on this compressed latent representation. After training, expectation values produced by the quantum generator are considered as new features in the latent space, which are then mapped back to the original data space by the autoencoder leading to a new set of large-size images. Compared with standard style-based quantum GANs, our method allows us to push a limit to generate much larger size images despite having the same quantum resources.

Our study focuses on two main objectives. First, we conduct empirical research on LaSt-QGAN in order to understand its potential in practical applications. In comparison to existing frameworks, we empirically showcase the model’s capacity to generate diverse images by testing it on the standard MNIST, the Fashion MNIST dataset, and on Earth observation image dataset, known as SAT4 [51]. Second, we perform further analysis on the model to assess its robustness against statistical fluctuations caused by shot noises and evaluate its trainability at the initial step, a pivotal factor in QML. Crucially, we investigate the barren plateau phenomena of continuous quantum GANs (using both analytical and numerical tools) for the first time, as a complementary contribution to [34, 44] who pioneered this work in the case of the discrete quantum generative models. In particular, this study suggests a possible method to trigger the training of LaSt-QGAN at the initial step with a small angle initialization around the iden-

tity for a polynomial depth quantum circuit, whose loss landscape is exponentially flat on average. As the training of our model happens at the level of latent space, the barren plateau results here are directly applied to continuous generative models based on expectation values in general, including the standard style-based quantum GANs.

The paper is organized as follows. Section II introduces the general training framework of our novel LaSt-QGAN approach. We apply the proposed model to three different datasets and summarize the training results in Section III. In addition, we compare the performance of LaSt-QGAN with a classical GAN which has the same training schema but with a classical generator. The results empirically demonstrate that LaSt-QGAN outperforms the classical generator with a similar model size on these particular tasks. In Section IV, we study the impact of the finite number of measurements and argue the robustness of the model against the statistical fluctuations. We confirm that the errors due to the shot noise cannot be detected by the standard methods for image evaluation. Section V investigates the trainability of LaSt-QGAN in the case of the shallow-depth circuit with numerical simulations and extends the study to the deep-depth circuit case. Finally, in Section VI, we summarize our study with proposals for future research.

II. GENERAL FRAMEWORK

A. Overall training schema

Our work proposes a hybrid classical-quantum GAN approach, so-called, *Latent Style-based Quantum GAN (LaSt-QGAN)*, which integrates two distinct components: a classical autoencoder and a quantum GAN. The autoencoder is an unsupervised neural network used for dimensionality reduction and data compression. It consists of an *encoder*, which embeds the high dimensional data into a low dimensional latent space, and a *decoder*, which reconstructs the data from these

latent features. In our approach, the autoencoder functions as an invertible image preprocessing tool, efficiently reducing the dimensionality of the complex images and reconstructing fake images from the generated latent features. On the other hand, the quantum GAN serves as a generative model for producing fake features, employing a quantum *generator* and a classical *discriminator*. The quantum generator is responsible for generating fake features from a randomly sampled noise, while the classical discriminator differentiates between real and fake features.

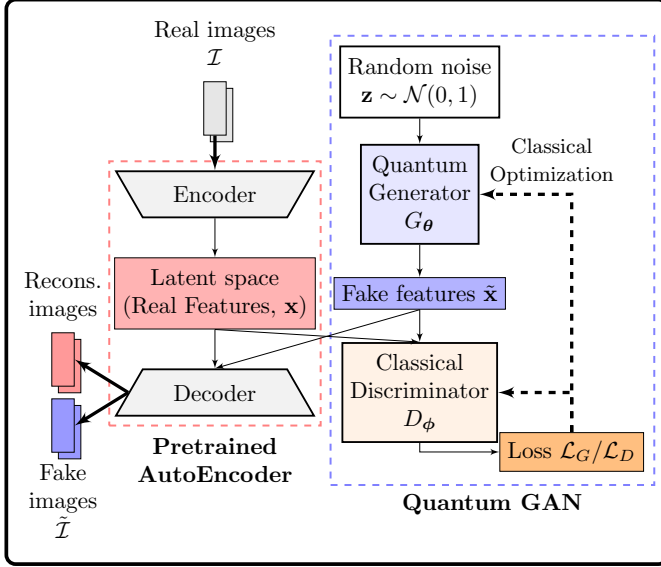


Figure 2. **Schematic diagram for LaSt-QGAN training.** The model consists of a convolutional auto-encoder that embeds the original images into a low-dimensional latent space and a quantum GAN with a quantum generator G_θ and a classical discriminator D_ϕ . The features extracted with the autoencoder are used as the training set of the GAN. At the end of the training, images are reconstructed by inversely transforming the features generated by the quantum generator using the pre-trained convolutional auto-encoder.

The overall training schema of LaSt-QGAN is illustrated on Figure 2. First of all, we extract the essential features, denoted as $\mathbf{x} \sim \mathbb{P}_r$, in the latent space of dimension \mathcal{D}_ℓ from real images \mathcal{I} via a classical convolutional auto-encoder. The auto-encoder is pre-trained on the original image dataset, thus used as an invertible dimensionality reduction technique. Those extracted features are utilized as the real training dataset $\mathcal{X}_{train} \subset \mathbb{R}^{\mathcal{D}_\ell}$ for the quantum GAN training. At each step, G_θ reproduces fake data $G_\theta(\mathbf{z}) = \tilde{\mathbf{x}} \in \tilde{\mathcal{X}} \sim \mathbb{P}_g$, $\tilde{\mathcal{X}} \subset \mathbb{R}^{\mathcal{D}_\ell}$ from a latent noise $\mathbf{z} \in \mathbb{R}^{\mathcal{D}_z}$ sampled randomly from a prior \mathbb{P}_z . Then, the fake and the real features are given as input to the discriminator D_ϕ , which returns a scalar value measuring the *realness* of the samples (i.e., a larger value implies an image is more likely to be real). We note that for the following of the paper, we will keep the tilde mark $\tilde{\cdot}$ to denote the generated samples.

Additionally, Wasserstein loss with gradient penalty [10, 52] is used for better convergence in the model. The gradient penalty corresponds to a regularization term to enforce Lipschitz constraint on the gradients of the discriminator (often

called as *critic* in Wasserstein GAN). This helps to avoid the vanishing gradients in the generator observed in the classical GAN, by excluding the sigmoid functions in the discriminator activations [52]. In this setup, the generator and the discriminator loss functions measure the Wasserstein distance or the *Earth Mover* distance between the output distributions of the real and the fake samples, with the following expression:

$$\mathcal{L}_G(\theta, \phi) = - \mathbb{E}_{\mathbf{z} \sim \mathbb{P}_z} [D_\phi(G_\theta(\mathbf{z}))] \quad (1)$$

$$\begin{aligned} \mathcal{L}_D(\theta, \phi) = & - \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_r} [D_\phi(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim \mathbb{P}_z} [D_\phi(G_\theta(\mathbf{z}))] \\ & + \lambda \mathbb{E}_{\tilde{\mathbf{x}} \sim \mathbb{P}_{\tilde{\mathbf{x}}}} [(\|\nabla_{\tilde{\mathbf{x}}} D_\phi(\tilde{\mathbf{x}})\|_2 - 1)^2] \end{aligned} \quad (2)$$

where the last term corresponds to the gradient penalty of the discriminator. In this term, $\tilde{\mathbf{x}} = \epsilon \mathbf{x} + (1 - \epsilon)\tilde{\mathbf{x}}$ correspond to points interpolated between real and generated samples with a random value ϵ sampled from a uniform distribution and λ the penalty coefficient. These formulas imply that the discriminator aims to maximize the distance, while the generator aims to minimize it.

At the end of the training, the generated data distribution \mathbb{P}_g should approach as close as possible to the real data distribution \mathbb{P}_r . The generated features are then passed back to the and inversely transformed into images $\tilde{\mathcal{I}}$. Thanks to continuity in the latent space, the inverse transform of the generated features leads to the reconstruction of the correct images in the image space.

B. Style-based quantum generator

The quantum generator takes the form of a parameterized quantum circuit, also known as a quantum neural network (QNN). The n -qubit unitary quantum circuit $\mathcal{U}_\theta(\mathbf{z})$ with parameters θ transforms the classical latent noise \mathbf{z} into an encoded quantum state $|\Psi_{\theta, \mathbf{z}}\rangle \in \mathcal{H}$ with \mathcal{H} as a 2^n dimensional Hilbert space. In other words, the parameterized circuit acts as a feature map that maps for the classical input.

Unlike the architecture firstly introduced in Ref [47] where the classical noise embedding layer and trainable layers are separated, the particularity of the style-based architecture is that the rotation angles in the learning layers are also parameterized by the latent noises. Mathematically, the unitary transformation $\mathcal{U}_\theta(\mathbf{z})$ can be written as a L -repetition of learning layers $U_{\theta_\ell}^\ell(\mathbf{z})$ parameterized by the set of parameters θ_ℓ for each layer $\ell = 1, \dots, L$:

$$\mathcal{U}_\theta(\mathbf{z}) = U_{\theta_L}^L(\mathbf{z}) \cdots U_{\theta_1}^1(\mathbf{z}). \quad (3)$$

Then, the latent vectors, \mathbf{z} are embedded into the angles of qubit rotation action, θ_ℓ for each layer ℓ by an affine transformation:

$$\theta_\ell = W_\ell \mathbf{z} + \mathbf{b}_\ell \quad (4)$$

where W_ℓ is the weight matrix of size $N_\theta \times \mathcal{D}_z$ with N_θ the number of rotation angles in QNNs and $\mathbf{b}_\ell \in \mathbb{R}^{N_\theta}$ the

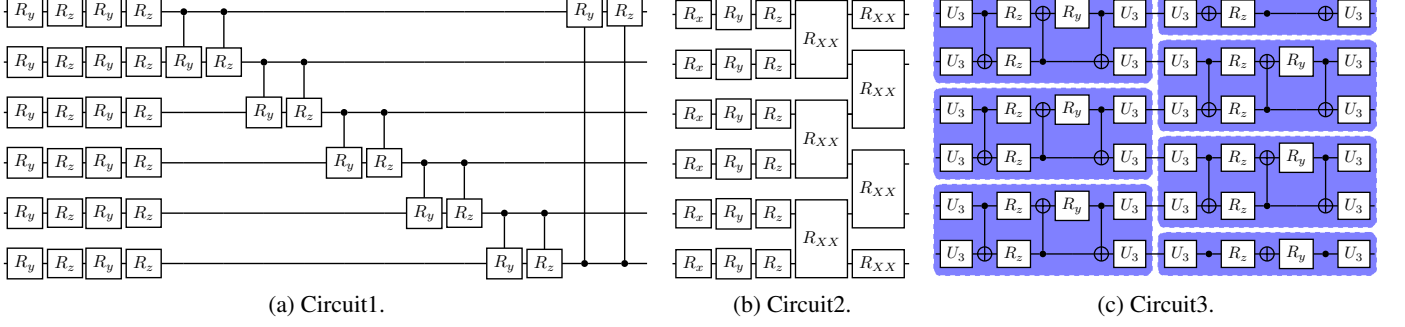


Figure 3. **Different circuit architecture used for learning layers, U_{θ}^{ℓ} , in the quantum generator.** (a) Circuit1 and (b) Circuit2 are taken from two different quantum GAN papers for continuous data generation by C. Bravo Prieto *et al.* [48] and J. Romero *et al.* [47], respectively. (c) Circuit3 is composed of repeated two-qubit quantum circuits (blue square), responsible for an arbitrary $SU(4)$ state generation [53].

bias. During the training, the model will be trained by varying $\Theta = \{W_{\ell}, \mathbf{b}_{\ell}\}_{\ell=1, \dots, L}$ with N_{Θ} the total number of trainable parameters. This can also be regarded as an equivalence of data reuploading technique, where the input data are embedded into the rotation angles in the learning layers for classification task [54, 55], in the context of generative models.

Figure 3 illustrates three different circuit types for a single parameterized layer, $U_{\theta}^{\ell}(\mathbf{z})$, used in this paper for numerical simulations. Circuit1 is the quantum circuit architecture employed in style-based quantum GAN for Monte Carlo event generation [48], and Circuit2 is inspired by the quantum circuit presented in Ref. [47], used as a variational quantum generator (VQG) for continuous distribution. Additionally, we also consider Circuit3, which consists of repeated two-qubit quantum filters (blue square). These filters are responsible for generating an arbitrary $SU(4)$ state [53], thus serving as a universal quantum state generator at least at the level of two-qubits.

After transforming the classical input through the QNN, we measure the expectation values of some observables at the end of the generator to extract some information from the encoded state. Unlike the original architecture [48], which performs only the measurement of the Pauli Z operator, σ_z , our architecture uses expectation values of both Pauli X and Z operators, σ_x and σ_z . The measured values are then concatenated into a single vector, also called a latent feature, which will be given as input to the discriminator:

$$\mathbf{x} = \{\langle \sigma_x^1 \rangle_{\mathbf{z}, \theta}, \dots, \langle \sigma_x^n \rangle_{\mathbf{z}, \theta}, \langle \sigma_z^1 \rangle_{\mathbf{z}, \theta}, \dots, \langle \sigma_z^n \rangle_{\mathbf{z}, \theta}\}_{\mathbf{z}, \theta} \in \mathbb{R}^{2n}, \quad (5)$$

where $\langle \sigma_x^i \rangle_{\mathbf{z}, \theta}, \langle \sigma_z^i \rangle_{\mathbf{z}, \theta}$ denote the expectation values of σ_x and σ_z on i -th qubit for an input latent noise \mathbf{z} and the generator angle θ , i.e., :

$$\langle \sigma_{\mu}^i \rangle_{\mathbf{z}, \theta} = \langle 0 | \mathcal{U}_{\theta}(\mathbf{z})^{\dagger} \sigma_{\mu}^i \mathcal{U}_{\theta}(\mathbf{z}) | 0 \rangle \quad (6)$$

with $\mu \in \{x, z\}$. We note that this strategy does not satisfy the sufficient conditions specified in Ref. [49] for universality. Nevertheless, the numerical results in the following sections demonstrate the model can be adequately used to generate the samples in the training set for the given tasks. More generally,

one can employ a polynomial number of expectation values to construct a latent feature with a larger dimension.

We note that the multi-observable or multi-basis strategy has also been employed in the previous study for multi-classification task [56] or probability learning task [37] to capture the hidden information of the quantum circuit adequately. This way of interpreting the quantum output state allows using only n qubits for $\mathcal{D}_{\ell} = 2n$ values, also bringing an advantage in terms of quantum resources.

C. Evaluation metrics

Unlike the classification task, where the evaluation methods are quite straightforward for the final test accuracy, it is less clear how to evaluate the generative models. There have been efforts to define the appropriate metrics to evaluate the performance of Quantum GAN in previous studies. However, the proposed metrics are limited for discrete generative models [58] as they require one-to-one comparisons of the dataset, or for continuous data with small dimensions [59] where the direct comparison of the probability distribution is available. Therefore, it is important to choose the appropriate metrics to compare the performance between models for image generation. In this paper, we evaluate the performance of GAN with three different metrics: Inception Score (IS) [60, 61], Fréchet Inception Distance (FID) [16, 62] and Jensen-Shannon divergence (JSD) [63].

IS evaluates the quality and the diversity of generated images by calculating the Kullback-Leibler (KL) divergence between marginal distributions obtained by summing up the outputs of the Inception V3 Network [64] applied on real and generated images. Inception V3 Network is a convolutional neural network widely used in classical ML for image recognition task, pretrained on ImageNet dataset (only used for metrics calculation in this paper). Note that the KL divergence for discrete distribution is computed with the following formula :

$$D_{KL}(P||Q) = \sum_{\mathbf{x} \in \mathcal{X}} P(\mathbf{x}) \log \left(\frac{P(\mathbf{x})}{Q(\mathbf{x})} \right) \quad (7)$$

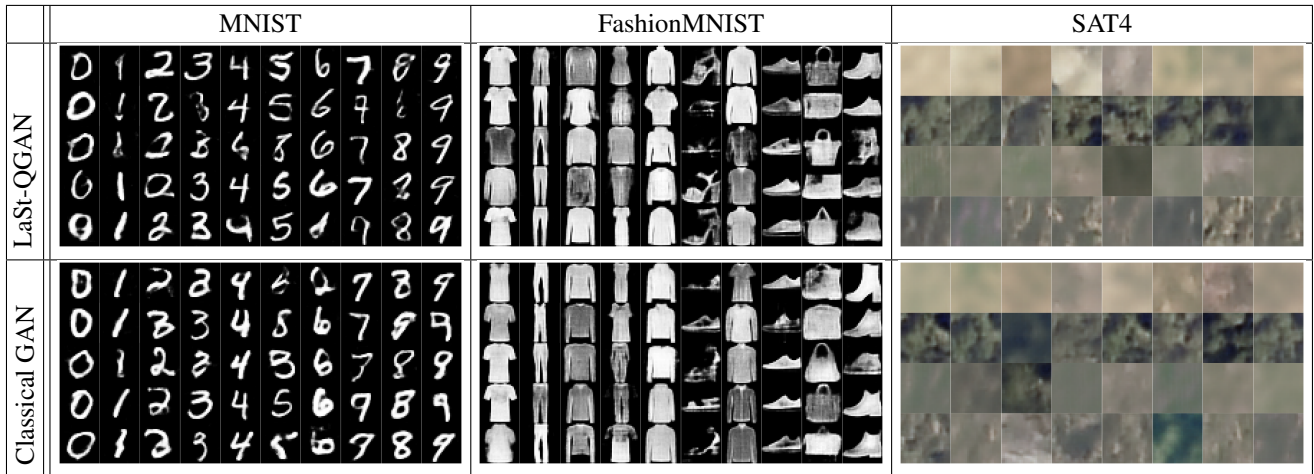


Figure 4. Examples of images generated via LaSt-QGAN (Circuit1, depth 2) and a classical GAN ([50, 30]) for different datasets: MNIST, FashionMNIST and SAT4. The fake features are obtained using $\mathcal{D}_z = 10$ and $\mathcal{D}_\ell = 20$ and the images are reconstructed using a pre-trained convolutional auto-encoder from the features obtained by the GAN in the latent space. The images are presented in columns classified using a pre-trained ResNet50 [57] for MNIST and FashionMNIST, and in rows for SAT4.

for the empirical distribution P and target distribution Q defined on discrete space \mathcal{X} . The maximum value of IS is the number of classes in the dataset, and the higher the IS value, the better the result.

FID also measures the quality of the images using the output of the Inception V3 model, but calculates the Fréchet distance between the real and fake embedding from the model given by the expression:

$$d(X, Y) = \|\mu_X - \mu_Y\|^2 - \text{Tr}(\Sigma_X + \Sigma_Y - 2\sqrt{\Sigma_X \Sigma_Y}) \quad (8)$$

where μ_X, μ_Y are the mean vector of multi-dimensional data X and Y (in this case, the real and fake embedding from the Inception V3) and Σ_X, Σ_Y their covariance matrices. This distance measures the similarity between the distribution of the real and the generated images in the feature space obtained using the Inception V3. The lower the FID value, the better the quality of the images.

Finally, the Jensen-Shannon divergence measures the distance between two discrete probability distributions, similar to the Kullback-Leibler divergence but symmetric and smoother with the following formula :

$$D_{JS}(P||Q) = \frac{1}{2}D_{KL}\left(P||\frac{1}{2}(P+Q)\right) + \frac{1}{2}D_{KL}\left(Q||\frac{1}{2}(P+Q)\right). \quad (9)$$

As JSD requires discrete distributions, in the case of the unlabelled continuous dataset, we first classify the train samples into K bins using K-mean clustering to generate a discrete target distribution, Q . The generated samples are also classified according to the lowest distance from the centers of the train set clusters, returning the generated distribution P , on which we compute the JSD value. The lower the JSD, the more diverse the images.

III. MAIN RESULTS

A. Experimental setup

This section presents the results of LaSt-QGAN trained on MNIST [65] ($28 \times 28 \times 1$ pixels), Fashion MNIST [66] ($28 \times 28 \times 1$ pixels) and SAT4 [51] ($28 \times 28 \times 4$ pixels), which contains 4 classes of Earth Observation images with RGB and Near Infrared channels. We then compare them with the results of their classical counterpart. To keep the latent space embedding model comparable, we used only the RGB channels in the SAT4 dataset. Unless specified, the same model architecture and hyperparameters are used for the following simulations.

The images are embedded into the latent space of dimension $\mathcal{D}_\ell = 20$ with a convolutional auto-encoder. The detailed architecture of the auto-encoder is given in Appendix B. As the output of quantum generator $\langle \sigma_x \rangle$ and $\langle \sigma_z \rangle$ are defined in $[-1, 1]$, the latent space should also be constrained in the same interval. The quantum generator takes $n = 10$ qubits with the latent noises of $\mathcal{D}_z = 10$, each component sampled independently from a normal distribution, $\mathcal{N}(0, 1)$. To guarantee the convergence of the model, the initial quantum generator parameters, W_ℓ and \mathbf{b}_ℓ , are chosen randomly from a uniform distribution between $[-0.01, 0.01]$. The classical discriminator consists of two hidden dense layers with 100 and 50 nodes for MNIST and FashionMNIST and 200 and 100 nodes for the SAT4 dataset, followed by leaky Relu activation functions and an output node of size one.

To assess the performance of LaSt-QGAN against classical models, we construct a classical GAN that follows the identical training framework as LaSt-QGAN, as depicted in Figure 2, but employing a classical linear generator instead of a quantum one. The classical generator consists of an input layer with \mathcal{D}_z nodes, two hidden layers with $[h_1, h_2]$ nodes followed by a leaky Relu activation function, and the output

	G_{θ} config.	N_{Θ}	FID \downarrow	IS \uparrow	JSD (features/ 10^{-2}) \downarrow	JSD (images/ 10^{-2}) \downarrow
LaSt-QGAN	Circ. 1 ($d = 2$)	1360	17.2 ± 0.35	8.29 ± 0.02	0.79 ± 0.05	1.63 ± 0.09
	Circ. 1 ($d = 4$)	2280	14.85 ± 0.34	8.49 ± 0.04	0.75 ± 0.07	1.49 ± 0.18
	Circ. 1 ($d = 6$)	3200	14.13 ± 0.73	8.53 ± 0.05	0.71 ± 0.07	1.29 ± 0.1
	Circ. 2 ($d = 2$)	1010	19.13 ± 0.54	8.10 ± 0.06	1.22 ± 0.19	2.08 ± 0.17
	Circ. 2 ($d = 4$)	1690	16.2 ± 0.32	8.34 ± 0.03	0.94 ± 0.09	1.66 ± 0.17
	Circ. 2 ($d = 6$)	2370	14.85 ± 0.61	8.47 ± 0.06	0.85 ± 0.05	1.39 ± 0.11
	Circ. 3 ($d = 2$)	3300	14.29 ± 0.38	8.5 ± 0.04	0.76 ± 0.06	1.5 ± 0.12
	Circ. 3 ($d = 4$)	6600	12.72 ± 0.4	8.65 ± 0.05	0.71 ± 0.07	1.14 ± 0.12
	Circ. 3 ($d = 6$)	9900	11.99 ± 0.56	8.71 ± 0.04	0.72 ± 0.09	1.13 ± 0.12
Classical	[50, 30]	2960	18.24 ± 3.6	8.24 ± 0.28	3.74 ± 1.64	4.51 ± 2.0
	[100, 50]	7660	12.56 ± 0.91	8.8 ± 0.06	1.18 ± 0.17	1.56 ± 0.13

Table I. **Training results of LaSt-QGAN and the classical GAN for MNIST dataset.** Number of parameters used in the generator and different metrics (averaged over 10 runs) to compare the performance of LaSt-QGAN and the corresponding classical GAN using 10,000 generated images (best results highlighted in bold). For FID and JSD, the lower, the better and for IS, the higher, the better. We observe that with a similar model size ($\approx 3k$ parameters), LaSt-QGAN outperforms the classical GAN for all metrics. Note that our results are close to the result of SoTA vanilla GAN models, which have FID of 7.87 [67] and 12.88 [68].

layer with \mathcal{D}_{ℓ} nodes attached to a Tanh function to constrain the generator output between -1 and 1. For the following simulations, we consider two different classical generators: 1) $[h_1, h_2] = [50, 30]$, 2) $[h_1, h_2] = [100, 50]$. The first one is chosen to have a similar number of parameters as the quantum generators, while the second one is constructed to have the same hidden layers as the discriminator for a balanced GAN architecture. In order to also guarantee faster convergence for classical neural networks, we use LeCun normal initialization [65] for the parameters.

In all cases, the model parameters are updated with an Adam optimizer using a learning rate of 0.001 for both discriminator and generator with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. Those hyperparameters are chosen empirically to assure the fastest convergence and stability of the model. For loss calculation, $\lambda = 10$ is chosen as the penalty coefficient (c.f. Eq. (2)).

Our codes used Jax [69] and Flax [70] packages for training algorithms implementation, and PennyLane [71] for quantum circuits construction and optimization.

B. Generic results

We display in Figure 4 the images of different datasets generated by LaSt-QGAN and the corresponding classical counterpart using the features extracted by the pre-trained convolutional auto-encoder. The results prove that the model can reproduce images correctly, although further improvements are required for a higher quality of the results.

Figure 5 visualizes the distribution of features generated by the classical and style-based quantum generators, downsampled using t-distributed Stochastic Neighbor Embedding (t-SNE) [74, 75] for MNIST and Fashion MNIST dataset. Each feature is labeled after classifying the generated images using the ResNet50 pre-trained on the real image dataset. Although

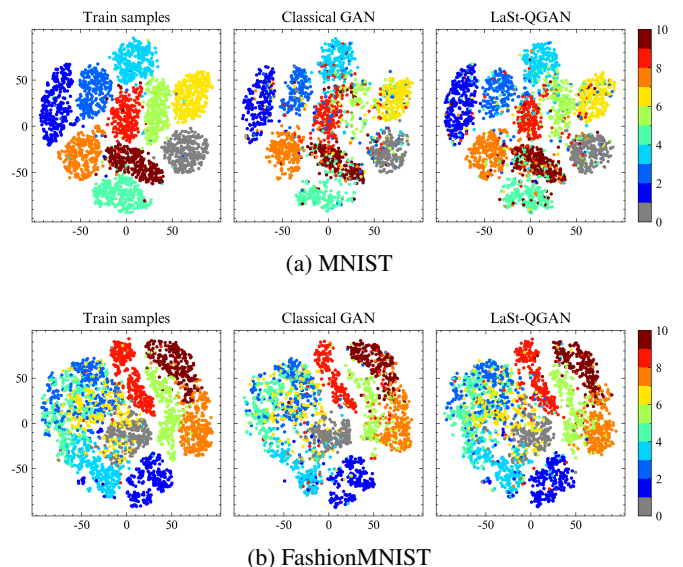


Figure 5. **Visualization of generated features embedded into two dimensions using t-SNE for MNIST and FashionMNIST dataset.** The labels of generated samples are obtained via classification with pre-trained ResNet50. The clustering of features reveals that the underlying similarity in each class is preserved in the latent space with the proposed models.

the separation of the generated features is not as clear as that of the real training set, the clustering of samples within a class reveals that the underlying structure of the data distribution is preserved during the data generation.

The performance of the GAN training is also quantified in terms of the metrics introduced in Section II C. In Table I, II and III, we present the comparison of the top-performing results of LaSt-QGAN with different quantum generator archi-

	G_θ config.	N_Θ	FID ↓	IS ↑	JSD (features/ 10^{-2}) ↓	JSD (images/ 10^{-2}) ↓
LaSt-QGAN	Circ. 1 ($d = 2$)	1360	29.42 ± 0.59	8.27 ± 0.04	1.01 ± 0.095	1.61 ± 0.2
	Circ. 1 ($d = 4$)	2280	27.59 ± 0.56	8.37 ± 0.02	0.85 ± 0.05	1.42 ± 0.1
	Circ. 1 ($d = 6$)	3200	26.89 ± 0.57	8.44 ± 0.02	0.76 ± 0.08	1.28 ± 0.11
	Circ. 2 ($d = 2$)	1010	32.26 ± 0.43	8.12 ± 0.05	1.28 ± 0.12	2.06 ± 0.14
	Circ. 2 ($d = 4$)	1690	29.2 ± 0.3	8.34 ± 0.03	0.94 ± 0.09	1.69 ± 0.12
	Circ. 2 ($d = 6$)	2370	28.1 ± 0.77	8.47 ± 0.06	0.85 ± 0.05	1.40 ± 0.16
	Circ. 3 ($d = 2$)	3300	27.8 ± 0.88	8.34 ± 0.03	0.81 ± 0.08	1.35 ± 0.12
	Circ. 3 ($d = 4$)	6600	25.96 ± 0.52	8.5 ± 0.05	0.75 ± 0.09	1.50 ± 0.23
	Circ. 3 ($d = 6$)	9900	25.43 ± 0.4	8.56 ± 0.04	1.08 ± 0.2	1.50 ± 0.23
Classical	[50, 30]	2960	28.32 ± 0.88	8.52 ± 0.09	3.06 ± 0.45	2.73 ± 0.29
	[100, 50]	7660	27.36 ± 1.51	8.57 ± 0.04	2.49 ± 0.63	2.81 ± 0.68

Table II. **Training results of LaSt-QGAN and the classical GAN for FashionMNIST dataset.** Number of parameters used in the generator and different metrics (averaged over 10 runs) to compare the performance of LaSt-QGAN and the corresponding classical GAN using 10,000 generated images (best results highlighted in bold). For FID and JSD, the lower, the better and for IS, the higher, the better. We observe that with a similar model size ($\approx 3k$ parameters), LaSt-QGAN outperforms the classical GAN for all metrics, except for IS. Note that our results are among the best results obtained with the classical SOTA generative models [72], close to the FID of 21.73 [67] and 28.0 [73].

	G_θ config.	N_Θ	FID ↓	IS ↑	JSD (features/ 10^{-2}) ↓	JSD (images/ 10^{-2}) ↓
LaSt-QGAN	Circ. 3 ($d = 2$)	3300	168.28 ± 2.06	3.57 ± 0.01	1.26 ± 0.21	2.07 ± 0.27
Classical	[100, 50]	7660	172.6 ± 5.02	3.5 ± 0.03	6.99 ± 1.13	4.25 ± 0.65

Table III. **Training results of LaSt-QGAN and the classical GAN for SAT4 dataset.** Number of parameters used in the generator and different metrics (averaged over 10 runs) to compare the performance of LaSt-QGAN and the corresponding classical GAN using 10,000 generated images. For FID and JSD, the lower, the better and for IS, the higher, the better. We observe that LaSt-QGAN outperforms the classical benchmark for all metrics by using only half the number of parameters. Note that the highest IS value for the SAT4 dataset is 4, as it consists of 4 classes.

tures and the classical GAN, using FID, IS and JSD computed on 10,000 samples. In particular, for JSD, we assess the performance by analyzing both the generated features and reconstructed images to gauge its ability to mimic the original data distribution before and after image reconstruction. We see that with a similar number of parameters, LaSt-QGAN outperforms the classical benchmark for all types of datasets not only in terms of quality (FID, IS) but also in terms of diversity (JSD) in both features and images, showing that the model can successfully learn the hidden distribution of the real data. Notably, our LaSt-QGAN achieves the FID value close to the state-of-the-art GAN techniques for the MNIST dataset, which are 7.87 [67] and 12.884 [68].

The rate of convergence serves as another crucial aspect in GAN training. Figure 6 illustrates the progression of various evaluation metrics for LaSt-QGAN and its classical counterpart with different model architectures. As empirically observed in the plot, faster convergence is exhibited for LaSt-QGAN compared to the classical GAN for both MNIST and FashionMNIST datasets. Notably, for the MNIST dataset, we reach the FID value below 20 in fewer than 20 training epochs for all depth d , which is at least twice as fast as the classical one. One might argue that the faster convergence is due to the fact that the quantum generator has a low number of parameters. However, the faster convergence is also observed using Circuit3 with depth 6 in the LaSt-QGAN which is com-

posed of more parameters compared to the classical GAN, empirically showing that this is independent of the number of parameters. We further stress that small standard deviations reveal the stability of training with LaSt-QGAN during the whole training process, solving the training instability, one of the major issues in GANs [11]. This aligns with the previous studies on the beneficial capacity properties and faster training convergence, which were experimentally proven in the previous papers in the context of classification task [76] and discrete QGAN [44].

C. Dependence on the dataset size

In this section, we train LaSt-QGAN with smaller training sets for MNIST and FashionMNIST datasets, comparing the outcomes against the classical GAN to study the generalization power of the quantum generator. That is, we study how close the underlying distribution of a generator trained on a small set of training data is to a true target distribution of the original images as a function of a training data size. In particular, the models are trained on varying sample sizes, $N = 2^k \times 1000$, $k = 0, \dots, 5$ samples, as well as on the complete training set, $N = 60,000$. To maintain consistency in the number of updates per epoch, we employ batch sizes of $N_{bs} = 4^k$ for each $N = 2^k \times 1000$, where $k = 0, \dots, 5$, and a

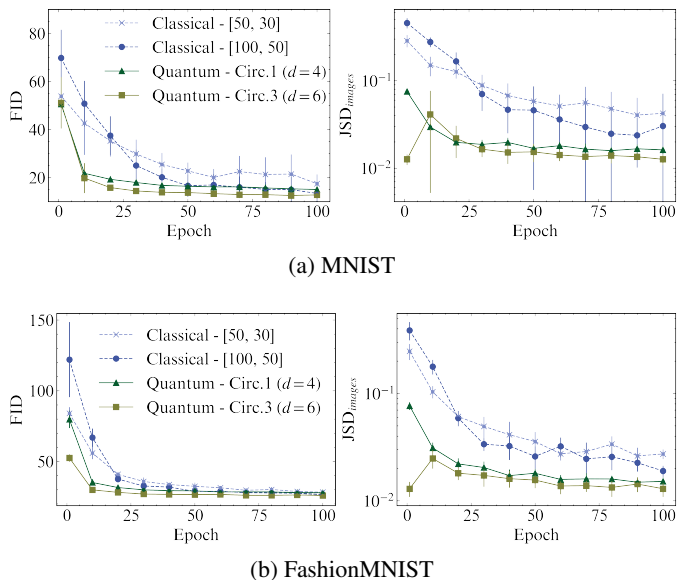


Figure 6. **Evaluation of training dynamics and stability (averaged over 10 runs) for LaSt-QGAN and classical GANs.** The metrics computed over 10,000 samples generated during the training of LaSt-QGAN and the classical GAN for MNIST and FashionMNIST dataset. We observe faster convergence and higher stability with LaSt-QGAN than the classical model for both datasets using a similar number of parameters. Furthermore, for all tested models, LaSt-QGAN reaches lower JSD compared to the classical model, highlighting its power to learn the hidden data distribution.

batch size of $N_{bs} = 4^6 = 256$ for the entire training set.

Figure 7 depicts the values of FID and JSD obtained at the end of the training with varying dataset sizes N using 10,000 generated samples every time. For both datasets, the LaSt-QGAN results in slightly better performance in terms of different evaluation metrics compared to the classical GAN with a similar number of parameters ($[h_1, h_2] = [50, 30]$) for smaller N , proving a higher generalization power with a small training set. In particular, the improvement is more pronounced with the MNIST dataset for the small generator size: we reach FID less than 20 only with $N = 4k$ samples with LaSt-QGAN. Conversely, with a larger generator size, the quantum generator demonstrates improved distribution learning capabilities for the FashionMNIST dataset. This dataset is distinguished by a strong correlation among latent features compared to the MNIST dataset, indicating the potential applicability of this architecture for datasets characterized by significant correlation. This observation can be elucidated through the measurements of Z and X observables, inherently correlated in their construction of outputs. It is also notable that in terms of JSD, we observe that it always outperforms the classical GAN for all model sizes, even with twice the number of parameters. Furthermore, lower standard deviations obtained in all cases with LaSt-QGAN prove the stability of the quantum generator compared to the classical one.

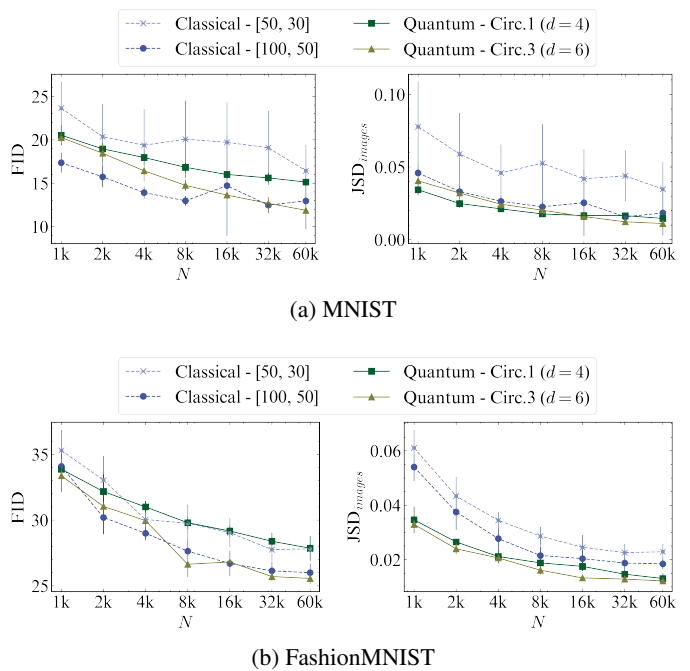


Figure 7. **Comparison of the generalization power in LaSt-QGAN and its classical counterpart with varying dataset sizes, N .** Metrics are computed over 10,000 samples generated at the end of the training with different training set sizes and averaged over 10 runs. For small dataset sizes, we observe that LaSt-QGAN using Circuit1 with $d = 4$ consistently performs better than the classical GAN with $[50, 30]$ hidden nodes, which has a similar number of parameters (see Table I), indicating its ability to generalize from limited data. Additionally, the larger quantum model (Circ3 - $d = 6$) shows stronger performance with the FashionMNIST dataset, making it suitable for datasets with a more complex feature correlation.

IV. ROBUSTNESS AGAINST STATISTICAL NOISE

Up to this section, LaSt-QGAN has been trained analytically, under the *infinite number of shots assumption*. Nonetheless, in practical application, the quantum states are sampled with a finite number of shots and one might argue that the resulting statistical noise might potentially degrade the quality of images in the real-case scenario. In this section, we demonstrate that the model is robust against the statistical fluctuation coming from the finite number of shots.

We denote $\tilde{\mathbf{x}}_\infty$ and $\tilde{\mathcal{I}}_\infty$ the feature and the image generated analytically, and \tilde{x}^i the i^{th} component in the sample $\tilde{\mathbf{x}}$. For simplicity, we use the parameters of LaSt-QGAN pre-trained analytically and generate the features $\tilde{\mathbf{x}}_{shots}$ to reconstruct images $\tilde{\mathcal{I}}_{shots}$ using varying numbers of shots, $N_{shots} = 2^k$ for $k = 4, \dots, 13$.

On Figure 8a, we plot the Euclidean distance $\|\Delta\tilde{\mathbf{x}}\| = \|\tilde{\mathbf{x}}_{shots} - \tilde{\mathbf{x}}_\infty\|$, where $\tilde{\mathbf{x}}_{shots}$ and $\tilde{\mathbf{x}}_\infty$ are generated with the same input noise, averaged over 10,000 samples for MNIST dataset. Furthermore, as a reference, we indicate the mean and minimum separation between two samples in the training set, i.e. $\|\Delta\mathbf{x}_{train}\| = \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}_{train}}$. It is noteworthy that $\|\Delta\tilde{\mathbf{x}}\|$ drops below $\min\|\Delta\mathbf{x}_{train}\|$ after $N_{shots} = 256$. This

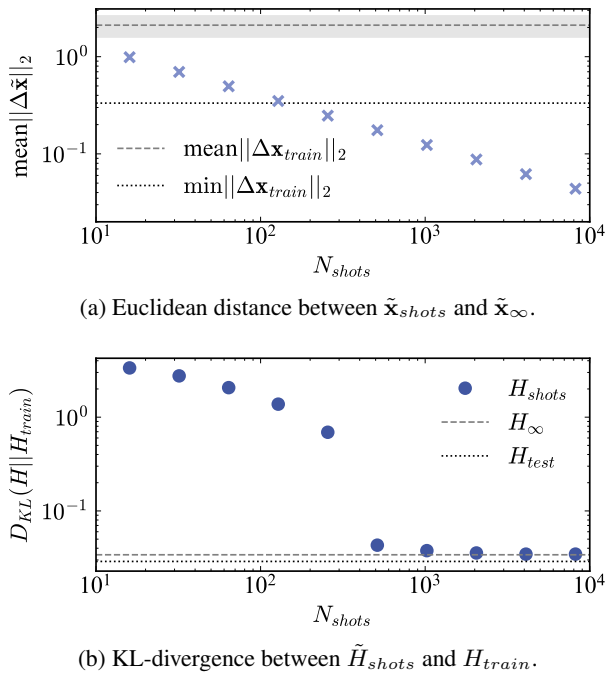


Figure 8. **Quality of the features generated with different numbers of measurements.** (a) The L_2 distance between $\tilde{\mathbf{x}}_{shots}$ and $\tilde{\mathbf{x}}_{\infty}$ for MNIST dataset. The dashed line and the dotted line represent the mean and the minimum separation between samples inside the training set, i.e., $\|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}_{train}}$. (b) The KL-divergence D_{KL} calculated between the histograms \tilde{H}_{shots}^i and H_{train}^i over x^i and \tilde{x}_{train}^i using 500 bins. The final values are averaged over all the components $i = 1, \dots, 20$. Unlike $\|\Delta\tilde{\mathbf{x}}\|_2$ which decays exponentially with respect to the number of shots, D_{KL} converges from $N_{shots} = 256$.

implies that the features generated with more than 256 shots are close enough to the analytical features, positioning them within the vicinity of the corresponding $\tilde{\mathbf{x}}_{\infty}$ to differentiate them from other samples.

To understand the general statistics over the generated features, we construct the histograms H_{shot}^i for \tilde{x}_{shot}^i and H_{train}^i for \tilde{x}_{train}^i to compute the KL-divergence between them. Figure 8b displays the KL-divergence averaged over $i = 1, \dots, 20$. This underlines that, despite an exponential number of shots required for the exact outcomes, the overall statistics of each feature converge towards those of the training set with a finite number of shots larger than $N_{shots} = 512$.

To make this line of argument more concrete, we analyze the impact of a finite number of measurements on the generated images. Figure 9 shows the images generated with different numbers of shots for the MNIST and the FashionMNIST datasets. With bare eyes, we observe that the quality of images becomes already faithful with $N_{shots} = 256$, which aligns with the threshold observed for $\|\Delta\tilde{\mathbf{x}}\|_2$. This can be confirmed with a quantitative analysis of the images using FID metrics. As shown on Figure 10a, the absolute pixel-by-pixel difference between $\tilde{\mathcal{I}}_{shot}$ and $\tilde{\mathcal{I}}_{\infty}$ decreases exponentially with the number of shots, which might lead the readers to confirm the necessity of the infinite number of shots. However, on con-

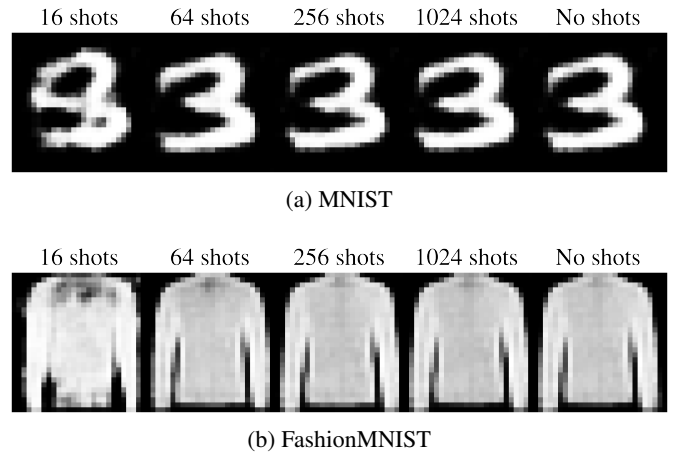


Figure 9. **(a) MNIST and (b) FashionMNIST images generated with various number of measurements.** We can observe that the images get closer to the $\tilde{\mathcal{I}}_{\infty}$ from 256 shots.

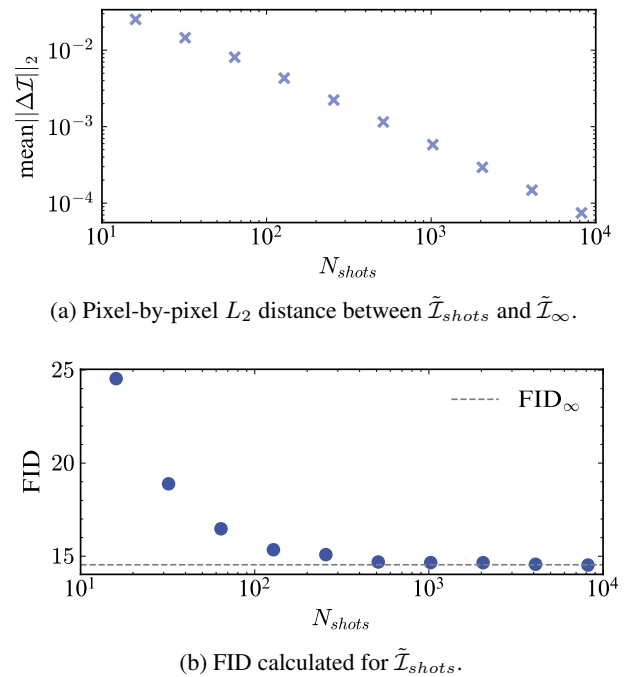


Figure 10. **Quality of the images generated with different numbers of measurements.** (a) The pixel-by-pixel L_2 distance between $\tilde{\mathcal{I}}_{shots}$ and $\tilde{\mathcal{I}}_{\infty}$ for MNIST dataset. The exponential decay in $\|\Delta\tilde{\mathbf{x}}\|_2$ shown on Figure 8a is also leveraged for $\|\Delta\tilde{\mathcal{I}}\|_2$. (b) FID value computed for $\tilde{\mathcal{I}}_{shots}$ with different number of shots. Despite an exponential decay in the absolute pixel-by-pixel difference between the $\tilde{\mathcal{I}}_{shots}$ and $\tilde{\mathcal{I}}_{\infty}$, the FID converges to FID_{∞} after $N_{shots} = 512$.

trary, the FID value shown on Figure 10b converges to the FID of $\tilde{\mathcal{I}}_{\infty}$ from $N_{shots} = 512$. Indeed, in practical implementation, the standard methods used to evaluate the quality of the images do not detect the difference occurring by statistical fluctuation. This empirically demonstrates that with a good construction of the classical autoencoder responsible for post-processing, the impact of shot noise can be alleviated, al-

cluding to the feasibility of using a finite number of shots for image generation.

It is important to acknowledge that when training on actual quantum hardware, the gradient computation will also be affected by finite shots, impacting the quality of the resulting features at the end of the training. Nevertheless, this study still provides valuable insight into mitigating the fluctuations in the features thanks to postprocessing, as long as the features converge towards real values within a certain range.

V. MITIGATING BARREN PLATEAUS

One of the main challenges in PQC training is the problem of the exponentially vanishing loss gradients, also known as *barren plateaus* [77, 78]. In particular, consider a loss function $\mathcal{L}(\theta)$ of the form

$$\mathcal{L}(\theta) = \langle \psi_0 | U^\dagger(\theta) O U(\theta) | \psi_0 \rangle, \quad (10)$$

where $U(\theta)$ is some parametrized circuit, O is some observable and $|\psi_0\rangle$ is some initial state. We say that the loss function $\mathcal{L}(\theta)$ exhibits a barren plateau if, for all the parameters θ_ν , there exists $b > 1$ such that :

$$\text{Var}_\theta[\partial_\nu \mathcal{L}(\theta)] \in \mathcal{O}\left(\frac{1}{b^n}\right), \quad (11)$$

where we introduce a shorthand notation $\partial_\nu \mathcal{L}(\theta) := \partial \mathcal{L}(\theta) / \partial \theta_\nu$. Note that the definition of the barren plateau is also equivalent to showing that $\text{Var}_\theta[\mathcal{L}(\theta)] \in \mathcal{O}(1/b^n)$ for some $b' > 1$, which implies the exponentially flat loss landscape [79]. Consequently, the number of measurement shots required to navigate through the flat region scales exponentially with the number of qubits, posing a serious scaling problem for trainability of PQCs.

Recently, it has been argued that various sources of barren plateaus previously discovered [77, 80–87] can be unified under one key concept of *the curse of dimensionality* whereby quantum states in the exponentially large Hilbert space are inappropriately handled [78, 88–91]. While initially discussed in the setting of the loss function in Eq. (10), the studies of BPs have largely been extended to various QML frameworks which take into account training data and non-linear loss functions [32, 37, 44, 87, 92, 93] – even quantum models that are trained solely on classical computers [29, 94–97]. Of our particular interest, Ref. [37, 44] investigates barren plateau in quantum generative models [37, 44], but only in the case of the discrete models.

In this section, we study the barren plateau phenomena in the LaSt-QGAN by analyzing the generator loss given by Eq. (1). Although the generator loss does not take the form of an expectation value as shown in Eq. (10), it can be seen as a post-processing of expectation values. We begin by empirically investigating the variance of the partial derivative $\partial_\nu \mathcal{L}_G$ of the generator loss. Here, the derivative is only with respect to the generator parameters (since those are parameters in the quantum circuits) and the variance is taken over both the

generator and the discriminator parameters, Θ and ϕ . For the quantum generator, the weights W and the biases \mathbf{b} are initialized randomly from a uniform distribution $[-\delta, \delta]$ and the input noises \mathbf{z} are sampled from a normal distribution, $\mathcal{N}(0, 1)$. In addition, the rotation angles are rescaled with respect to the latent space dimension \mathcal{D}_z i.e.,

$$\theta = \frac{1}{\sqrt{\mathcal{D}_z}} W \mathbf{z} + \mathbf{b}. \quad (12)$$

Due to the Central Limit Theorem [98], each element of θ independently follows a normal distribution, centered around 0 with a standard deviation $\sigma \approx \delta$.

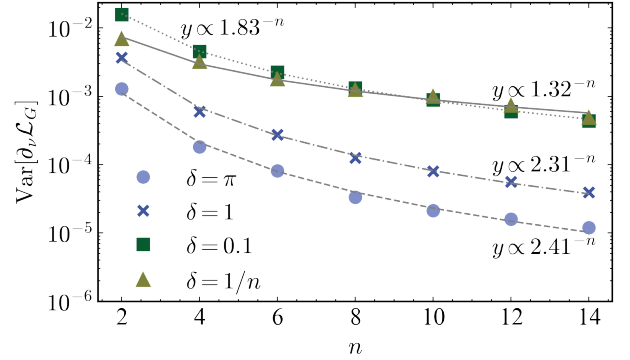


Figure 11. **Variance of the partial derivative of \mathcal{L}_G versus the number of qubits n using logarithmic depth quantum circuit.** The variance is computed with $\mathcal{D}_z = n$ for different initialization ranges, δ , and averaged over the parameters of the first layer. The quantum generator consists of Circuit1, with logarithmic depth, $d = \lfloor \log(n) \rfloor$. Regardless of δ , \mathcal{L}_G does not exhibit BP with polynomially decaying variance, as the loss function only contains local observables, with zero initial state.

To understand the behavior of the gradients, we numerically compute the variance of partial derivatives $\partial_\nu \mathcal{L}_G$ with respect to the number of n for different initialization bounds δ as shown on Figure 11. Here, we note that a quantum circuit is said to be free of the barren plateau if $\partial_\nu \mathcal{L}_G$ decays polynomially *at least* with respect to one of the parameters. Therefore, in our analysis, we focus on calculating the derivatives with respect to the parameters in the first layer of the generator circuit and take an average over them. On Figure 11, the fitting curves clearly prove that $\text{Var}_{\Theta, \phi}[\partial_\nu \mathcal{L}_G]$ decays polynomially with n , i.e., $\text{Var}_{\Theta, \phi}[\partial_\nu \mathcal{L}_G] \in \mathcal{O}(1/n^b)$ with $b > 1$, although b increase with n . This polynomial decay indicates an absence of barren plateaus and is indeed expected from the fact that the quantum generator only consists of single-qubit local observables together with limited expressivity of log-depth circuits [81].

We extend the study to the polynomial depth scenario with different initialization range δ . In this regime, the circuit is sufficiently expressive to give rise to barren plateaus. As shown in Figure 12, when randomly initializing the parameters with $\delta = \pi$, the variance of the loss gradients vanishes exponentially in the number of qubits. On the other hand, in the case where we use a small angle initialization with the initial

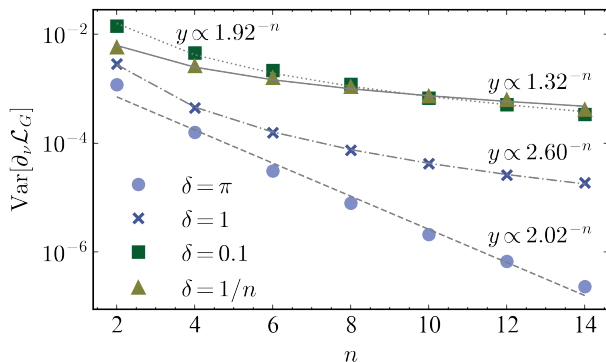


Figure 12. **Variance of the partial derivative of \mathcal{L}_G versus the number of qubits n using polynomial depth quantum circuit.** The variance is computed with $\mathcal{D}_z = n$ for different initialization ranges, δ , and averaged over the parameters of the first layer. The quantum generator consists of Circuit1, with logarithmic depth, $d = \lfloor \text{poly}(n) \rfloor$. We observe a clear existence of the BP for $\delta = \pi$. However, using warm start with $\delta = 0.1$ and $\delta = 1$, \mathcal{L}_G decays polynomially, showing BP free regime.

parameters sampled from a certain range with $\delta = 1$, $\delta = 0.1$ and $\delta = 1/n$ around the circuit identity, $\text{Var}_{\Theta, \phi}[\partial_\mu \mathcal{L}]$ is empirically observed to decay polynomially, which mitigates the effect of barren plateaus. For larger system size, while there is no analytical guarantee on the scaling with fixed small δ and one could in principle expect the scaling to turn into exponential, we can analytically guarantee that the variance scaling remains polynomial with δ scaling with the system size.

To further probe this with some analytics, we first note that, since the loss is a post-processing of expectation values which essentially are of the form Eq. (10), the loss function does not suffer from barren plateaus if these expectation values do not exponentially concentrate over the parameters. For more details, Appendix C gives an insight regarding the lower bound on the loss concentration in the polynomial depth circuit with the small angle initialization initialization [99, 100] using EfficientSU2 ansatz [101] for both local and general observables. It supplements the prior research on normal initialization [100] by providing a tight lower bound and a comprehensive insight into the behavior of the loss function based on the initial quantum state and final measurement. In particular, for the EfficientSU2 ansatz, if the initial range δ scales as $1/n$, the loss function \mathcal{L} decays as :

$$\text{Var}_\theta[\mathcal{L}(\theta)] \gtrsim \frac{1}{n^b}, \quad b > 2. \quad (13)$$

On Figure 11 and 12, we plot as well $\text{Var}_{\Theta, \phi}[\partial_\nu \mathcal{L}_G]$ with varying $\delta = 1/n$. In this plot, we use Circuit1 as the quantum generator instead of EfficientSU2 ansatz, but we observe that the variance also decays polynomially as expected by Eq. (13), indicating the mitigation of barren plateau within a certain range.

Lastly, we remark that while the circuits with log-depth or the small angle initialization are shown to evade barren plateaus in our model, the loss landscape can be classically

simulable as discussed in the recent study [88]. The subtle difference between the two cases is that the circuit with log-depth leads to a classical simulability of the loss at any point of an entire landscape while only a small region around identity initialization can be classically simulated on average with the deep circuit. Although there is no guarantee of achieving the optimal solution, the small angle initialization will allow at least reaching the local or suboptimal minimum in the loss landscape.

VI. DISCUSSION

Quantum generative modeling has attracted much recent attention as one of the promising applications of quantum computers in data analysis. Nonetheless, there remain fundamental and practical challenges before a practical quantum advantage could be achieved. One of which is how to generate images with a dimension comparable to those generated by classical generative models.

In this paper, we introduced LaSt-QGAN, which combines a classical latent embedding and a quantum GAN under a unified framework. By using the latent technique, images are mapped into a latent space with smaller dimensions where a style-based quantum GAN is trained to learn the latent representation of the images. This combined approach enables us to larger image generation with a hybrid quantum-classical generative model.

Our empirical results demonstrate that the model can effectively synthesize images with a better quality level than the classical counterpart using approximately the same resources. In particular, from various quantitative evaluation metrics, we empirically observe that for these specific learning tasks, the quantum GAN is capable of achieving, and in some cases even surpassing, the performance of classical GAN in terms of both quality and diversity of the generated samples across all tested datasets while maintaining a similar number of trainable parameters. Furthermore, we investigated the performance of the models under varying dataset sizes and observed that LaSt-QGAN reaches a comparable level of performance to the classical GAN, even when using a smaller dataset size, as well as the effect of shot noise on the resolution of the generated images. These empirical findings constitute a first step to demonstrate our model's potential for practical applications. Nevertheless, since our model relies on the classical autoencoders to amplify the outputs from a quantum GAN, it is a fundamental open question to see whether the interplay between the classical and quantum parts can be quantified.

We also study the barren plateau phenomena in the continuous generative models. Crucially, by using a mix of analytical and numerical tools, we show that LaSt-QGAN with a polynomially deep generator circuit can be trained with a small angle initialization around the identity. Despite the loss landscape being exponentially flat on average, the strategy allows us to initialize on a region with substantial gradients and train towards some local minimum. Nevertheless, while providing a temporary remedy to a barren plateau problem, the strategy has certain drawbacks. We cannot guarantee the quality of the

local minimum if the circuit itself contains no inductive bias that aligns with the target distribution. In addition, the small region around the initialization can be classically simulable on average. Crucially, we note that our barren plateau results here are also directly applicable to other continuous quantum generative models based on sampling expectation values such as the original style-based quantum GANs.

To go beyond the identity initialization, one could consider a warm-start strategy, i.e., a smart initialization that incorporates the problem structure into consideration [88, 102–104]. Recently, a warm start in the context of variational quantum simulation has been analytically studied in Ref. [102], showing the potential of a warm-start strategy to circumvent barren plateaus but at the same time highlight additional challenges for achieving global minimum. Further investigation of warm-starts in the generative modeling setting is of particular importance for both fundamental and practical aspects.

Lastly, it is crucial to remark that the role of a quantum circuit in the continuous generative model as a feature map shares a great similarity in the supervised quantum machine learning with classical data. This implies some of the pieces of knowledge in the literature can be applied to the generative setting. For example, one fundamental concern is the risk of

the continuous generative models being classically surrogatable by similar techniques such as random Fourier feature [105]. On the other hand, a provable quantum advantage based on cryptographic hardness [26, 31] strongly suggests the existence of classically hard continuous quantum generative models. Since the fundamental natures between discriminative and generative models do not perfectly align, to what extent one can apply the results from one field to another remains unanswered, leaving a great opportunity for future research.

ACKNOWLEDGMENTS

We would like to acknowledge Zoë Holmes for insightful discussions. SC is supported by the quantum computing for earth observation (QC4EO) initiative of ESA Φ -lab, partially funded under contract 4000135723/21/I-DT-Ir, in the FutureEO program. ST acknowledges support from the Sandoz Family Foundation-Monique de Meuron program for Academic Promotion. MG is supported by CERN through the CERN Quantum Technology Initiative.

-
- [1] P. P. Ray, Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope, *Internet of Things and Cyber-Physical Systems* **3**, 121 (2023).
 - [2] G. Daras and A. G. Dimakis, Discovering the hidden vocabulary of dalle-2 (2022), [arXiv:2206.00169](https://arxiv.org/abs/2206.00169) [cs.LG].
 - [3] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, Image-to-image translation with conditional adversarial networks, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE Computer Society, Los Alamitos, CA, USA, 2017) pp. 5967–5976.
 - [4] M. Paganini, L. de Oliveira, and B. Nachman, Calogan: Simulating 3d high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks, *Phys. Rev. D* **97**, 014021 (2018).
 - [5] D. P. Kingma and M. Welling, Auto-encoding variational bayes (2022), [arXiv:1312.6114](https://arxiv.org/abs/1312.6114) [stat.ML].
 - [6] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Generative adversarial networks (2014), [arXiv:1406.2661](https://arxiv.org/abs/1406.2661) [stat.ML].
 - [7] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, in *Proceedings of the 32nd International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 37 (PMLR, 2015) pp. 2256–2265.
 - [8] Y. Du and I. Mordatch, Implicit generation and modeling with energy based models, in *Advances in Neural Information Processing Systems*, Vol. 32, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., 2019).
 - [9] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, Normalizing flows for probabilistic modeling and inference, *J. Mach. Learn. Res.* **22** (2021).
 - [10] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, Improved training of wasserstein gans, in *Advances in Neural Information Processing Systems*, Vol. 30 (Curran Associates, Inc., 2017) p. 5769–5779.
 - [11] M. Arjovsky and L. Bottou, Towards principled methods for training generative adversarial networks (2017), [arXiv:1701.04862](https://arxiv.org/abs/1701.04862) [stat.ML].
 - [12] M. Mirza and S. Osindero, Conditional generative adversarial nets (2014), [arXiv:1411.1784](https://arxiv.org/abs/1411.1784) [cs.LG].
 - [13] J. Ho, A. Jain, and P. Abbeel, Denoising diffusion probabilistic models, in *Advances in Neural Information Processing Systems*, Vol. 33 (Curran Associates, Inc., 2020) pp. 6840–6851.
 - [14] P. Dhariwal and A. Nichol, Diffusion models beat gans on image synthesis (2021), [arXiv:2105.05233](https://arxiv.org/abs/2105.05233) [cs.LG].
 - [15] A. Radford, L. Metz, and S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks (2016), [arXiv:1511.06434](https://arxiv.org/abs/1511.06434) [cs.LG].
 - [16] T. Karras, S. Laine, and T. Aila, A style-based generator architecture for generative adversarial networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **43**, 4217 (2021).
 - [17] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, Generative adversarial text to image synthesis, in *Proceedings of The 33rd International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 48, edited by M. F. Balcan and K. Q. Weinberger (PMLR, New York, New York, USA, 2016) pp. 1060–1069.
 - [18] M. Kang, J.-Y. Zhu, R. Zhang, J. Park, E. Shechtman, S. Paris, and T. Park, Scaling up gans for text-to-image synthesis (2023), [arXiv:2303.05511](https://arxiv.org/abs/2303.05511) [cs.CV].
 - [19] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in *2017 IEEE International Conference on Computer Vision (ICCV)* (2017) pp. 2242–2251.
 - [20] Z. Yi, H. Zhang, P. Tan, and M. Gong, Dualgan: Unsuper-

- vised dual learning for image-to-image translation, in *2017 IEEE International Conference on Computer Vision (ICCV)* (IEEE Computer Society, Los Alamitos, CA, USA, 2017) pp. 2868–2876.
- [21] L. de Oliveira, M. Paganini, and B. Nachman, Learning particle physics by example: Location-aware generative adversarial networks for physics synthesis, *Computing and Software for Big Science* **1**, 10.1007/s41781-017-0004-6 (2017).
- [22] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, High-resolution image synthesis with latent diffusion models, in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE Computer Society, 2022) pp. 10674–10685.
- [23] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, *Nature* **549**, 195 (2017).
- [24] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms, *Nature Reviews Physics* **3**, 625–644 (2021).
- [25] H.-Y. Huang, M. Broughton, J. Cotler, S. Chen, J. Li, M. Mohseni, H. Neven, R. Babbush, R. Kueng, J. Preskill, and J. R. McClean, Quantum advantage in learning from experiments, *Science* **376**, 1182 (2022).
- [26] Y. Liu, S. Arunachalam, and K. Temme, A rigorous and robust quantum speed-up in supervised machine learning, *Nature Physics* **17**, 1013 (2021).
- [27] D. Aharonov, J. Cotler, and X.-L. Qi, Quantum algorithmic measurement, *Nature Communications* **13**, 1 (2022).
- [28] X. Gao, E. R. Anschuetz, S.-T. Wang, J. I. Cirac, and M. D. Lukin, Enhancing generative models via quantum correlations, *Phys. Rev. X* **12**, 021037 (2022).
- [29] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, Power of data in quantum machine learning, *Nature Communications* **12**, 2631 (2021).
- [30] Y. Wu, B. Wu, J. Wang, and X. Yuan, Quantum phase recognition via quantum kernel methods, *Quantum* **7**, 981 (2023).
- [31] A. Nietner, M. Ioannou, R. Sweke, R. Kueng, J. Eisert, M. Hinsche, and J. Haferkamp, On the average-case complexity of learning output distributions of quantum circuits, arXiv preprint [arXiv:2305.05765](https://arxiv.org/abs/2305.05765) (2023).
- [32] J. Tangpanitanon, S. Thanasilp, N. Dangniam, M.-A. Lemonde, and D. G. Angelakis, Expressibility and trainability of parametrized analog quantum systems for machine learning applications, *Physical Review Research* **2**, 043364 (2020).
- [33] M. Benedetti, D. Garcia-Pintos, O. Perdomo, V. Leyton-Ortega, Y. Nam, and A. Perdomo-Ortiz, A generative modeling approach for benchmarking and training shallow quantum circuits, *npj Quantum Information* **5**, 45 (2019).
- [34] M. S. Rudolph, S. Lerch, S. Thanasilp, O. Kiss, S. Vallecorsa, M. Grossi, and Z. Holmes, Trainability barriers and opportunities in quantum generative modeling (2023), [arXiv:2305.02881 \[quant-ph\]](https://arxiv.org/abs/2305.02881).
- [35] B. Coyle, D. Mills, V. Danos, and E. Kashefi, The born supremacy: quantum advantage and training of an ising born machine, *npj Quantum Information* **6**, 60 (2020).
- [36] J.-G. Liu and L. Wang, Differentiable learning of quantum circuit born machines, *Phys. Rev. A* **98**, 062324 (2018).
- [37] M. S. Rudolph, N. B. Toussaint, A. Katarbwa, S. Johri, B. Peropadre, and A. Perdomo-Ortiz, Generation of high-resolution handwritten digits with an ion-trap quantum computer, *Phys. Rev. X* **12**, 031010 (2022).
- [38] O. Kiss, M. Grossi, E. Kajomovitz, and S. Vallecorsa, Conditional born machine for monte carlo event generation, *Phys. Rev. A* **106**, 022612 (2022).
- [39] O. Kyriienko, A. E. Paine, and V. E. Elfving, Protocols for trainable and differentiable quantum generative modelling (2022), [arXiv:2202.08253 \[quant-ph\]](https://arxiv.org/abs/2202.08253).
- [40] C. Zoufal, A. Lucchi, and S. Woerner, Quantum generative adversarial networks for learning and loading random distributions, *npj Quantum Information* **5**, 103 (2019).
- [41] S. Y. Chang, S. Vallecorsa, E. F. Combarro, and F. Carminati, Quantum generative adversarial networks in a continuous-variable architecture to simulate high energy physics detectors (2021), [arXiv:2101.11132 \[quant-ph\]](https://arxiv.org/abs/2101.11132).
- [42] Chang, Su Yeon, Herbert, Steven, Vallecorsa, Sofia, Combarro, Elías F., and Duncan, Ross, Dual-parameterized quantum circuit gan model in high energy physics, *EPJ Web Conf.* **251**, 03050 (2021).
- [43] H.-L. Huang, Y. Du, M. Gong, Y. Zhao, Y. Wu, C. Wang, S. Li, F. Liang, J. Lin, Y. Xu, R. Yang, T. Liu, M.-H. Hsieh, H. Deng, H. Rong, C.-Z. Peng, C.-Y. Lu, Y.-A. Chen, D. Tao, X. Zhu, and J.-W. Pan, Experimental quantum generative adversarial networks for image generation, *Phys. Rev. Appl.* **16**, 024051 (2021).
- [44] A. Letcher, S. Woerner, and C. Zoufal, From tight gradient bounds for parameterized quantum circuits to the absence of barren plateaus in qgans (2023), [arXiv:2309.12681 \[quant-ph\]](https://arxiv.org/abs/2309.12681).
- [45] M. H. Amin, E. Andriyash, J. Rolfe, B. Kulchytskyy, and R. Melko, Quantum boltzmann machine, *Phys. Rev. X* **8**, 021050 (2018).
- [46] L. Coopmans and M. Benedetti, On the sample complexity of quantum boltzmann machine learning, [arXiv preprint arXiv:2306.14969](https://arxiv.org/abs/2306.14969) (2023).
- [47] J. Romero and A. Aspuru-Guzik, Variational quantum generators: Generative adversarial quantum machine learning for continuous distributions, *Advanced Quantum Technologies* **4**, 2000003 (2021).
- [48] C. Bravo-Prieto, J. Baglio, M. Cè, A. Francis, D. M. Grabowska, and S. Carrazza, Style-based quantum generative adversarial networks for Monte Carlo events, *Quantum* **6**, 777 (2022).
- [49] A. Barthe, M. Grossi, S. Vallecorsa, J. Tura, and V. Dunjko, Expressivity of parameterized quantum circuits for generative modeling of continuous multivariate distributions (2024), [arXiv:2402.09848 \[quant-ph\]](https://arxiv.org/abs/2402.09848).
- [50] K. Gili, M. Hibat-Allah, M. Mauri, C. Ballance, and A. Perdomo-Ortiz, Do quantum circuit born machines generalize?, *Quantum Science and Technology* **8**, 035021 (2023).
- [51] S. Basu, S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki, and R. Nemani, Deepsat: a learning framework for satellite imagery, in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '15 (Association for Computing Machinery, 2015).
- [52] M. Arjovsky, S. Chintala, and L. Bottou, Wasserstein gan (2017), [arXiv:1701.07875 \[stat.ML\]](https://arxiv.org/abs/1701.07875).
- [53] I. MacCormack, C. Delaney, A. Galda, N. Aggarwal, and P. Narang, Branching quantum convolutional neural networks, *Phys. Rev. Res.* **4**, 013117 (2022).
- [54] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre, Data re-uploading for a universal quantum classifier, *Quantum* **4**, 226 (2020).
- [55] P. Easom-McCaldin, A. Bouridane, A. Belatreche, and R. Jiang, On depth, robustness and performance using the data re-uploading single-qubit classifier, *IEEE Access* **9**, 65127

- (2021).
- [56] Y. Zeng, H. Wang, J. He, Q. Huang, and S. Chang, A multi-classification hybrid quantum neural network using an all-qubit multi-observable measurement strategy, *Entropy* **24** (2022).
- [57] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016) pp. 770–778.
- [58] K. Gili, M. Mauri, and A. Perdomo-Ortiz, Generalization metrics for practical quantum advantage in generative models (2023), [arXiv:2201.08770 \[cs.LG\]](https://arxiv.org/abs/2201.08770).
- [59] C. A. Riofrío, O. Mitevski, C. Jones, F. Krellner, A. Vučković, J. Doetsch, J. Klepsch, T. Ehmer, and A. Luckow, A performance characterization of quantum generative models (2023), [arXiv:2301.09363 \[quant-ph\]](https://arxiv.org/abs/2301.09363).
- [60] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, Improved techniques for training gans, in *Advances in Neural Information Processing Systems*, Vol. 29 (Curran Associates, Inc., 2016).
- [61] S. Barratt and R. Sharma, A note on the inception score (2018), [arXiv:1801.01973 \[stat.ML\]](https://arxiv.org/abs/1801.01973).
- [62] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, Gans trained by a two time-scale update rule converge to a local nash equilibrium, in *Advances in Neural Information Processing Systems*, Vol. 30 (Curran Associates, Inc., 2017).
- [63] Y. Liu and Y. Li, Metrics of GANs, <https://github.com/yhlleo/GAN-Metrics> (2021), [Online; accessed Nov-23-2022].
- [64] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, Rethinking the inception architecture for computer vision, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016) pp. 2818–2826.
- [65] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, *Proceedings of the IEEE* **86**, 2278 (1998).
- [66] H. Xiao, K. Rasul, and R. Vollgraf, Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms (2017), [arXiv:1708.07747 \[cs.LG\]](https://arxiv.org/abs/1708.07747).
- [67] J. Wei, M. Liu, J. Luo, A. Zhu, J. Davis, and Y. Liu, Duelgan: A duel between two discriminators stabilizes the gan training, in *Computer Vision – ECCV 2022* (Springer Nature Switzerland, 2022) pp. 290–317.
- [68] D. Lazcano, N. F. Franco, and W. Creixell, Hgan: Hyperbolic generative adversarial network, *IEEE Access* **9**, 96309 (2021).
- [69] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, *JAX: composable transformations of Python+NumPy programs* (2018).
- [70] J. Heek, A. Levsikaya, A. Oliver, M. Ritter, B. Rondepierre, A. Steiner, and M. van Zee, *Flax: A neural network library and ecosystem for JAX* (2020).
- [71] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M. S. Alam, G. Alonso-Linaje, B. AkashNarayanan, A. Asadi, J. M. Arrazola, U. Azad, S. Banning, C. Blank, T. R. Bromley, B. A. Cordier, J. Ceroni, A. Delgado, O. D. Matteo, A. Dusko, T. Garg, D. Guala, A. Hayes, R. Hill, A. Ijaz, T. Isaacsson, D. Ittah, S. Jahangiri, P. Jain, E. Jiang, A. Khandelwal, K. Kottmann, R. A. Lang, C. Lee, T. Loke, A. Lowe, K. McKiernan, J. J. Meyer, J. A. Montañez-Barrera, R. Moyard, Z. Niu, L. J. O’Riordan, S. Oud, A. Panigrahi, C.-Y. Park, D. Polatajko, N. Quesada, C. Roberts, N. Sá, I. Schoch, B. Shi, S. Shu, S. Sim, A. Singh, I. Strandberg, J. Soni, A. Száva, S. Thabet, R. A. Vargas-Hernández, T. Vincent, N. Vitucci, M. Weber, D. Wierichs, R. Wiersema, M. Willmann, V. Wong, S. Zhang, and N. Killoran, *Pennylane: Automatic differentiation of hybrid quantum-classical computations* (2022), [arXiv:1811.04968 \[quant-ph\]](https://arxiv.org/abs/1811.04968).
- [72] Image Generation on Fashion-MNIST, <https://paperswithcode.com/sota/image-generation-on-fashion-mnist>, [Online; accessed Apr-26-2024].
- [73] V. Böhm and U. Seljak, Probabilistic autoencoder (2022), [arXiv:2006.05479 \[cs.LG\]](https://arxiv.org/abs/2006.05479).
- [74] L. van der Maaten and G. Hinton, Visualizing data using t-sne, *Journal of Machine Learning Research* **9**, 2579 (2008).
- [75] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay, Scikit-learn: Machine learning in python, *Journal of Machine Learning Research* **12**, 2825 (2011).
- [76] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, The power of quantum neural networks, *Nature Computational Science* **1**, 403 (2021).
- [77] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, *Nature Communications* **9**, 4812 (2018).
- [78] M. Larocca, S. Thanasilp, S. Wang, K. Sharma, J. Biamonte, P. J. Coles, L. Cincio, J. R. McClean, Z. Holmes, and M. Cerezo, A review of barren plateaus in variational quantum computing (2024), [arXiv:2405.00781 \[quant-ph\]](https://arxiv.org/abs/2405.00781).
- [79] A. Arrasmith, Z. Holmes, M. Cerezo, and P. J. Coles, Equivalence of quantum barren plateaus to cost concentration and narrow gorges, *Quantum Science and Technology* **7**, 045015 (2022).
- [80] Z. Holmes, K. Sharma, M. Cerezo, and P. J. Coles, Connecting ansatz expressibility to gradient magnitudes and barren plateaus, *PRX Quantum* **3**, 010313 (2022).
- [81] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, Cost function dependent barren plateaus in shallow parametrized quantum circuits, *Nature Communications* **12**, 1791 (2021).
- [82] S. Wang, E. Fontana, M. Cerezo, K. Sharma, A. Sone, L. Cincio, and P. J. Coles, Noise-induced barren plateaus in variational quantum algorithms, *Nature Communications* **12**, 6961 (2021).
- [83] C. O. Marrero, M. Kieferová, and N. Wiebe, Entanglement-induced barren plateaus, *PRX Quantum* **2**, 040316 (2021).
- [84] T. L. Patti, K. Najafi, X. Gao, and S. F. Yelin, Entanglement devised barren plateau mitigation, *Physical Review Research* **3**, 033090 (2021).
- [85] M. Larocca, P. Czarnik, K. Sharma, G. Muraleedharan, P. J. Coles, and M. Cerezo, Diagnosing Barren Plateaus with Tools from Quantum Optimal Control, *Quantum* **6**, 824 (2022).
- [86] Z. Holmes, A. Arrasmith, B. Yan, P. J. Coles, A. Albrecht, and A. T. Sornborger, Barren plateaus preclude learning scramblers, *Physical Review Letters* **126**, 190501 (2021).
- [87] S. Thanasilp, S. Wang, N. A. Nghiem, P. Coles, and M. Cerezo, Subtleties in the trainability of quantum machine learning models, *Quantum Machine Intelligence* **5**, 21 (2023).
- [88] M. Cerezo, M. Larocca, D. García-Martín, N. L. Diaz, P. Braccia, E. Fontana, M. S. Rudolph, P. Bermejo, A. Ijaz, S. Thanasilp, E. R. Anschuetz, and Z. Holmes, Does provable absence of barren plateaus imply classical simulability? or, why we need to rethink variational quantum computing (2023), [arXiv:2312.09121 \[quant-ph\]](https://arxiv.org/abs/2312.09121).
- [89] M. Ragone, B. N. Bakalov, F. Sauvage, A. F. Kemper, C. O. Marrero, M. Larocca, and M. Cerezo, A unified theory of bar-

- ren plateaus for deep parametrized quantum circuits (2023), [arXiv:2309.09342](https://arxiv.org/abs/2309.09342) [quant-ph].
- [90] E. Fontana, D. Herman, S. Chakrabarti, N. Kumar, R. Yalovetzky, J. Heredge, S. Hari Sureshbabu, and M. Pistoia, The adjoint is all you need: Characterizing barren plateaus in quantum ansätze, [arXiv preprint arXiv:2309.07902](https://arxiv.org/abs/2309.07902) (2023).
- [91] N. L. Diaz, D. García-Martín, S. Kazi, M. Larocca, and M. Cerezo, Showcasing a barren plateau theory beyond the dynamical lie algebra, [arXiv preprint arXiv:2310.11505](https://arxiv.org/abs/2310.11505) (2023).
- [92] L. Leone, S. F. Oliviero, L. Cincio, and M. Cerezo, On the practical usefulness of the hardware efficient ansatz, [arXiv preprint arXiv:2211.01477](https://arxiv.org/abs/2211.01477) (2022).
- [93] A. Barthe and A. Pérez-Salinas, Gradients and frequency profiles of quantum re-uploading models, [arXiv preprint arXiv:2311.10822](https://arxiv.org/abs/2311.10822) (2023).
- [94] S. Thanasilp, S. Wang, M. Cerezo, and Z. Holmes, Exponential concentration in quantum kernel methods, [arXiv preprint arXiv:2208.11060](https://arxiv.org/abs/2208.11060) (2022).
- [95] W. Xiong, G. Facelli, M. Sahebi, O. Agnel, T. Chotibut, S. Thanasilp, and Z. Holmes, On fundamental aspects of quantum extreme learning machines, [arXiv preprint arXiv:2312.15124](https://arxiv.org/abs/2312.15124) (2023).
- [96] Y. Suzuki, H. Kawaguchi, and N. Yamamoto, Quantum fisher kernel for mitigating the vanishing similarity issue, [arXiv preprint arXiv:2210.16581](https://arxiv.org/abs/2210.16581) (2022).
- [97] Y. Suzuki and M. Li, Effect of alternating layered ansatzes on trainability of projected quantum kernel, [arXiv preprint arXiv:2310.00361](https://arxiv.org/abs/2310.00361) (2023).
- [98] Z. Rychlik, A central limit theorem for sums of a random number of independent random variables, in *Colloquium Mathematicum*, Vol. 1 (1976) pp. 147–158.
- [99] E. Grant, L. Wossnig, M. Ostaszewski, and M. Benedetti, An initialization strategy for addressing barren plateaus in parametrized quantum circuits, *Quantum* **3**, 214 (2019).
- [100] K. Zhang, L. Liu, M.-H. Hsieh, and D. Tao, Escaping from the barren plateau via gaussian initializations in deep variational quantum circuits (2022), [arXiv:2203.09376](https://arxiv.org/abs/2203.09376) [quant-ph].
- [101] IBM Quantum EfficientSU2, <https://docs.quantum.ibm.com/api/qiskit/qiskit.circuit.library.EfficientSU2>, [Online; accessed Mar-15-2024].
- [102] R. P. i Valls, M. Drudis, S. Thanasilp, and Z. Holmes, Variational quantum simulation: a case study for understanding warm starts (2024), [arXiv:2404.10044](https://arxiv.org/abs/2404.10044) [quant-ph].
- [103] M. S. Rudolph, J. Miller, D. Motlagh, J. Chen, A. Acharya, and A. Perdomo-Ortiz, Synergistic pretraining of parametrized quantum circuits via tensor networks, *Nature Communications* **14**, 8367 (2023).
- [104] A. A. Mele, G. B. Mbeng, G. E. Santoro, M. Collura, and P. Torta, Avoiding barren plateaus via transferability of smooth solutions in a hamiltonian variational ansatz, *Physical Review A* **106**, L060401 (2022).
- [105] J. Landman, S. Thabet, C. Dalyac, H. Mhiri, and E. Kashefi, Classically approximating variational quantum machine learning with random fourier features (2022), [arXiv:2210.13200](https://arxiv.org/abs/2210.13200) [quant-ph].
- [106] T. Karras, T. Aila, S. Laine, and J. Lehtinen, Progressive growing of gans for improved quality, stability, and variation (2018), [arXiv:1710.10196](https://arxiv.org/abs/1710.10196) [cs.NE].
- [107] S. Lloyd and C. Weedbrook, Quantum generative adversarial learning, *Phys. Rev. Lett.* **121**, 040502 (2018).
- [108] J. Zeng, Y. Wu, J.-G. Liu, L. Wang, and J. Hu, Learning and inference on generative adversarial quantum circuits, *Phys. Rev. A* **99**, 052306 (2019).
- [109] A. Assouel, A. Jacquier, and A. Kondratyev, A quantum generative adversarial network for distributions, *Quantum Machine Intelligence* **4**, 28 (2022).
- [110] M. Schuld and N. Killoran, Quantum machine learning in feature hilbert spaces, *Phys. Rev. Lett.* **122**, 040504 (2019).
- [111] J. Li, R. O. Topaloglu, and S. Ghosh, Quantum generative models for small molecule drug discovery, *IEEE Transactions on Quantum Engineering* **2**, 1 (2021).
- [112] S. Tsang, M. T. West, S. M. Erfani, and M. Usman, Hybrid quantum–classical generative adversarial network for high-resolution image generation, *IEEE Transactions on Quantum Engineering* **4**, 1 (2023).

Appendix A: Related Work

In this section, we provide a brief summary of the recent research on classical and quantum generative models. Especially, we underline the difference between the discrete and the continuous quantum GAN. Table IV summarize the characteristics of the two different generative models for comparison.

	Discrete Quantum Generative Models	Continuous Quantum Generative Models
Task	Encode a probability distribution over discrete values	Generate continuous outputs
Outputs	Discrete bit strings	Continuous values
Sample Complexity	One measurement per sample	Set of measurements per samples
Randomness	Use the probabilistic nature of quantum physics (quantum randomness)	Sampled from a classical random distribution (classical randomness)
Projector	$ x\rangle\langle x $	Estimate a vector of expectation values
Output size	$\mathcal{O}(2^n)$	$\mathcal{O}(n)$
Examples	Quantum Circuit Born Machine (QCBM) [33] Quantum Generative Adversarial Networks (qGAN) [40]	Variational Quantum Generator (VQG) [47] Style-based Quantum GAN [48]

Table IV. **Comparison between discrete and continuous quantum generative models.** The discrete generative model treats each quantum measurement as a single output, focusing on learning the probability distribution across computational basis states. In contrast, the continuous model uses expectation values obtained from multiple shots, embedding external classical noise into the quantum circuit to generate samples.

Classical generative models for image generation. GANs were proposed by I. Goodfellow in 2014 as an effective way of learning to generate data which follow a given distribution [6]. It consists of two neural networks competing with each other: a generator and a discriminator of fake data. Being a successful generative model for the creation of realistic data and images, variations of GANs were also explored, such as conditional GAN [12] to generate data of a given class, the more stable Wasserstein GAN [52], and Style-GAN [106] for detailed image generation.

Recently, Diffusion Models (DMs) proved to be powerful alternative generative models, trained by injecting noise into the images and then learning the reverse process to remove it [7, 13]. In the recent work, Rombach *et al.* introduced Latent DM, which operate on the latent space instead of the image space by mapping the image to a lower-dimensional latent space and learning the latent representation to reduce the complexity of the model and improve the visual fidelity [22].

Quantum GANs for discrete data The introduction of quantum GANs by Ref. [107] has suggested the possibility of learning the hidden statistics of a quantum or classical data set based on the intrinsically probabilistic nature of quantum systems. For example, C. Zoufal *et al.* introduced a hybrid GAN model with a classical discriminator and a n -qubit quantum generator which can efficiently learn a classical probability distribution over 2^n discrete variables with QNNs [40]. They have demonstrated that the model can be used for an efficient initialization of a quantum state with an arbitrary probability distribution, one of the most crucial challenges in quantum computing, or even used for a realistic use case such as finance. This discrete quantum GAN handles the computational basis of the quantum circuit Hilbert space as discrete data and explicitly constructs the probability distribution over them by performing a set of measurements at the end of the quantum generator. A similar strategy was applied to mimic the prototypical Bars-and-Stripes dataset images in Ref. [108]. Assouel *et al.* also proposed a quantum GAN model, which is called QuGAN, for discrete data generation in the context of finance but using a quantum discriminator directly connected to the quantum generator [109].

Quantum generative models for continuous data generation. As a complement to the previously presented studies, which focus on reproducing the probability distribution over discrete data, several papers studied constructing quantum GANs to learn the hidden data distribution over continuous data. Romero and Aspuru-Guzik introduce the idea of a quantum generator to learn a continuous distribution with latent noise embedded via rotation gates [47]. Unlike the discrete quantum GAN which explicitly generates the probability distribution over the discrete data, the continuous GANs work in a similar way as the classical GAN by generating samples at the end of the generator. The classical latent noises are embedded into the quantum circuit by an encoding process, the so-called *quantum feature map* or *quantum encoding* [110]. We measure the expectation value of quantum observables such as Pauli operators ($\sigma_x, \sigma_y, \sigma_z$), which are continuous by definition, to construct the output sample for each latent noise. This leads to learning the hidden data distribution from the continuous latent noise distribution in an implicit way. In Ref. [48], C. Bravo Prieto *et al.* employed a style-based quantum generator for Monte Carlo event generation, proving that the quantum GAN is able to reproduce a highly correlated multi-dimensional probability distribution. The particularity of this architecture is that the latent noises are embedded in the rotation angles of the learning layers via an affine transformation, with the weights and biases updated during the training. Instead of using a purely quantum generator, the possibility of using a hybrid generator (HG) has also been proposed by J. Li *et al.* for small molecule drug discovery [111]. The proposed QGAN-HG architecture consists of a quantum circuit attached to a classical layer with the latent noise embedded with single qubit gates. It showed a learning accuracy comparable to the classical MolGAN with 98% reduced parameters.

Quantum generative models for image generation. In the realm of image generation, quantum patch GAN has been proposed to generate the patches in images using a set of subgenerator sequence [43, 112]. However, their applications were only tested to a limited number of classes in MNIST [43] or in FashionMNIST dataset [112].

Instead of using a quantum circuit as a data generator, certain studies propose the possibility of using it as an additional component in GAN to improve performance. For example, Rudolph *et al.* suggested a hybrid GANs schema where the prior distribution of the classical GAN is generated by a Quantum Circuit Born Machine (QCBM) [37]. The architecture showed an ability to generate high-quality MNIST images on discretized latent space with up to $2^{16} = 65536$ samples using 8 qubits.

Appendix B: Architecture of Classical Autoencoder

On Figure 13, we display the classical convolutional autoencoder used to reduce the image dimension. The model is trained based on the Mean Squared Error (MSE) loss using the Adam optimizer with a learning rate of 0.001 for 100 epochs.

Figure 14 display the original and the reconstructed images obtained using the suggested autoencoder architecture with latent space of dimension 20 for MNIST and FashionMNIST datasets. The reconstructed images are slightly blurry compared to the original ones, with a loss of details, but their general shape is recovered.

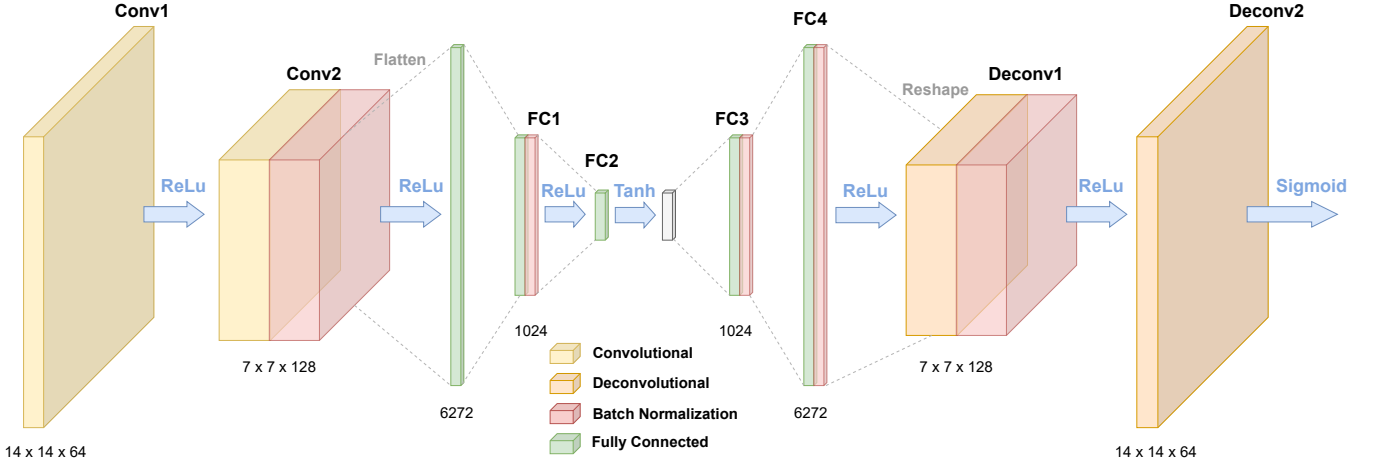


Figure 13. **Architecture of the convolutional autoencoder used in the paper.** We apply the Tanh activation function at the end of the encoder to ensure that the latent features are confined within the range of $[-1, 1]$. The autoencoder is pre-trained on the original image datasets with $28 \times 28 \times 1$ pixels, following the format of *width* \times *height* \times *channel*.

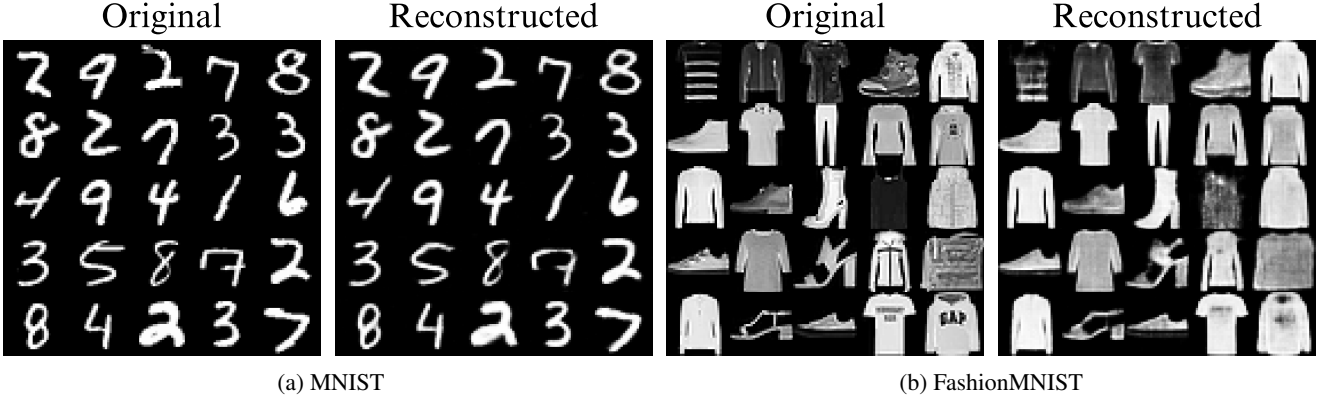


Figure 14. **Comparison of original and reconstructed images utilizing the autoencoder in Figure 13.** Employing a 20-dimensional latent space, the reconstructed images maintain their overall form, with a loss of finer details.

Appendix C: Proof on Absence of Barren Plateau

In Section V of the main text, we have numerically demonstrated that the identity initialization with normally distributed quantum circuit parameters mitigates the barren plateau of the generator loss function of the form :

$$\mathcal{L}_G = - \mathbb{E}_{\mathbf{z} \sim \mathbb{P}_z} [D_\phi(G_\theta(\mathbf{z}))] \tag{C1}$$

where $G_\theta(\mathbf{z})$ is a vector of the expectation value with a classical input \mathbf{z}_i . In this section, we provide analytical insight into the absence of a barren plateau using the small angle initialization mentioned in Section V. To do so, we leverage the most general form of the quantum circuit, without any style-based architecture. The generator loss function can be seen as a function depending on the generator outputs, post-processed with the classical discriminator. Hence, under the realistic assumption that the classical discriminator does not exhibit any vanishing gradient, it is enough to prove that the generator outputs do not decay exponentially to imply the absence of barren plateau for the generator loss function $\mathcal{L}_G(\theta)$. In other words, we need to find a polynomial decaying lower bound for an expectation value of the quantum circuit with a certain observable. Furthermore, in Section V, while we sampled the trainable parameters W_ℓ and \mathbf{b}_ℓ from a uniform distribution, the final rotation angles θ_ℓ in the style-based quantum circuits are normally distributed due to the Central Limit Theorem [98]. Consequently, the analysis of the barren plateau phenomenon in the style-based architecture can be considered equivalent when the parameters are initialized using a normal distribution.

From now on, we take the loss function of the form $\mathcal{L}(\boldsymbol{\theta}) = \langle \psi_0 | U^\dagger(\boldsymbol{\theta}) O U(\boldsymbol{\theta}) | \psi_0 \rangle = \text{Tr}(O U(\boldsymbol{\theta}) \rho U^\dagger(\boldsymbol{\theta}))$ with $U(\boldsymbol{\theta})$ an arbitrary quantum circuit, O the observable, $|\psi_0\rangle$ the initial state, and ρ its corresponding density matrix. We evaluate the lower bound on the variance of the loss function for a normal initialization $\mathcal{N}(0, \sigma)$ with zero mean and standard deviation σ , following an argument similar to that given in the appendix of Ref. [44] for a uniform initialization $[-\pi, \pi]$. Normal initialization has been suggested as a strategy to escape barren plateau in the prior study [100]. As an extension of this research, we will introduce a lower bound on the variance, linked to the types of the initial states and the measurement operator in general circuits, building a more rigorous understanding of the loss decay. In particular, we compute a tight lower bound for a specific quantum circuit ansatz, EfficientSU2 [101], and prove it numerically.

Let us denote $P_\alpha \in \mathbf{P}_n$ the Pauli string which consists of n single-qubit Pauli matrices written as $\sigma_i \in \{\mathbb{I}, X, Y, Z\} = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3\}$:

$$P_\alpha = \bigotimes_{i=1}^n \sigma_{\alpha_i}, \quad \alpha \in \{0, 1, 2, 3\}^n, \quad (\text{C2})$$

where α_i is the i -th component of α . We use a bold symbol α in order to clarify that it is a vector of n indices. Similarly, the Pauli string rotation gates $P_\alpha(\theta)$ with shorthand notation:

$$P_\alpha(\theta) := R_{P_\alpha}(\theta) = \exp\left(-i P_\alpha \frac{\theta}{2}\right) = \cos(\theta) \mathbb{I} - i \sin(\theta) P_\alpha, \quad (\text{C3})$$

for some $\theta \in \mathbb{R}$. We consider a general quantum ansatz $U_L(\boldsymbol{\Theta})$ [44] with trainable parameters $\boldsymbol{\Theta}$, which consists of two orthogonal layers of single-qubit rotations for state initialization, V_1 and V_2 , and an entangling layer W_L of depth L :

$$U_L(\boldsymbol{\Theta}) = W_L(\boldsymbol{\theta}) V_2(\boldsymbol{\phi}) V_1(\boldsymbol{\omega}), \quad \boldsymbol{\Theta} = (\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}). \quad (\text{C4})$$

with

$$V_1(\boldsymbol{\omega}) = \bigotimes_{i=1}^n \sigma_{\mu_i}(\omega_i), \quad V_2(\boldsymbol{\phi}) = \bigotimes_{i=1}^n \sigma_{\nu_i}(\phi_i), \quad W(\boldsymbol{\theta}) = \prod_{\ell=1}^L \tilde{W}_{\ell(K_\ell)}(\boldsymbol{\theta}_\ell) = \prod_{\ell=1}^L \left(\prod_{i=1}^{K_\ell} P_{\ell,i}(\theta_{\ell,i}) \right) C_\ell \quad (\text{C5})$$

where C_ℓ are n -qubit Clifford gates, $P_{\ell,i}$ the i^{th} rotation gate and K_ℓ the number of Pauli rotation gate at layer ℓ . It is important to note that $\boldsymbol{\Theta}$ being the collection of rotation angles in U_L does not correspond to the generator parameters defined in Section II for the style-based architecture. By definition of orthogonality, we ensure that $\mu_i, \nu_i \in \{1, 2, 3\}$ with $\mu_i \neq \nu_i$ for all i . Additionally, we take into account the most general form of the loss function with the observable $O = \sum_\alpha a_\alpha P_\alpha$:

$$\mathcal{L} = \text{Tr}(O U(\boldsymbol{\Theta}) \rho U^\dagger(\boldsymbol{\Theta})) = \sum_\alpha a_\alpha \text{Tr}(P_\alpha U(\boldsymbol{\Theta}) \rho U^\dagger(\boldsymbol{\Theta})) = \sum_\alpha a_\alpha \mathcal{L}_\alpha, \quad (\text{C6})$$

where ρ is the initial density matrix, decomposed as a sum of Pauli strings:

$$\rho = \frac{1}{2^n} \sum_\lambda c_\lambda P_\lambda, \quad c_\lambda \in \mathbb{R}. \quad (\text{C7})$$

In particular, if ρ is a product state, c_λ should be less or equal to 1 for all λ .

1. Expectation value with normal initialization

a. $L = 0$ case

To start with, let us calculate the expectation value of \mathcal{L} over the initial parameters $\boldsymbol{\Theta} \sim \mathcal{N}(0, \sigma)$ ¹. By linearity of the expectation, we have $\mathbb{E}_\Theta[\mathcal{L}] = \sum_\alpha a_\alpha \mathbb{E}_\Theta[\mathcal{L}_\alpha]$ from Eq. (C6), and thus, it is enough to compute $\mathbb{E}_\Theta[\mathcal{L}_\alpha]$ to find the final formula. We first consider the base case with $L = 0$ without any entangling layer W_L . The loss function \mathcal{L}_α can be explicitly

¹ To avoid potential confusion, we clarify the notation used in this paper: σ_{α_i} denotes the Pauli matrices, while σ (without subscript) represents the standard deviation. The presence of the subscript α_i distinguishes the Pauli matrix notation from the standard deviation symbol σ .

written as a product of the loss functions on each qubit :

$$\begin{aligned}\mathcal{L}_\alpha &= \text{Tr}(P_\alpha U(\Theta)\rho U^\dagger(\Theta)) = \frac{1}{2^n} \sum_\lambda c_\lambda \text{Tr}(U^\dagger(\Theta)P_\alpha U(\Theta)P_\lambda) \\ &= \sum_\lambda c_\lambda \prod_i \frac{1}{2} \text{Tr}(\sigma_{\mu_i}(-\omega_i)\sigma_{\nu_i}(-\phi_i)\sigma_{\alpha_i}\sigma_{\nu_i}(\phi_i)\sigma_{\mu_i}(\omega_i)\sigma_{\lambda_i}) = \sum_\lambda c_\lambda \prod_i \mathcal{L}_{\alpha\lambda}^i,\end{aligned}\quad (\text{C8})$$

where $\mathcal{L}_{\alpha\lambda}^i$ is the loss defined on each qubit i . On one hand, if $\sigma_{\alpha_i} = \mathbb{I}$ or $\sigma_{\lambda_i} = \mathbb{I}$ with $\alpha_i \neq \lambda_i$, it is evident that $\mathcal{L}_{\alpha\lambda}^i = 0$. On the other hand, if $\sigma_{\alpha_i} = \sigma_{\lambda_i} = \mathbb{I}$, we have $\mathcal{L}_{\alpha\lambda}^i = 1$. Thus, we only need to consider the qubits where α_j and λ_j are both non-trivial.

In case of the uniform initialization over $[-\pi, \pi]$, it is straightforward to show that $\mathbb{E}_\Theta[\mathcal{L}_{\alpha\lambda}^i]$ vanishes due to the periodicity of the trigonometric functions [44]. However, the calculation requires a bit more effort in the case of the normal initialization as the results vary depending on μ_j, ν_j, α_j and λ . First of all, if $\alpha_j = \nu_j$, then σ_{α_j} commutes with $\sigma_{\nu_j}(\phi_j)$ and Eq. (C8) simplifies into

$$\begin{aligned}\mathcal{L}_{\alpha\lambda}^j &= \frac{1}{2} \text{Tr}(\sigma_{\mu_j}(-\omega_j)\sigma_{\alpha_j}\sigma_{\mu_j}(\omega_j)\sigma_{\lambda_j}) = \frac{1}{2} \text{Tr}(\sigma_{\mu_j}(-2\omega_j)\sigma_{\alpha_j}\sigma_{\lambda_j}) \\ &= \frac{1}{2} \cos(\omega_j) \text{Tr}(\sigma_{\alpha_j}\sigma_{\lambda_j}) + \frac{i}{2} \sin(\omega_j) \text{Tr}(\sigma_{\mu_j}\sigma_{\alpha_j}\sigma_{\lambda_j})\end{aligned}\quad (\text{C9})$$

recalling that $\sigma_\beta\sigma_\gamma(\omega) = \sigma_\gamma(-\omega)\sigma_\beta$ for $\beta \neq \gamma$. We have three different cases :

$$\mathcal{L}_{\alpha\lambda}^j = \begin{cases} \frac{1}{2} \cos(\omega_j) \text{Tr}(\mathbb{I}) + \frac{i}{2} \sin(\omega_j) \text{Tr}(\sigma_{\mu_j}) = \cos(\omega_j) & \alpha_j = \lambda_j \\ -\frac{i}{2} \sin(\omega_j) \text{Tr}(\sigma_{\mu_j}\sigma_{\lambda_j}\sigma_{\alpha_j}) = -\frac{i}{2} \sin(\omega_j) \text{Tr}(\sigma_{\alpha_j}) = 0 & \alpha_j \neq \lambda_j = \mu_j \\ -\frac{i}{2} \sin(\omega_j) \text{Tr}(\sigma_{\mu_j}\sigma_{\lambda_j}\sigma_{\alpha_j}) = -\frac{i}{2} \frac{1}{2} \sin(\omega_j) \text{Tr}(\pm 2i\sigma_{\mu_j}\sigma_{\mu_j}) = \pm \sin(\omega_j) & \alpha_j \neq \lambda_j \neq \mu_j \end{cases} \quad (\text{C10})$$

The third line is derived from the fact that $[\sigma_{\lambda_j}, \sigma_{\alpha_j}] = \pm 2i\sigma_{\mu_j}$ as $\alpha_j \neq \mu_j$ by orthogonality. From the equation above, the loss function can be summarized as :

$$\mathcal{L}_{\alpha\lambda}^j = \cos(\omega_j)\delta_{\alpha_j\lambda_j} \pm \sin(\omega_j)\bar{\delta}_{\alpha_j\lambda_j}\bar{\delta}_{\lambda_j\mu_j}, \quad (\text{C11})$$

where we denote $\bar{\delta}_{ij} = 1 - \delta_{ij}$, i.e. $\bar{\delta}_{ij} = 1$ if $i \neq j$ and 0 if $i = j$.

Let's consider the normal initialization of the parameters with the mean $\mu = 0$ and the standard deviation σ . If we take the expectation value $\mathbb{E}_{\omega_j}[\mathcal{L}_{\alpha\lambda}]$ over the parameter space ω_j , the contribution of the $\sin(\omega_j)$ will cancel out, because $\sin(\omega_j)$ is an odd function at $\omega_j = 0$ over \mathbb{R} , while the Gaussian distribution is an even function, leaving only the contribution of the even function $\cos(\omega_j)$. Thus, $\mathbb{E}_{\omega_j}[\mathcal{L}_{\alpha\lambda}^j]$ can be written as :

$$\mathbb{E}_{\omega_j}[\mathcal{L}_{\alpha\lambda}^j] = \delta_{\alpha_j\lambda_j} \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\infty} \cos(\omega_j) e^{-\frac{\omega_j^2}{2\sigma^2}} d\omega_j = e^{-\frac{\sigma^2}{2}} \delta_{\alpha_j\lambda_j}. \quad (\text{C12})$$

Similarly, in the case of $\alpha_j \neq \nu_j$, each term in Eq. (C8) is written as :

$$\mathcal{L}_{\alpha\lambda}^j = \frac{1}{2} \cos(\phi_j) \text{Tr}(\sigma_{\mu_j}(-\omega_j)\sigma_{\alpha_j}\sigma_{\mu_j}(\omega_j)\sigma_{\lambda_j}) + \frac{i}{2} \sin(\phi_j) \text{Tr}(\sigma_{\mu_j}(-\omega_j)\sigma_{\nu_j}\sigma_{\alpha_j}\sigma_{\mu_j}(\omega_j)\sigma_{\lambda_j}). \quad (\text{C13})$$

We should distinguish two different cases: 1) $\alpha_j = \mu_j$ and 2) $\alpha_j \neq \mu_j$. In 1) σ_{α_j} and $\sigma_{\mu_j}(\omega_j)$ commute, hence :

$$\begin{aligned}\mathcal{L}_{\alpha\lambda}^j &= \frac{1}{2} \cos(\phi_j) \text{Tr}(\sigma_{\alpha_j}\sigma_{\lambda_j}) + \frac{i}{2} \sin(\phi_j) \text{Tr}(\sigma_{\mu_j}(-2\omega_j)\sigma_{\nu_j}\sigma_{\alpha_j}\sigma_{\lambda_j}) \\ &= \frac{1}{2} \cos(\phi_j) \text{Tr}(\sigma_{\alpha_j}\sigma_{\lambda_j}) + \frac{i}{2} \sin(\phi_j) (\cos(\omega_j) \text{Tr}(\sigma_{\nu_j}\sigma_{\alpha_j}\sigma_{\lambda_j}) + i \sin(\omega_j) \text{Tr}(\sigma_{\mu_j}\sigma_{\nu_j}\sigma_{\alpha_j}\sigma_{\lambda_j})) \\ &= \begin{cases} \cos(\phi_j) & \alpha_j = \lambda_j \\ \sin(\phi_j) \sin(\omega_j) & \alpha_j \neq \lambda_j = \nu_j \\ \pm \sin(\phi_j) \cos(\omega_j) & \alpha_j \neq \lambda_j \neq \nu_j \end{cases} \quad (\text{C14})\end{aligned}$$

When we compute the expectation value, the last two cases vanish due to the parity over ϕ_j . On the other hand, in 2) we have :

$$\begin{aligned}\mathcal{L}_{\alpha\lambda}^j &= \frac{1}{2} \cos(\phi_j) \text{Tr}(\sigma_{\mu_j}(-2\omega_j)\sigma_{\alpha_j}\sigma_{\lambda_j}) + \frac{i}{2} \sin(\phi_j) \text{Tr}(\sigma_{\nu_j}\sigma_{\alpha_j}\sigma_{\lambda_j}), \\ &= \frac{1}{2} \cos(\phi_j) \cos(\omega_j) \text{Tr}(\sigma_{\alpha_j}\sigma_{\lambda_j}) + \frac{i}{2} \cos(\phi_j) \sin(\omega_j) \text{Tr}(\sigma_{\mu_j}\sigma_{\alpha_j}\sigma_{\lambda_j}) + \frac{i}{2} \sin(\phi_j) \text{Tr}(\sigma_{\nu_j}\sigma_{\alpha_j}\sigma_{\lambda_j}) \\ &= \begin{cases} \cos(\phi_j) \cos(\omega_j) & \alpha_j = \lambda_j \\ \pm \sin(\phi_j) & \alpha_j \neq \lambda_j = \mu_j \\ \pm \cos(\phi_j) \sin(\omega_j) & \alpha_j \neq \lambda_j = \nu_j \end{cases} \end{aligned} \quad (\text{C15})$$

Similar to before, the expectation value of the last two terms vanishes over ϕ_j and ω_j due to parity, leading to :

$$\mathbb{E}_{\omega_j, \phi_j}[\mathcal{L}_{\alpha\lambda}^j] = \begin{cases} e^{-\frac{\sigma^2}{2}} \delta_{\alpha_j \lambda_j} & \mu_j = \alpha_j \\ \left(e^{-\frac{\sigma^2}{2}}\right)^2 \delta_{\alpha_j \lambda_j} & \mu_j \neq \alpha_j \end{cases} \quad (\text{C16})$$

Combining Eq. (C12) and Eq. (C16), the expectation value $\mathbb{E}_{\Theta}[\mathcal{L}_{\alpha}^i]$ can be summarized as :

$$\mathbb{E}_{\Theta}[\mathcal{L}_{\alpha\lambda}^j] = \left(e^{-\frac{\sigma^2}{2}}\right)^{1+\bar{\delta}_{\alpha_j \mu_j} \bar{\delta}_{\alpha_j \nu_j}} \delta_{\alpha_j \lambda_j}. \quad (\text{C17})$$

From now on, we define $\mathcal{J}_{\alpha} = \{j \mid \alpha_j \neq 0\}$ as a set of the qubit indices where the Pauli string P_{α} has non-trivial Pauli matrices and the weight of P_{α} , denoted as $w(P_{\alpha}) := |\mathcal{J}_{\alpha}|$. Then, the expectation of the final loss across the whole input state ρ (c.f. Eq. (C8)) will be :

$$\mathbb{E}_{\Theta}[\mathcal{L}_{\alpha}] = \sum_{\lambda} c_{\lambda} \prod_j \mathbb{E}_{\Theta}[\mathcal{L}_{\alpha\lambda}^j] = \sum_{\lambda} c_{\lambda} \left(e^{-\frac{\sigma^2}{2}}\right)^{w(P_{\alpha})} \prod_{j \in \mathcal{J}_{\alpha}} \delta_{\alpha_j \lambda_j} \left(e^{-\frac{\sigma^2}{2}}\right)^{\bar{\delta}_{\alpha_j \mu_j} \bar{\delta}_{\alpha_j \nu_j}} = K_{\alpha}(c_{\lambda}) \left(e^{-\frac{\sigma^2}{2}}\right)^{w(P_{\alpha})}, \quad (\text{C18})$$

where $K_{\alpha}(c_{\lambda}) = c_{\lambda^*}$ if there exists λ^* such that $\lambda^* = \alpha$, and $K_{\alpha}(c_{\lambda}) = 0$ otherwise. We underline that if there is at least one qubit j such that $\alpha_j \neq \lambda_j$, the expectation value $\mathbb{E}_{\Theta}[\mathcal{L}_{\alpha\lambda}]$ will be zero. In particular, if ρ is a product state, we have $c_{\lambda} \leq 1$ for all c_{λ} and the inequality can be simplified as:

$$\mathbb{E}_{\Theta}[\mathcal{L}_{\alpha}] \leq \left(e^{-\frac{\sigma^2}{2}}\right)^{w(P_{\alpha})}. \quad (\text{C19})$$

b. $L \geq 1$ case

We use recursive steps to derive the expectation value for $L \geq 1$. For clarity, we will use superscript $\mathcal{L}^{(L)}$ to denote the loss function with $U_L(\Theta)$. Taking into account the definition of U_L and $\tilde{W}_{L(K_L)}$ in Eq. (C4) and Eq. (C5), we write explicitly the loss function in terms of U_{L-1} and take out $P_{L,K_L}(\theta_{L,K_L})$ from $\tilde{W}_{L(K_L)}$ in order to make the recursion steps clear:

$$\begin{aligned}\mathcal{L}_{\alpha}^{(L)} &= \text{Tr}(U_L^{\dagger} P_{\alpha} U_L \rho) = \text{Tr}(U_{L-1}^{\dagger} \tilde{W}_{L(K_L)}^{\dagger} P_{\alpha} \tilde{W}_{L(K_L)} U_L \rho) \\ &= \text{Tr}(U_{L-1}^{\dagger} \tilde{W}_{L(K_L-1)}^{\dagger} P_{L,K_L}(-\theta_{L,K_L}) P_{\alpha} P_{L,K_L}(\theta_{L,K_L}) \tilde{W}_{L(K_L-1)} U_{L-1} \rho), \end{aligned} \quad (\text{C20})$$

where $\tilde{W}_{L(K_L-1)} = \left(\prod_{i=1}^{K_L-1} P_{L,i}(\theta_{L,i})\right) C_L$. As P_{α} is a Pauli string, it will either commute or anti-commute with $P_{L,i}$ which is also a Pauli string. In the former, $P_{L,K_L}(-\theta_{L,K_L})$ commutes with P_{α} and Eq. (C20) will be written as :

$$\mathcal{L}_{\alpha}^{(L)} = \tilde{\mathcal{L}}_{\alpha}^{(L)} := \text{Tr}(U_{L-1}^{\dagger} \tilde{W}_{L(K_L-1)}^{\dagger} P_{\alpha} \tilde{W}_{L(K_L-1)} U_{L-1} \rho), \quad (\text{C21})$$

which has the same form as Eq. (C20) but with $\tilde{W}_{L(K_L-1)}$.

On the other hand, if P_{L,K_L} anti-commutes with P_{α} , we have $P_{\alpha} P_{L,K_L}(\theta_{L,K_L}) = P_{L,K_L}(-\theta_{L,K_L}) P_{\alpha}$. This leads to the

following expression :

$$\begin{aligned}\mathcal{L}_\alpha^{(L)} &= \text{Tr}\left(U_{L-1}^\dagger \tilde{W}_{L(K_L-1)}^\dagger P_{L,K_L}(-2\theta_{L,K_L}) P_\alpha \tilde{W}_{L(K_L-1)} U_{L-1}\rho\right) \\ &= \cos(\theta_{L,K_L}) \text{Tr}\left(U_{L-1}^\dagger \tilde{W}_{L(K_L-1)}^\dagger P_\alpha \tilde{W}_{L(K_L-1)} U_{L-1}\rho\right) + i \sin(\theta_{L,K_L}) \text{Tr}\left(U_{L-1}^\dagger \tilde{W}_{L(K_L-1)}^\dagger P_{L,K_L} P_\alpha \tilde{W}_{L(K_L-1)} U_{L-1}\rho\right)\end{aligned}\quad (\text{C22})$$

As both P_α and P_{L,K_L} are Pauli strings, there exists a Pauli string P_τ such that $[P_{L,K_L}, P_\alpha] = 2iP_\tau$. Furthermore, due to the additivity of the trace, we have :

$$\text{Tr}(AP_{L,K_L}P_\alpha B) = -\text{Tr}(AP_\alpha P_{L,K_L} B) = \frac{1}{2} \text{Tr}(A[P_{L,K_L}, P_\alpha]B) = i \text{Tr}(AP_\tau B), \quad (\text{C23})$$

with A and B two matrices. While taking into account the definition of $\tilde{\mathcal{L}}_\alpha^{(L)}$ in Eq. (C21), we can simplify Eq. (C22) as following:

$$\mathcal{L}_\alpha^{(L)} = \cos(\theta_{L,K_L}) \tilde{\mathcal{L}}_\alpha^{(L)} + \sin(\theta_{L,K_L}) \tilde{\mathcal{L}}_\tau^{(L)}. \quad (\text{C24})$$

From its expression, we notice that $\mathcal{L}_\alpha^{(L)}$ can be expressed as a weighted sum of two loss functions $\tilde{\mathcal{L}}_\alpha^{(L)}$ and $\tilde{\mathcal{L}}_\tau^{(L)}$ for the observable P_α and P_τ while excluding P_{L,K_L} from U_L . Hence, the loss function can be extended as a product of cosine and sine functions of θ_L by repeating Eq. (C21) and Eq. (C22) for all $P_{L,i}$ with $i = K_L - 1, \dots, 1$ depending on whether $P_{L,i}$ and P_α commute.

For the following calculations, we define $\mathcal{N}_A^\ell(P_\alpha)$ the set of indices of the Pauli string in ℓ -th layer which anticommutes with P_α , i.e.,

$$\mathcal{N}_A^\ell(P_\alpha) = \{k \mid [P_{\ell,k}, P_\alpha] \neq 0, k \in \{1, \dots, K_\ell\}\}, \quad (\text{C25})$$

and $n_A^\ell(P_\alpha)$ its cardinality. At the end of the recursive steps, we obtain:

$$\begin{aligned}\mathcal{L}_\alpha^{(L)} &= \prod_{i_j \in \mathcal{N}_A^L(P_\alpha)} \cos(\theta_{L,i_j}) \text{Tr}\left(U_{L-1}^\dagger C_L^\dagger P_\alpha C_L U_{L-1}\rho\right) \\ &+ \sum_{i_j \in \mathcal{N}_A^L(P_\alpha)} \sin(\theta_{L,i_j}) \prod_{\substack{i_k \in \mathcal{N}_A^L(P_\alpha) \\ i_k > i_j}} \cos(\theta_{L,i_k}) \prod_{\substack{i_m \in \mathcal{N}_A^L([P_{L,i_j}, P_\alpha]) \\ i_m < i_j}} \cos(\theta_{L,i_m}) \text{Tr}\left(U_{L-1}^\dagger C_L^\dagger \frac{i}{2} [P_{L,i_j}, P_\alpha] C_L U_{L-1}\rho\right) \\ &+ \mathcal{O}(\sin(\theta_{L,i_j}) \sin(\theta_{L,i_k})).\end{aligned}\quad (\text{C26})$$

More explicitly, the first term in Eq. (C26) captures the contributions involving only the cosine terms for the Pauli strings P_{i_j}

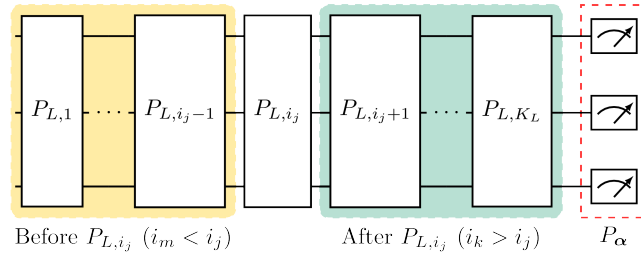


Figure 15. **Schematic diagram of \tilde{W}_L to illustrate the second term in Eq. (C26).** The figure visualizes the Pauli gates that are located *before* P_{i_j} (in yellow) and *after* P_{i_j} (in green). The second term of Eq. (C26) consists of $\cos(\theta_{i_k})$ for $i_k \in \mathcal{N}_A^L(P_\alpha)$ located *after* P_{i_j} and $\cos(\theta_{i_m})$ for $i_m \in \mathcal{N}_A^L(P_\alpha)$ located *before* P_{i_j} .

that anti-commute with P_α . The second term comprises all the sine contributions of P_{i_j} . Specifically, each term in the sum captures the cosine contribution from $P_{i_k}(\theta_{i_k})$ with $i_k > i_j$ that anti-commute with P_α and from $P_{i_m}(\theta_{i_m})$ with $i_m < i_j$ that anti-commute with $[P_{i_j}, P_\alpha]$. Figure 15 is provided to aid in understanding this term. Finally, the last term in Eq. (C26) is dependent on the higher order of $\sin(\theta)$ and contains all the possible combinations of $\cos(\theta_{i_j})$ and $\sin(\theta_{i_j})$, multiplied the nested commutation relations. Since we are interested in small angle initialization, $\sin(\theta_{i_j})$ is much smaller compared to

$\cos(\theta_{i_j})$. Hence, the last term is negligible compared to the first two terms. This natural emergence of nested commutators in the loss function is fundamentally related to the *Dynamical Lie Algebra* (DLA) that determines the expressivity of the circuit.

although the equation above looks overwhelming, taking an expectation over it significantly simplifies the final equation. As C_L is a Clifford gate, there exists a Pauli string $P_\gamma \in \mathbf{P}_n$ such that $C_L^\dagger P_\alpha C_L = P_\gamma$. Thus, the trace in the first term can be rewritten as

$$\mathrm{Tr}\left(U_{L-1}^\dagger C_L^\dagger P_\alpha C_L U_{L-1} \rho\right) = \mathrm{Tr}\left(U_{L-1}^\dagger P_\gamma U_{L-1} \rho\right) := \mathcal{L}_\gamma^{(L-1)}, \quad (\text{C27})$$

where $\mathcal{L}_\gamma^{(L-1)}$ corresponds to the loss function of the observable P_γ with $L - 1$ layers. Similar to the previous justifications, all the expectation values except the first term in Eq. (C26) vanish due to the parity of the sine function. Thus, we have:

$$\begin{aligned} \mathbb{E}_\Theta \left[\mathcal{L}_\alpha^{(L)} \right] &= \mathbb{E}_{\theta_L} \left[\prod_{i_j \in \mathcal{N}_A^L(P_\alpha)} \cos(\theta_{i_j}) \right] \mathbb{E}_{\Theta/\theta_L} \left[\mathcal{L}_\gamma^{(L-1)} \right] \\ &= \prod_{i_j \in \mathcal{N}_A^L(P_\alpha)} \mathbb{E}_{\theta_L} [\cos(\theta_{i_j})] \mathbb{E}_{\Theta/\theta_L} \left[\mathcal{L}_\gamma^{(L-1)} \right] \\ &= e^{-n_A^L(P_\alpha) \frac{\sigma^2}{2}} \mathbb{E}_{\Theta/\theta_L} \left[\mathcal{L}_\gamma^{(L-1)} \right], \end{aligned} \quad (\text{C28})$$

where we use the commutativity between the product and expectation for independent variables in the second line.

Now, let us call $C_{i_1}^{i_n} := C_{i_n} \cdots C_{i_1}$, $C_{i_n}^{i_1 \dagger} := C_{i_1}^\dagger \cdots C_{i_n}^\dagger$ and $C_{i_n}^{i_1 \dagger} P_\alpha C_{i_1}^{i_n}$, the Pauli observable P_α , conjugated between C_i^\dagger and C_i for all $i = i_1, \dots, i_n$. By definition of the Clifford gates, $C_L^{\ell \dagger} P_\alpha C_L^\ell$ is also a Pauli string for all $\ell = 1, \dots, L$ with $C_L^{L+1 \dagger} P_\alpha C_L^{L+1} = P_\alpha$, and hence, there exists a Pauli string P_η such that $P_\eta = C_L^{1 \dagger} P_\alpha C_L^1$. Then, we have the final equation for the expectation value of the loss :

$$\mathbb{E}_\Theta \left[\mathcal{L}_\alpha^{(L)} \right] = \left(e^{-\frac{\sigma^2}{2}} \right)^{N_A^L} \mathbb{E}_\Theta \left[\mathcal{L}_\eta^{(0)} \right] = K_\eta(c_\lambda) \left(e^{-\frac{\sigma^2}{2}} \right)^{N_A^L + w(P_\eta)}, \quad (\text{C29})$$

with $N_A^L = \sum_{\ell=1}^L n_A^\ell \left(C_L^{\ell+1 \dagger} P_\alpha C_L^{\ell+1} \right)$.

In particular, if ρ is a product state and there exists at least one $P_{\ell,i}$ which anti-commutes with $C_L^{\ell+1 \dagger} P_\alpha C_L^{\ell+1}$ for all ℓ , N_A^L will scale linearly with respect to L and the expectation value will decay exponentially with respect to the number of qubits for a polynomial depth circuit :

$$\mathbb{E}_\Theta \left[\mathcal{L}_\alpha^{(L)} \right] \leq \left(e^{-\frac{\sigma^2}{2}} \right)^{\mathrm{poly}(n) + w(P_\eta)}. \quad (\text{C30})$$

This will be the case of EfficientSU2 ansatz which will be presented in Appendix C 3.

2. Variance with normal initialization

a. $L = 0$ case

In this section, we calculate the lower bound for the variance of the loss function, $\mathrm{Var}_\Theta[\mathcal{L}]$. For simplicity, we are interested in the scenario where the covariance between the loss functions for two different Pauli strings, \mathcal{L}_α and \mathcal{L}_β vanishes, i.e. $\mathrm{Var}_\Theta[\mathcal{L}_\alpha \mathcal{L}_\beta] = 0$ for $\alpha \neq \beta$. Then, $\mathrm{Var}_\Theta[\mathcal{L}]$ can be written as a sum :

$$\mathrm{Var}_\Theta[\mathcal{L}] = \sum_\alpha a_\alpha \mathrm{Var}_\Theta[\mathcal{L}_\alpha] = \sum_\alpha a_\alpha \left(\mathbb{E}_\Theta[\mathcal{L}_\alpha^2] - \mathbb{E}_\Theta[\mathcal{L}_\alpha]^2 \right). \quad (\text{C31})$$

with \mathcal{L}_α^2 that can be decomposed explicitly as :

$$\mathcal{L}_\alpha^2 = \sum_\lambda \sum_{\lambda'} c_\lambda c_{\lambda'} \prod_i \mathcal{L}_{\alpha\lambda}^i \prod_j \mathcal{L}_{\alpha\lambda'}^j = \sum_\lambda \sum_{\lambda'} c_\lambda c_{\lambda'} \prod_i \mathcal{L}_{\alpha\lambda}^i \mathcal{L}_{\alpha\lambda'}^i. \quad (\text{C32})$$

Now, we will explicitly compute $\mathbb{E}_\Theta[\mathcal{L}_{\alpha\lambda}^i \mathcal{L}_{\alpha\lambda'}^i]$ by distinguishing different cases of $\alpha_j \neq 0$. To begin with, let us consider

the case where $\alpha_j = \nu_j$. With Eq. (C11), we have :

$$\mathcal{L}_{\alpha\lambda}^j \mathcal{L}_{\alpha\lambda'}^j = \cos^2(\omega_j) \delta_{\alpha_j \lambda_j} \delta_{\lambda_j \lambda'_j} + \sin^2(\omega_j) \bar{\delta}_{\alpha_j \lambda_j} \bar{\delta}_{\lambda_j \mu_j} \delta_{\lambda_j \lambda'_j} + \text{odd} , \quad (\text{C33})$$

where the last term indicates the term which is proportional to $\cos(\omega_j) \sin(\omega_j)$, thus odd at $\omega_j = 0$ over \mathbb{R} . This term cancels out as it is an odd function integrated with an even probability distribution. By integrating the first two terms over normal probability distribution for ω_j , we obtain :

$$\mathbb{E}_{\omega_j} [\cos^2(\omega_j)] = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\infty} \cos^2(\omega_j) e^{-\frac{\omega_j^2}{2\sigma^2}} d\omega_j = e^{-\sigma^2} \cosh \sigma^2 = \frac{1 + e^{-2\sigma^2}}{2} , \quad (\text{C34})$$

$$\mathbb{E}_{\omega_j} [\sin^2(\omega_j)] = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\infty} \sin^2(\omega_j) e^{-\frac{\omega_j^2}{2\sigma^2}} d\omega_j = e^{-\sigma^2} \sinh \sigma^2 = \frac{1 - e^{-2\sigma^2}}{2} . \quad (\text{C35})$$

Therefore, we can write the expectation value of Eq. (C33) as :

$$\mathbb{E}_{\omega_j} [\mathcal{L}_{\alpha\lambda}^j \mathcal{L}_{\alpha\lambda'}^j] = \frac{1 + e^{-2\sigma^2}}{2} \delta_{\alpha_j \lambda_j} \delta_{\lambda_j \lambda'_j} + \frac{1 - e^{-2\sigma^2}}{2} \bar{\delta}_{\alpha_j \lambda_j} \bar{\delta}_{\lambda_j \mu_j} \delta_{\lambda_j \lambda'_j} . \quad (\text{C36})$$

On the other hand, if $\alpha_j \neq \nu_j$, there are two different cases : 1) $\alpha_j = \mu_j$ and 2) $\alpha_j \neq \mu_j$. We start with the first case of $\alpha_j = \mu_j$. Using Eq. (C14), we can expand the expression of $\mathcal{L}_{\alpha\lambda}^j \mathcal{L}_{\alpha\lambda'}^j$ as:

$$\mathcal{L}_{\alpha\lambda}^j \mathcal{L}_{\alpha\lambda'}^j = \cos^2(\phi_j) \delta_{\alpha_j \lambda_j} \delta_{\lambda_j \lambda'_j} + \sin^2(\omega_j) \sin^2(\phi_j) \bar{\delta}_{\alpha_j \lambda_j} \delta_{\lambda_j \nu_j} \delta_{\lambda_j \lambda'_j} + \cos^2(\omega_j) \sin^2(\phi_j) \bar{\delta}_{\alpha_j \lambda_j} \bar{\delta}_{\lambda_j \nu_j} \delta_{\lambda_j \lambda'_j} + \text{odd} , \quad (\text{C37})$$

which gives the following expectation value $\mathbb{E}_{\omega_j, \phi_j} [\mathcal{L}_{\alpha\lambda}^j \mathcal{L}_{\alpha\lambda'}^j]$ using Eq. (C34) and Eq. (C35):

$$\mathbb{E}_{\omega_j, \phi_j} [\mathcal{L}_{\alpha\lambda}^j \mathcal{L}_{\alpha\lambda'}^j] = \frac{1 + e^{-2\sigma^2}}{2} \delta_{\alpha_j \lambda_j} \delta_{\lambda_j \lambda'_j} + \left(\frac{1 - e^{-2\sigma^2}}{2} \right)^2 \bar{\delta}_{\alpha_j \lambda_j} \delta_{\lambda_j \nu_j} \delta_{\lambda_j \lambda'_j} + \frac{1 - e^{-4\sigma^2}}{4} \bar{\delta}_{\alpha_j \lambda_j} \bar{\delta}_{\lambda_j \nu_j} \delta_{\lambda_j \lambda'_j} . \quad (\text{C38})$$

Now, let's consider the case with $\alpha_j \neq \mu_j$. With the similar justification as before, $\mathcal{L}_{\alpha\lambda}^j \mathcal{L}_{\alpha\lambda'}^j$ can be written as :

$$\mathcal{L}_{\alpha\lambda}^j \mathcal{L}_{\alpha\lambda'}^j = \cos^2(\omega_j) \cos^2(\phi_j) \delta_{\alpha_j \lambda_j} \delta_{\lambda_j \lambda'_j} + \sin^2(\phi_j) \bar{\delta}_{\alpha_j \lambda_j} \delta_{\lambda_j \mu_j} \delta_{\lambda_j \lambda'_j} + \cos^2(\phi_j) \sin^2(\omega_j) \bar{\delta}_{\alpha_j \lambda_j} \delta_{\lambda_j \nu_j} \delta_{\lambda_j \lambda'_j} + \text{odd} . \quad (\text{C39})$$

where the last odd term is proportional to $\cos(\omega_j) \sin(\omega_j)$ and $\cos(\phi_j) \sin(\phi_j)$. This leads to :

$$\mathbb{E}_{\omega_j, \phi_j} [\mathcal{L}_{\alpha\lambda}^j \mathcal{L}_{\alpha\lambda'}^j] = \left(\frac{1 + e^{-2\sigma^2}}{2} \right)^2 \delta_{\alpha_j \lambda_j} \delta_{\lambda_j \lambda'_j} + \frac{1 - e^{-2\sigma^2}}{2} \bar{\delta}_{\alpha_j \lambda_j} \delta_{\lambda_j \mu_j} \delta_{\lambda_j \lambda'_j} + \frac{1 - e^{-4\sigma^2}}{4} \bar{\delta}_{\alpha_j \lambda_j} \delta_{\lambda_j \nu_j} \delta_{\lambda_j \lambda'_j} . \quad (\text{C40})$$

We can notice that in all cases, $\mathbb{E}_{\omega_j, \phi_j} [\mathcal{L}_{\alpha\lambda}^j \mathcal{L}_{\alpha\lambda'}^j]$ cancels out if $\lambda_j \neq \lambda'_j$. Thus, the final expression will be summarized as a sum over λ . Combining the expressions above, we have the expression for the expectation value $\mathbb{E}[\mathcal{L}_{\alpha}^2]$:

$$\mathbb{E}_{\Theta} [\mathcal{L}_{\alpha}^2] = \sum_{\lambda} c_{\lambda}^2 \prod_j \mathbb{E}_{\Theta} \left[\left(\mathcal{L}_{\alpha\lambda}^j \right)^2 \right] = \sum_{\lambda} c_{\lambda}^2 \prod_{j \in \mathcal{J}_{\alpha}} \tilde{c}_{\alpha_j \mu_j \nu_j \lambda_j} \left(\frac{1 + e^{-2\sigma^2}}{2} \delta_{\alpha_j \lambda_j} + \frac{1 - e^{-2\sigma^2}}{2} \bar{\delta}_{\alpha_j \lambda_j} \right) , \quad (\text{C41})$$

where $\tilde{c}_{\alpha_j \mu_j \nu_j \lambda_j}$ is the coefficient indicating different possibilities depending on the value of α_j, μ_j, ν_j and λ_j as follows:

$$\begin{aligned} \tilde{c}_{\alpha_j \mu_j \nu_j \lambda_j} = & \left(\delta_{\alpha_j \nu_j} + \bar{\delta}_{\alpha_j \nu_j} \delta_{\alpha_j \mu_j} + \bar{\delta}_{\alpha_j \nu_j} \bar{\delta}_{\alpha_j \mu_j} \right) \delta_{\alpha_j \lambda_j} + \frac{1 + e^{-2\sigma^2}}{2} \bar{\delta}_{\alpha_j \lambda_j} \delta_{\alpha_j \nu_j} \bar{\delta}_{\lambda_j \mu_j} \\ & + \bar{\delta}_{\alpha_j \lambda_j} \bar{\delta}_{\alpha_j \nu_j} \left(\bar{\delta}_{\alpha_j \mu_j} \delta_{\lambda_j \mu_j} + \frac{1 - e^{-2\sigma^2}}{2} \delta_{\alpha_j \mu_j} \delta_{\lambda_j \nu_j} + \frac{1 + e^{-2\sigma^2}}{2} (\delta_{\alpha_j \mu_j} \bar{\delta}_{\lambda_j \nu_j} + \bar{\delta}_{\alpha_j \mu_j} \delta_{\lambda_j \nu_j}) \right) . \end{aligned} \quad (\text{C42})$$

Eq. (C41) implies that $\mathbb{E}_{\Theta} [\mathcal{L}_{\alpha}^2]$ scales differently depending on whether the Pauli matrix in P_{α} is the same as the one in the

initial state ρ at each qubit j and this will highly influence the lower bound of the variance calculated in the following steps.

b. $L \geq 1$ case

To calculate the lower bound of $\mathbb{E}_{\Theta} [(\mathcal{L}_{\alpha}^{(L)})^2]$ for $L > 0$, we can follow a recursive approach similar to the previous steps. We begin by squaring Eq. (C26) to compute $(\mathcal{L}_{\alpha}^{(L)})^2$:

$$\begin{aligned} (\mathcal{L}_{\alpha}^{(L)})^2 &= \prod_{i_j \in \mathcal{N}_A^L(P_{\alpha})} \cos(\theta_{L,i_j})^2 (\mathcal{L}_{\gamma}^{(L-1)})^2 \\ &+ \sum_{i_j \in \mathcal{N}_A^L(P_{\alpha})} \sin^2(\theta_{i_j}) \prod_{\substack{i_k \in \mathcal{N}_A^L(P_{\alpha}) \\ i_k > i_j}} \cos^2(\theta_{L,i_k}) \prod_{\substack{i_m \in \mathcal{N}_A^L([P_{L,i_j}, P_{\alpha}]) \\ i_m < i_j}} \cos^2(\theta_{L,i_m}) (\mathcal{L}_{\delta_{i_j}}^{(L-1)})^2 \\ &+ \mathcal{O}(\sin^4(\theta)) + \text{odd} \end{aligned} \quad (\text{C43})$$

where $P_{\delta_{i_j}} = \frac{i}{2} C^{\dagger} [P_{L,i_j}, P_{\alpha}] C \in \mathcal{G}$ and the odd term containing all the terms depending on $\cos(\theta_{L,i}) \sin(\theta_{L,i})$ and $\sin(\theta_{L,i}) \sin(\theta_{L,k})$ for $i \neq k$. Taking the expectation value over Eq. (C43) and cancelling out all the odd terms leads to:

$$\begin{aligned} \mathbb{E}_{\Theta} [(\mathcal{L}_{\alpha}^{(L)})^2] &= \left(\frac{1 + e^{-2\sigma^2}}{2} \right)^{n_A^L(P_{\alpha})} \mathbb{E}_{\Theta/\theta_L} \left[(\mathcal{L}_{\gamma}^{(L-1)})^2 \right] \\ &+ \sum_{i_j \in \mathcal{N}_A^L(P_{\alpha})} \left(\frac{1 - e^{-2\sigma^2}}{2} \right) \left(\frac{1 + e^{-2\sigma^2}}{2} \right)^{\tilde{n}_A^L(i_j)} \mathbb{E}_{\Theta/\theta_L} \left[(\mathcal{L}_{\delta_{i_j}}^{(L-1)})^2 \right] \\ &+ \mathcal{O} \left(\left(\frac{1 - e^{-2\sigma^2}}{2} \right)^2 \right), \end{aligned} \quad (\text{C44})$$

where $\tilde{n}_A^L(i_j)$ is the number of cosine terms multiplied to $\sin(\theta_{i_j})$ in Eq. (C43). Rigorously, it can be mathematically written as:

$$\tilde{n}_A^L(i_j) = |\{i_k \in \mathcal{N}_A^L(P_{\alpha}) \mid i_k > i_j\}| + |\{i_m \in \mathcal{N}_A^L([P_{L,i_j}, P_{\alpha}]) \mid i_m < i_j\}| \quad (\text{C45})$$

Indeed, it counts all the Pauli gates in \tilde{W}_{ℓ} that anti-commute with P_{α} if they are located after P_{L,i_j} , and those that anti-commute with $[P_{L,i_j}, P_{\alpha}]$ if they are located before P_{L,i_j} (see Figure 15).

As $\mathbb{E}_{\Theta} [(\mathcal{L}_{\alpha}^{(L)})^2]$ depends on P_{α} and all possible nested commutators including $P_{L,i}$, it is extremely complicated to generalize the exact equation. Therefore, for the general case, we will compute an exact lower bound depending on σ . However, this bound does not provide much information about the scaling with respect to the system's size without further assumption on the circuit architecture. In Appendix C3, we consider an EfficientSU2 architecture as a specific example and provide some *approximate* bounds which show the polynomial scaling of the variance for the leading order in σ . These bounds, despite being approximate, are rather tight as supported by our numerics.

First of all, let us consider the case with $\sigma \ll 1$. As $e^{-2\sigma^2}$ is close to 1, $\frac{1+e^{-2\sigma^2}}{2}$ dominates over $\frac{1-e^{-2\sigma^2}}{2}$, and therefore, the first term will mainly contribute in the lower bound:

$$\mathbb{E}_{\Theta} [(\mathcal{L}_{\alpha}^{(L)})^2] > \left(\frac{1 + e^{-2\sigma^2}}{2} \right)^{n_A^L(P_{\alpha})} \mathbb{E}_{\Theta} [(\mathcal{L}_{\gamma}^{(L-1)})^2] > \left(\frac{1 + e^{-2\sigma^2}}{2} \right)^{N_A^L} \mathbb{E}_{\Theta} [(\mathcal{L}_{\eta}^{(0)})^2], \quad (\text{C46})$$

where we obtain the last inequality by induction. Combining Eq. (C29) and Eq. (C46), we have the lower bound for the variance $\text{Var}_{\Theta} [\mathcal{L}_{\alpha}^{(L)}]$:

$$\text{Var}[\mathcal{L}_{\alpha}] = \mathbb{E}[(\mathcal{L}_{\alpha})^2] - \mathbb{E}[\mathcal{L}_{\alpha}]^2 \geq \left(\frac{1 + e^{-2\sigma^2}}{2} \right)^{N_A^L} \mathbb{E}_{\Theta} [(\mathcal{L}_{\eta}^{(0)})^2] - (e^{-\sigma^2})^{N_A^L} \mathbb{E}_{\Theta} [(\mathcal{L}_{\eta}^{(0)})]^2. \quad (\text{C47})$$

In particular, if there exist a λ^* such that $\eta_j = \lambda_j$ for all $\eta_j \neq 0$, we have :

$$\text{Var}[\mathcal{L}_\alpha] = \mathbb{E}[(\mathcal{L}_\alpha)^2] - \mathbb{E}[\mathcal{L}_\alpha]^2 > c_{\lambda^*}^2 \left(\frac{1 + e^{-2\sigma^2}}{2} \right)^{N_A^L + w(P_\eta)} - c_{\lambda^*} (e^{-\sigma^2})^{N_A^L + w(P_\eta)}, \quad (\text{C48})$$

otherwise, $\mathbb{E}_\Theta[\mathcal{L}_\eta^{(L)}] = 0$, and thus,

$$\text{Var}[\mathcal{L}_\alpha] = \mathbb{E}[(\mathcal{L}_\alpha)^2] > \sum_{\lambda} c_{\lambda}^2 \left(\frac{1 + e^{-2\sigma^2}}{2} \right)^{N_A^L + \sum_{j \in \mathcal{J}_\eta} \delta_{\eta_j \lambda_j}} \left(\frac{1 - e^{-2\sigma^2}}{2} \right)^{\sum_{j \in \mathcal{J}_\eta} \bar{\delta}_{\eta_j \lambda_j}}. \quad (\text{C49})$$

On the other hand, if $\sigma \gg 0$, the identity initialization is not valid anymore, and a $\text{poly}(n)$ depth unstructured circuit is sufficiently expressive to induce exponential concentration of the loss function.

3. Case study: Absence of barren plateau in EfficientSU2 ansatz

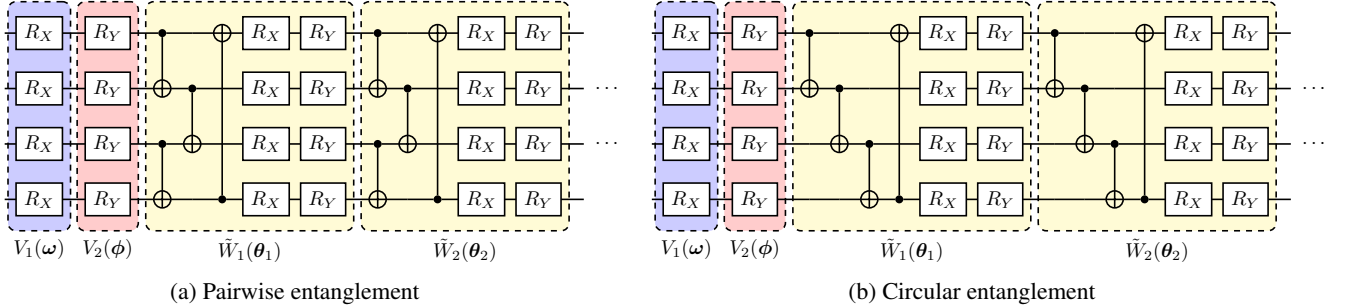


Figure 16. **Circuit architecture of the EfficientSU2 ansatz [101].** The circuit is initialized with two orthogonal layers of single-qubit Pauli rotation gates and L layers of entanglement layers with CNOT gates and single-qubit rotation gates. The entanglement gates can have either (a) a pairwise structure, or (b) a circular structure.

In this section, we will analyze the decay of the gradient variance in the EfficientSU2 ansatz shown on Figure 16, which consists only of single-qubit Pauli rotations and CNOT gates. The single-qubit rotations can be chosen in different combinations, and the entanglement map can be either *pairwise* or *circular*. Although we only use the EfficientSU2 ansatz for the study, the result can also be generalized to other types of quantum circuits. Our results show that with small σ the variance of the loss scales polynomially with the number of qubits. We note that while we rely on some approximations to obtain these *approximate* lower bounds of the variance (i.e., the bounds are written in the leading order of σ), these bounds are rather tight, as supported by our numerics.

We will analyze how the loss function behaves depending on the type of observables and the entanglement map for both local and global observables. Let us start with two different local observables $P_\alpha = \sigma_{\alpha_0} \otimes \mathbb{I}^{\otimes n-1}$ and $P_\beta = \sigma_{\beta_0} \otimes \mathbb{I}^{\otimes n-1}$ such that $\alpha_0 \neq \beta_0$. Furthermore, we assume that ρ only consists of P_λ such that $\lambda_j \in \{0, \alpha_0\}$ for all λ . This is the case if we have zero initial state $\rho = |0\rangle\langle 0|^{\otimes n} = (\mathbb{I} + Z)^{\otimes n} / 2^n$ with $\sigma_{\alpha_0} = Z$ and $\sigma_{\beta_0} = X$.

Table V summarizes \mathcal{L}_α and \mathcal{L}_β for all possible combinations of μ_0 and ν_0 in $L = 0$ case. As the table shows, the covariance between \mathcal{L}_α and \mathcal{L}_β vanishes, as assumed in Appendix C2 and thus, Eq. (C31) holds.

It is more straightforward to show the lower bound of $\text{Var}_\Theta[\mathcal{L}_\alpha]$, as the $\mathbb{E}_\Theta[\mathcal{L}_\alpha^{(0)}]$ scales following the dominant term, $(1 + e^{-2\sigma^2})/2$. As EfficientSU2 ansatz only consists of CNOT gates among the Clifford gates, the Pauli strings $C_L^{\ell\dagger} P_\alpha C_L^\ell$, in particular, $C_L^{1\dagger} P_\alpha C_L^1 = P_\eta$, only consists of σ_{α_0} . Furthermore, the ansatz only contains single-qubit Pauli rotation gates, hence, the number of anti-commuting Pauli operators, $n_A^\ell \left(C_L^{\ell+1\dagger} P_\alpha C_L^\ell \right)$ will be proportional to the weight $w \left(C_L^{\ell+1\dagger} P_\alpha C_L^\ell \right)$ of the Pauli string at each layer ℓ and, as a result, the total number of anti-commuting gates will be proportional to the sum of weights, $N_A^L \approx \sum_{\ell=1}^L w \left(C_L^{\ell+1\dagger} P_\alpha C_L^\ell \right)$.

As N_A^L depends on the position of the observable and the types of the entanglement map, it is complex to find a general formula. Therefore, we will assume that the non-trivial Pauli matrices can span over n -qubits for $C_L^{\ell\dagger} P_\alpha C_L^\ell$, $\ell = 1, \dots, n$, i.e. for all $d \in \{1, \dots, n\}$, there exists an $\ell \in \{1, \dots, n\}$ such that $w(C_L^{\ell\dagger} P_\alpha C_L^\ell) = d$. Under this assumption, we take the average

	\mathcal{L}_{α_0}	\mathcal{L}_{β_0}	$\mathbb{E}[\mathcal{L}_{\alpha_0}]$	$\mathbb{E}[\mathcal{L}_{\beta_0}]$	$\mathbb{E}[\mathcal{L}_{\alpha_0}^2]$	$\mathbb{E}[\mathcal{L}_{\beta_0}^2]$	$\mathbb{E}[\mathcal{L}_{\alpha_0}\mathcal{L}_{\beta_0}]$
$\alpha_0 = \nu_0, \beta_0 = \mu_0$	$\cos(\omega_j)$	$\sin(\omega_j) \sin(\phi_j)$	$e^{-\frac{\sigma^2}{2}}$	0	$\frac{1+e^{-2\sigma^2}}{2}$	$\left(\frac{1-e^{-2\sigma^2}}{2}\right)^2$	0
$\alpha_0 = \nu_0, \beta_0 \neq \mu_0$	$\cos(\omega_j)$	$\sin(\omega_j) \cos(\phi_j)$	$e^{-\frac{\sigma^2}{2}}$	0	$\frac{1+e^{-2\sigma^2}}{2}$	$\frac{1-e^{-4\sigma^2}}{4}$	0
$\alpha_0 \neq \nu_0, \beta_0 = \mu_0$	$\cos(\omega_j) \cos(\phi_j)$	$\cos(\omega_j) \sin(\phi_j)$	$e^{-\sigma^2}$	0	$\left(\frac{1+e^{-2\sigma^2}}{2}\right)^2$	$\frac{1-e^{-4\sigma^2}}{4}$	0
$\alpha_0 = \mu_0, \beta_0 = \nu_0$	$\cos(\phi_j)$	0	$e^{-\frac{\sigma^2}{2}}$	0	$\frac{1+e^{-2\sigma^2}}{2}$	$\frac{1-e^{-2\sigma^2}}{2}$	0
$\alpha_0 \neq \mu_0, \beta_0 = \nu_0$	$\cos(\phi_j) \cos(\omega_j)$	$\sin(\omega_j)$	$e^{-\sigma^2}$	0	$\left(\frac{1+e^{-2\sigma^2}}{2}\right)^2$	$\frac{1-e^{-2\sigma^2}}{2}$	0
$\alpha_0 = \mu_0, \beta_0 \neq \nu_0$	$\cos(\phi_j)$	$\sin(\phi_j)$	$e^{-\frac{\sigma^2}{2}}$	0	$\frac{1+e^{-2\sigma^2}}{2}$	$\frac{1-e^{-4\sigma^2}}{4}$	0

Table V. **Summary of the loss functions \mathcal{L}_α and \mathcal{L}_β for different circuit architectures.** The covariance $\mathbb{E}_\Theta[\mathcal{L}_{\alpha_0}\mathcal{L}_{\beta_0}]$ vanish for all cases as $\mathcal{L}_{\alpha_0}\mathcal{L}_{\beta_0}$ is always odd with respect to ω_j or ϕ_j over \mathbb{R} .

over $w(C_L^{\ell\dagger} P_\alpha C_\ell^L)$ to have $w(P_\eta) \approx n/2$ and $N_A^L \approx (n^2 + n)/2$. In addition, we also consider the extreme-case scenario for the polynomial depth circuit where the $N_A^L \approx Ln = n^2$, and $w(P_\eta) = n$. For $\sigma \ll 1$, Eq. (C48) and Table V lead to :

$$\text{Var}[\mathcal{L}_\alpha] \geq \min \left[c_{\lambda^*}^2 \left(\frac{1+e^{-2\sigma^2}}{2} \right)^{\frac{n^2}{2}+n} - c_{\lambda^*} \left(e^{-\sigma^2} \right)^{\frac{n^2}{2}+n}, c_{\lambda^*}^2 \left(\frac{1+e^{-2\sigma^2}}{2} \right)^{n^2+n} - c_{\lambda^*} \left(e^{-\sigma^2} \right)^{n^2+n} \right], \quad (\text{C50})$$

with λ^* such that $\lambda^* = \eta$. In particular, in the case of the zero state, the equation simplifies with $c_{\lambda^*} = 1$. It is important to consider both cases, because the interplay between $\mathbb{E}_\Theta[\mathcal{L}_\alpha]^2$ and $\mathbb{E}_\Theta[(\mathcal{L}_\alpha)^2]$ depends greatly on the value of σ and n . Taking into account only the extreme case will make the calculation deviate too much from the real lower bound.

We also compute the scaling of the variance while assuming the initial zero state. Taking a Taylor expansion with respect to σ around 0 gives us:

$$\text{Var}[\mathcal{L}_\alpha] > (1 - \sigma^2 + \sigma^4)^{n^2+n} - \left(1 - \sigma^2 + \frac{\sigma^4}{2}\right)^{n^2+n} \quad (\text{C51})$$

$$> \frac{1}{2}n(n+1)\sigma^4 - \frac{1}{2}(n^2(n+1))^2\sigma^6 \quad (\text{C52})$$

$$\approx \frac{1}{2}n^2\sigma^4(1 - n^2\sigma^2) \quad (\text{C53})$$

$$> \frac{1}{n^b}. \quad (\text{C54})$$

with a $b > 1$ independent of n , which implies that $\text{Var}_\Theta[\mathcal{L}_\alpha]$ decays polynomially. Rearranging Eq. (C54), we can conclude that $\text{Var}_\Theta[\mathcal{L}_\alpha]$ will scale as $\mathcal{O}(1/n^b)$ with $b > 2$ if :

$$\sigma \in \Theta\left(\frac{1}{n}\right). \quad (\text{C55})$$

Note that the $\max\left(w\left(C_L^{\ell\dagger} P_\alpha C_\ell^L\right)\right) = n$ and $\max(N_A^L) = n^2$, and thus, the lower bound given by Eq. (C50) also applies to the global observable case with $P_\alpha = \sigma_{\alpha_0}^{\otimes n}$ as shown on Figure 18. On Figure 17 and 18, we display $\text{Var}[\mathcal{L}_\alpha^{(L)}]$ for $L = n$ and its lower bound calculated with Eq. (C50) with respect to the number of qubits. Furthermore, for $\sigma \geq 1$, we take the lower bound as $\text{Var}_\Theta[\mathcal{L}_\alpha] = 2^{-n}$ as justified in Appendix C2. The figures clearly show that the variance follows the computed lower bound for both local and global observables in the case of $\sigma \ll 1$ while it decays exponentially for large σ .

On the other hand, understanding the variance of $\mathbb{E}_\Theta[\mathcal{L}_\beta]$ requires additional insight due to the presence of sine terms. As shown on Table V, $\mathbb{E}_\Theta[\mathcal{L}_\beta]$ vanishes regardless of the ansatz, thus it suffices to find the scaling of $\mathbb{E}_\Theta[\mathcal{L}_\beta^2]$. We start by rewriting

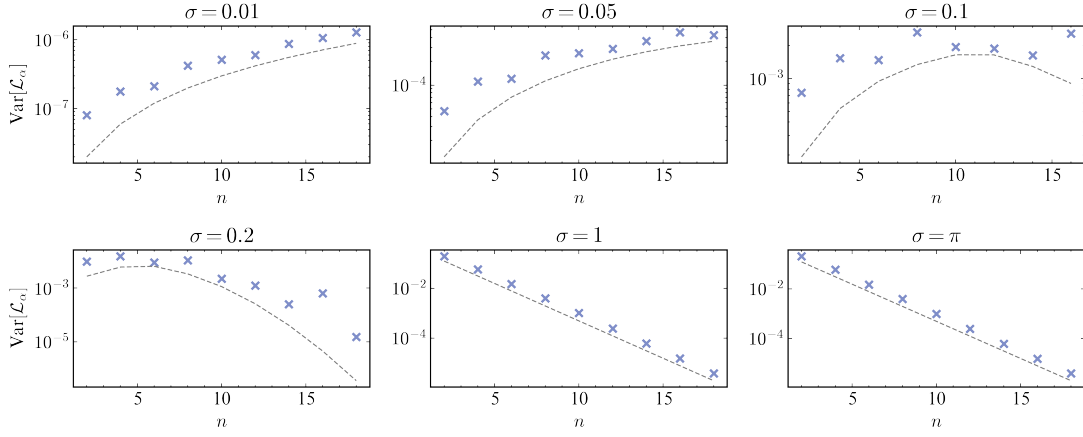


Figure 17. **Variance of \mathcal{L}_α for the local Z observable and its lower bound in polynomial depth circuit with different initialization ranges.** The blue cross represents the simulated variance obtained in EfficientSU2 ansatz with $\text{poly}(n)$ depth and the gray dashed line its lower bound calculated with Eq. (C50).

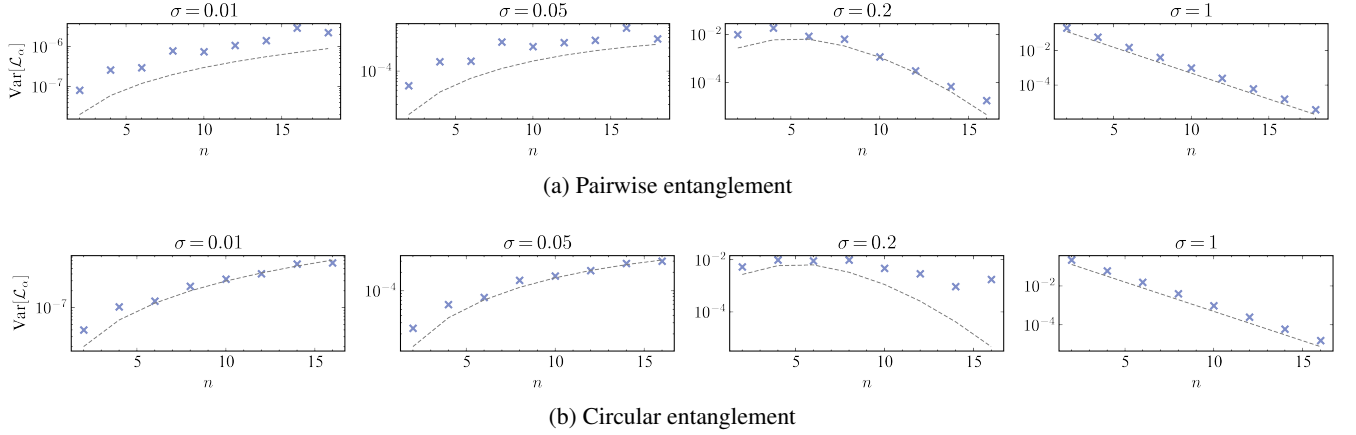


Figure 18. **Variance of \mathcal{L}_α for the global Z observable and its lower bound in the polynomial depth circuit with different initialization ranges.** The blue cross represents the simulated variance obtained in the EfficientSU2 ansatz with $\text{poly}(n)$ depth and the gray dashed line its lower bound calculated with Eq. (C50). We observe that $\text{Var}_\Theta[\mathcal{L}_\alpha]$ for both (a) pairwise and (b) circular entanglement also follows the theoretical lower bound, and does not exhibit a barren plateau with an appropriate choice of initialization.

Eq. (C44) for the EfficientSU2 ansatz with $K_\ell = 2$ (c.f. Figure 16) as follows :

$$\begin{aligned} \mathbb{E}_\Theta \left[\left(\mathcal{L}_\beta^{(L)} \right)^2 \right] &= \left(\frac{1 + e^{-2\sigma^2}}{2} \right)^2 \mathbb{E}_\Theta \left[\left(\mathcal{L}_{\gamma'}^{(L-1)} \right)^2 \right] + \left(\frac{1 + e^{-2\sigma^2}}{2} \right) \left(\frac{1 - e^{-2\sigma^2}}{2} \right) \mathbb{E}_\Theta \left[\left(\mathcal{L}_{\delta'_1}^{(L-1)} \right)^2 \right] \\ &\quad + \left(\frac{1 + e^{-2\sigma^2}}{2} \right) \left(\frac{1 - e^{-2\sigma^2}}{2} \right) \mathbb{E}_\Theta \left[\left(\mathcal{L}_{\delta'_2}^{(L-1)} \right)^2 \right] + \mathcal{O} \left(\frac{1 - e^{-2\sigma^2}}{2} \right)^2 \end{aligned} \quad (\text{C56})$$

where we denote $P_{\gamma'} = C^\dagger P_\beta C$, $P_{\delta'_1} = -iC^{\dagger \frac{1}{2}} [P_{L,1}, P_\beta] C$ and $P_{\delta'_2} = -iC^{\dagger \frac{1}{2}} [P_{L,2}, P_\beta] C$.

Without losing generality, we assume that $[P_{L,i_j}, P_\beta] = 2i\sigma_{\alpha_0} \otimes \mathbb{I}^{n-1}$ for $i_j \in \{1, 2\}$. This corresponds to the case of an X observable with RZ or RY rotations in \tilde{W}_ℓ . This leads $\mathbb{E}_\Theta \left[\left(\mathcal{L}_{\delta'_j}^{(L-1)} \right)^2 \right]$ back to the previous case for \mathcal{L}_α , scaling as $((1 + e^{-2\sigma^2})/2)^{n^2+n}$. On the other hand, since $P_{\gamma'}$ only contains σ_{β_0} , in the worst case scenario, we have $w(C_L^{\dagger \ell} P_\beta C_L^L) = n$, leading to the the first term scaling as $((1 - e^{-2\sigma^2})/2)^n$. Therefore, the second term will be the dominant term in case of the

small angle initialization and we can conclude that :

$$\text{Var}_{\Theta}[\mathcal{L}_{\beta}^2] > c_{\lambda^{**}} \left(\frac{1 + e^{-2\sigma^2}}{2} \right)^{n^2+n} \left(\frac{1 - e^{-2\sigma^2}}{2} \right), \quad (\text{C57})$$

with $c_{\lambda^{**}}$ such that $P_{\lambda^{**}} = C_L^{\dagger}[P_{L,i_j}, P_{\beta}]C_L^L$. The equation is confirmed with Figure 19 showing $\text{Var}[\mathcal{L}_{\beta}^{(L)}]$ for $L = n$ and its lower bound calculated with Eq. (C57).

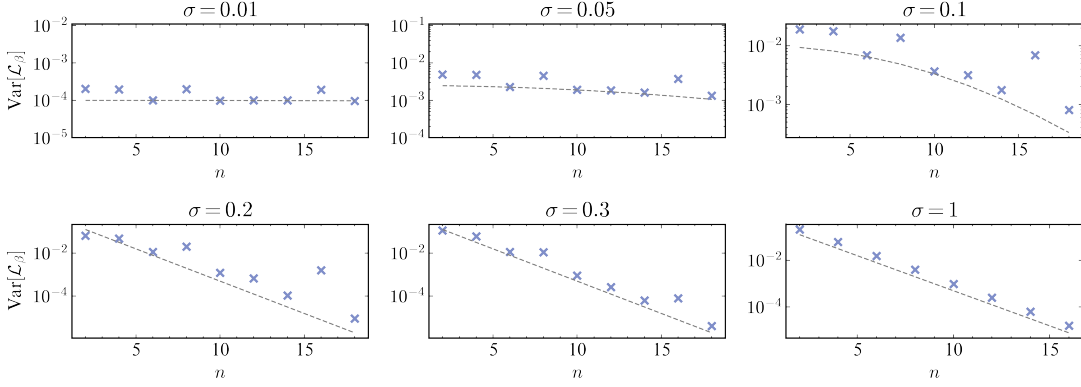


Figure 19. **Variance of \mathcal{L}_{β} for the local X observable and its lower bound in polynomial depth circuit with different initialization ranges.** The blue cross represents the simulated variance obtained in EfficientSU2 ansatz with $\text{poly}(n)$ depth and the pairwise entanglement map. The gray dashed line corresponds to the lower bound calculated with Eq. (C57).

For the zero initial state, by taking the Taylor expansion with respect to σ around 0, Eq. (C57) can be approximated as :

$$\text{Var}_{\Theta}[\mathcal{L}_{\beta}^2] > (1 - \sigma^2)^{n^2+n} \sigma^2 > \sigma^2 - (n^2 + n)\sigma^4 > \sigma^2 - 2n^2\sigma^4 > \frac{1}{n^b}, \quad (\text{C58})$$

resulting in the same conclusion as before, that $\text{Var}_{\Theta}[\mathcal{L}_{\beta}^2]$ decays as $\mathcal{O}(1/n^2)$ if $\sigma \in \Theta(1/n)$.

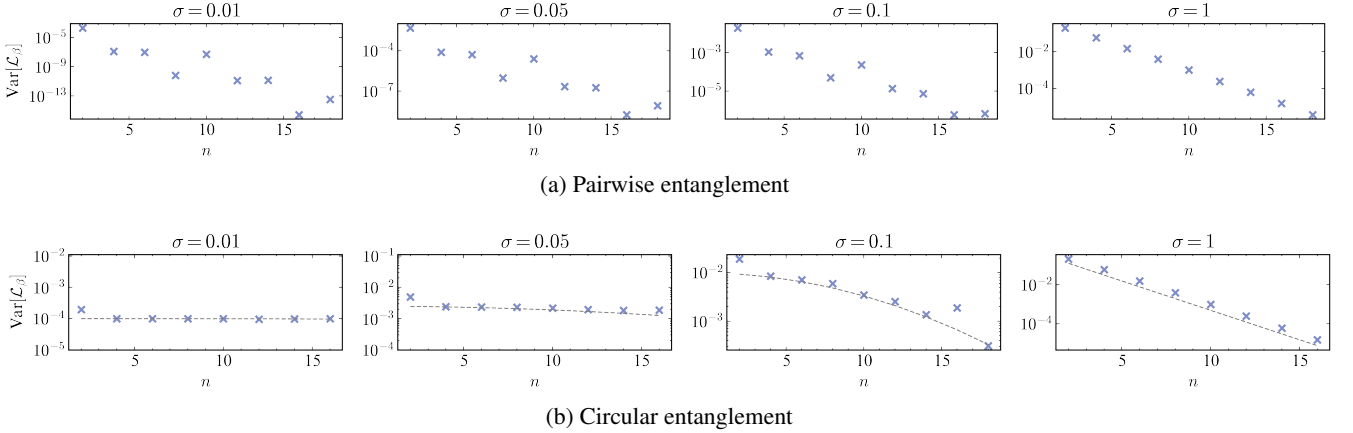


Figure 20. **Variance of \mathcal{L}_{β} for the global X observable and its lower bound in the polynomial depth circuit with different initialization ranges.** The blue cross represents the simulated variance obtained in the EfficientSU2 ansatz with $\text{poly}(n)$ depth and the gray dashed line its lower bound calculated with Eq. (C57). We observe that $\text{Var}_{\Theta}[\mathcal{L}_{\beta}]$ decays exponentially for the (a) pairwise entanglement independent of σ , while it follows the theoretical lower bound given for (b) circular entanglement. This discrepancy comes from the fact that $C_L^{\ell\dagger} P_{\beta} C_L^{\ell}$ behaves differently depending on the entanglement map.

Unlike the Z observable, which exhibits similar behavior in both the global and local observables, the behavior of the global X observable varies depending on the type of entanglement. In Figure 20, we observe that the variance decays exponentially regardless of σ with the pairwise entanglement map. However, with the circular entanglement map, it follows the lower bound given by Eq. (C57), which was computed for the local observable.

Although it is complicated to justify the result mathematically, we can explain this difference by analyzing $C_L^{\ell\dagger} P_{\beta} C_L^L$. For the circular entanglement map, it is straightforward to see that $P_{\gamma'} = C^{\dagger} X^{\otimes n} C = \mathbb{I} X \mathbb{I}^{\otimes(n-2)}$. This brings us back to the case of the local X observable with $\mathcal{L}_{\gamma'}^{(L-1)}$ in Eq. (C56), resulting in the same lower bound. Conversely, for the pairwise entanglement map, we have $P_{\gamma'} = C^{\dagger} X^{\otimes n} C = (\mathbb{I} X)^{\otimes n/2}$, introducing an additional contribution of $((1 - e^{-2\sigma^2})/2)^n$ in Eq. (C57). Indeed, we can easily find that $w(C_L^{\ell\dagger} P_{\beta} C_L^L) = n$ or $n/2$ for all $\ell = 1, \dots, L$. Therefore, this higher weight of the X observable leads to the exponential decay of the variance.

Figure 21 shows the variance of the loss function $\text{Var}_{\Theta}[\mathcal{L}_{\alpha}]$ of EfficientSU2 ansatz with $\text{poly}(n)$ depth versus the number of qubits n using the initialization range varying as $\sigma = 1/n$ for Z and X observables. As expected, the loss function scales as $\mathcal{O}(1/n^b)$ with $b > 2$, clearly proving the mitigation of the barren plateau in the polynomial depth circuit.

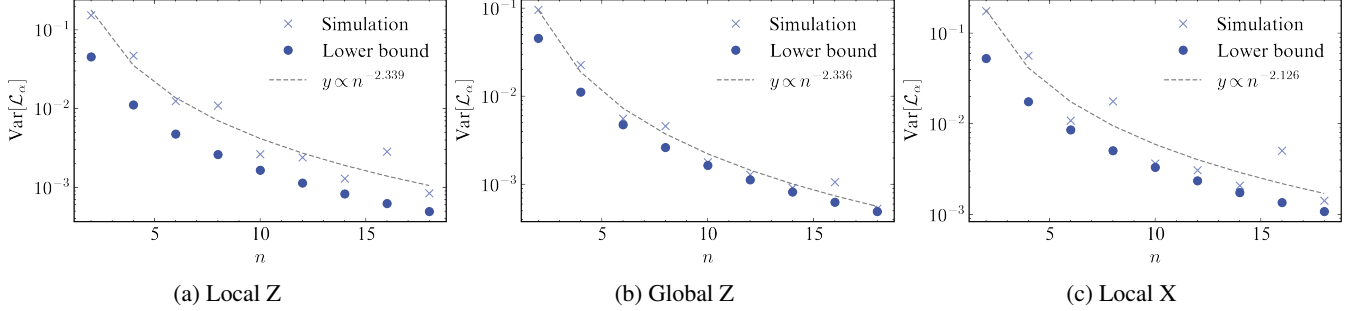


Figure 21. **Variance of the loss function with respect to n using $\sigma = 1/n$.** The blue cross corresponds to the simulated result, the dashed line its polynomial fit, and the green points the theoretical bounds computed with (a), (b) Eq. (C50) and (c) Eq. (C57). As predicted, the variance decays as $\mathcal{O}(1/n^2)$ with σ scaling as $1/n$.

Appendix D: Study of BP with Other Circuits

Previously, in Section V, we only displayed the results for the variance of the gradients calculated with Circuit1 (see Figure 3) as a quantum generator in LaSt-QGAN. In this section, we compute the variance for other types of circuits to confirm that this absence of barren plateau is not only limited to a specific type of circuit. As shown on Figure 22, we observe that $\text{Var}_{\Theta, \phi}[\partial_{\nu} \mathcal{L}_G]$ decays polynomially with a small angle initialization for other circuits as well. Notably, the varying $\sigma = 1/n$ found in Appendix C3 ensures a polynomial decay, with the slope decreasing as the system size increases.

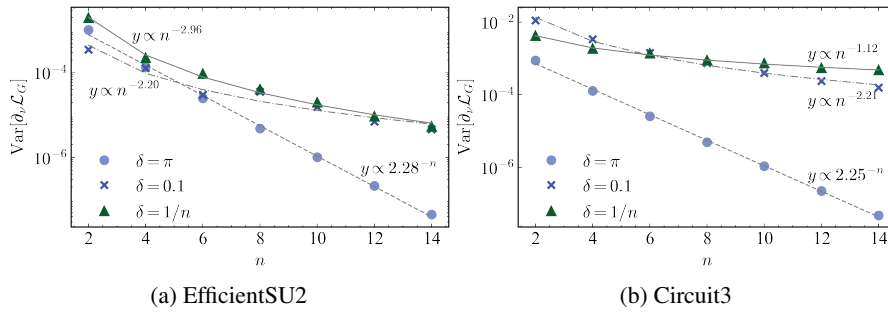


Figure 22. **Variance of the partial derivative of \mathcal{L}_G versus the number of qubits n using polynomial depth quantum generator with different circuit architecture in LaSt-QGAN.** The variance is computed with $D_{\mathbf{z}} = n$ for different initialization ranges, δ , and averaged over the parameters of the first layer. The quantum generator consists of different circuits presented in Figure 3 with polynomial depth, $d = \lfloor \log(n) \rfloor$.