Supplementary Information for "Quantum deep generative prior with programmable quantum circuits"

Tailong Xiao,[1,2,3,*] Xinliang Zhai,[1] Jingzheng Huang,[1,2,3] Jianping Fan,[4] and Guihua Zeng[1,2,3,†]

[1]*State Key Laboratory of Advanced Optical Communication Systems and Networks, Institute for Quantum Sensing and Information Processing, Shanghai Jiao Tong University, Shanghai 200240, China*
[2]*Hefei National Laboratory, Hefei, 230088, China*
[3]*Shanghai Research Center for Quantum Sciences, Shanghai, 201315, P.R. China*
[4]*Lenovo Research, Beijing 100094, China*

(Dated: July 27, 2024)

SUPPLEMENTARY NOTE 1: PARAMETERIZED QUANTUM CIRCUITS

PQCs represent a common framework for quantum machine learning design [1, 2]. Given the constraints of near-term hardware, hardware-efficient PQCs emerge as the most viable option for experimentally exploring the practical advantages of quantum computation [3]. Several noteworthy quantum machine learning algorithms have been proposed based on PQC, encompassing the quantum Boltzmann machine [4], quantum autoencoder [5, 6], and quantum reinforcement learning [7, 8]. These algorithms have been scrutinized for tasks such as MNIST classification, reconstruction, and low-dimensional distribution learning. Despite these investigations, there has been limited focus on their practical applications in the classical world. Classical machine learning has achieved remarkable success in computer vision, natural language processing, and audio signal processing [9]. Quantum machine learning, facing challenges in near-term hardware, appears less amenable to handling real-world tasks. To bridge this gap, our focus is on investigating the practical advantages of quantum machine learning in real, rather than toy, applications. Accomplishing this task solely using near-term quantum computers is challenging. Hence, we propose a hybrid quantum-classical quantum machine learning algorithm. In this approach, a classical neural network processes the heavy load of information, while quantum machine learning generates essential patterns. Consequently, quantum machine learning seamlessly complements the classical processing. This strategy aims to propel quantum computation into cutting-edge classical applications. Pretraining and fine-tuning, extensively explored in classical machine learning, have become powerful frameworks for training large models like ChatGPT, BERT, and others, which exhibit impressive performance in natural language processing and multi-modal learning. Recognizing this, PQCs hold great potential as pivotal components to enhance the out-of-distribution generalization ability of large models.

Another critical aspect of PQCs is their resilience to noise. Given the inherent noise in near-term quantum devices, qubits are susceptible to both coherent and incoherent errors. In this study, we specifically focus on coherent errors, characterized by the random insertion of Pauli gates into the circuit. For clarity, we assume that single-qubit Pauli gates are perfect, while two-qubit gates exhibit noise. To simulate errors in two-qubit gates, we consider each two-qubit gate to undergo two single-qubit depolarization errors. As different qubits experience distinct noise, a unique random seed is assigned to each two-qubit gate, leading to diverse Monte Carlo simulations. Furthermore, we assume the error types to be consistent within a layer during coherent time. This assumption ensures the use of the same random seed for different layers of the PQC. The representation of the noisy quantum circuit is provided in Supplementary Figure 1. The depolarization channel can be mathematical described by the linear combination of Kraus operators

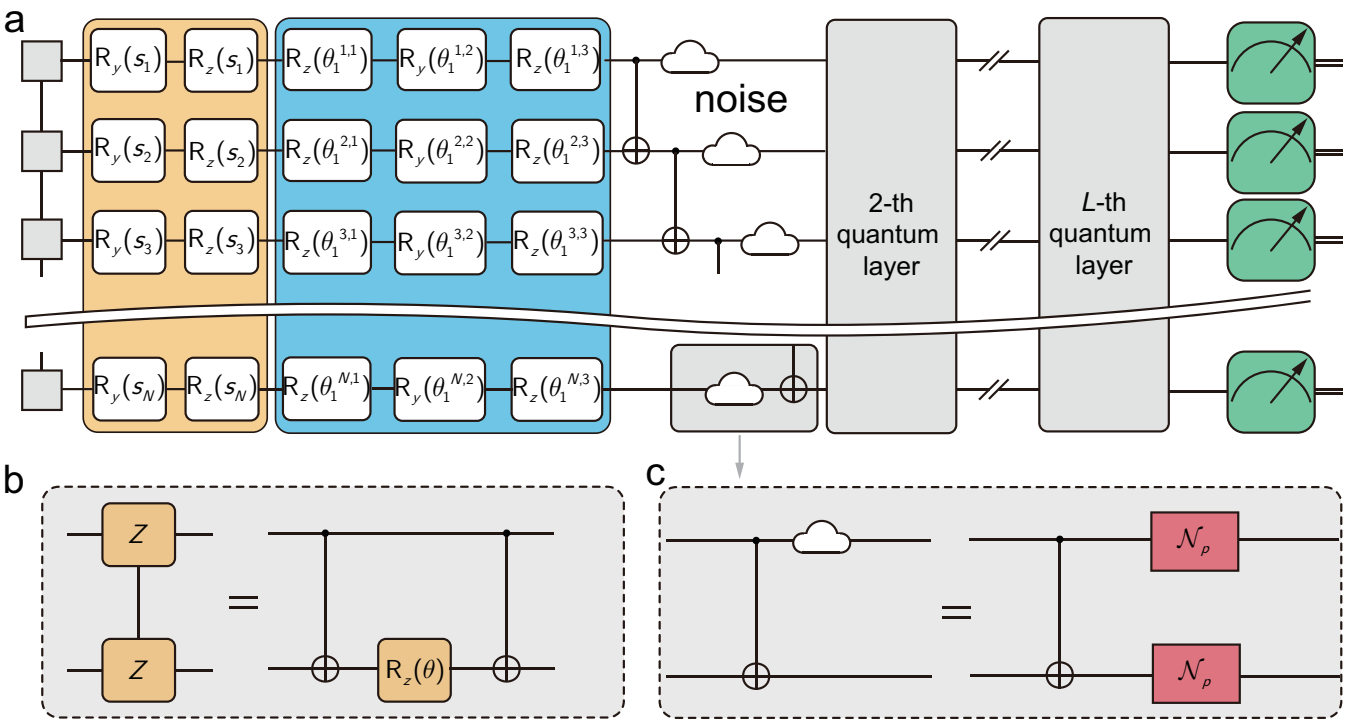
$$\mathcal{N}_p(\rho) = \sum_{i=0}^{3} \mathbf{K}_i \rho \mathbf{K}_i^\dagger, \tag{1}$$

where p denotes the depolarizing rate, ρ denotes the original quantum state and the Kraus operators can be written as

$$\mathbf{K}_0 = \sqrt{1-\frac{3p}{4}}\mathbf{I}, \quad \mathbf{K}_1 = \sqrt{\frac{p}{4}}\mathbf{X}, \quad \mathbf{K}_2 = \sqrt{\frac{p}{4}}\mathbf{Y}, \quad \mathbf{K}_3 = \sqrt{\frac{p}{4}}\mathbf{Z}. \tag{2}$$

The Kraus operators must satisfy the completeness relation, i.e., $\sum_i \mathbf{K}_i\mathbf{K}^\dagger = \mathbf{I}$. The depolarization channel $\mathcal{N}_p(\rho)$ can be understood as a uniform contraction of the Bloch sphere, parameterized by p. When $p = 1$, the channel returns the maximally-mixed state for any input state ρ, leading to the complete contraction of the Bloch sphere to the single point $\mathbf{I}/2$ at the origin.

* tailong_shaw@sjtu.edu.cn
† ghzeng@sjtu.edu.cn



Supplementary Figure 1. The detailed noisy quantum circuit structure for generating quantum latent code is illustrated as follows: (a) Parameterized quantum circuits are composed of interleaved encoding layers and variational layers. After L layers, the final quantum state is measured to obtain classical statistics. (b) The exponential **ZZ** gate can be decomposed into elementary quantum gates such as the **CNOT** gate and the rotational Pauli-Z gate. (c) The noisy depolarization operation follows the two-qubit gates with a depolarization rate p.

To simulate varied quantum noise levels, the depolarization rate can be adjusted. In practical quantum hardware, single-qubit gates can be prepared with high fidelity, while two-qubit gate fidelity is nearly 99%. Therefore, choosing $p = 0.01$ effectively simulates real noise. Consequently, the depolarization gate is placed after the two-qubit gate, as illustrated in Supplementary Figure 1c. Notably, when the noise rate is 0, the quantum circuit operates perfectly.

SUPPLEMENTARY NOTE 2: ADDITIONAL RESULTS

All the numerical results of QDGP and DGP evaluated with ImageNet validation set are presented in Supplementary Table I,II,III.

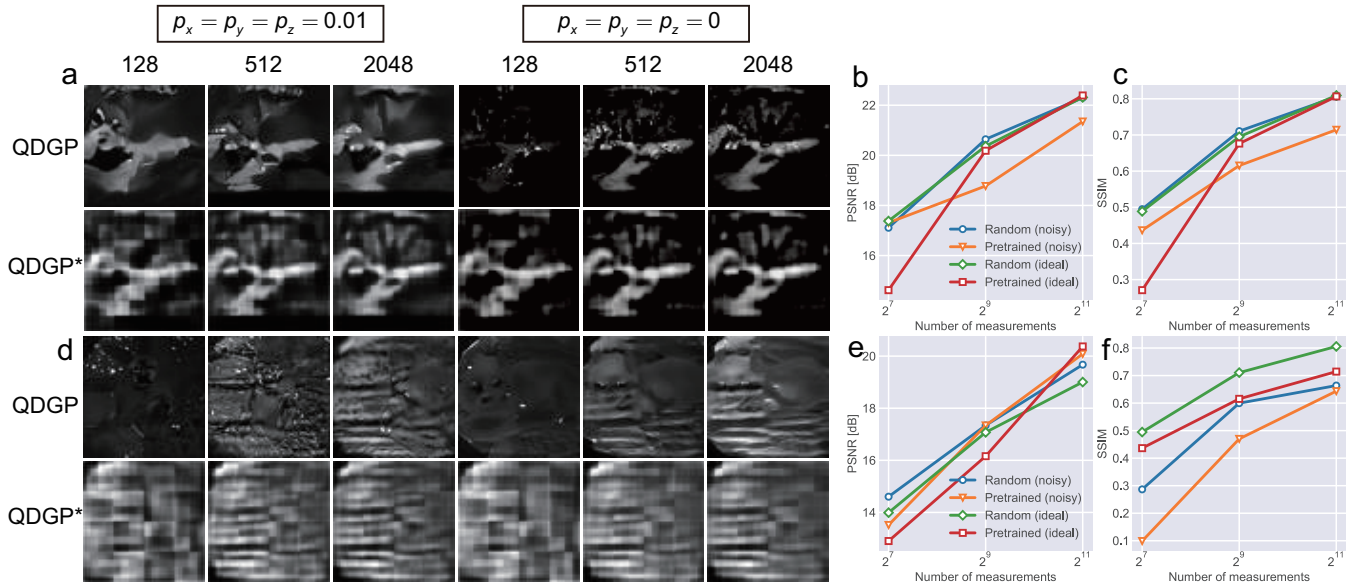
Supplementary Table I. Colorization performance. Deep generative prior (DGP) and quantum deep generative prior (QDGP) share the same hyperparameter settings and both use progressive reconstruction method. The mean and standard variance are calculated over independent 20 runs.

	DGP	QDGP
PSNR [dB]↑	21.93 (±0.022)	23.52 (±0.025)
SSIM [1e-2]↑	90.11 (±0.026)	93.08 (±0.031)

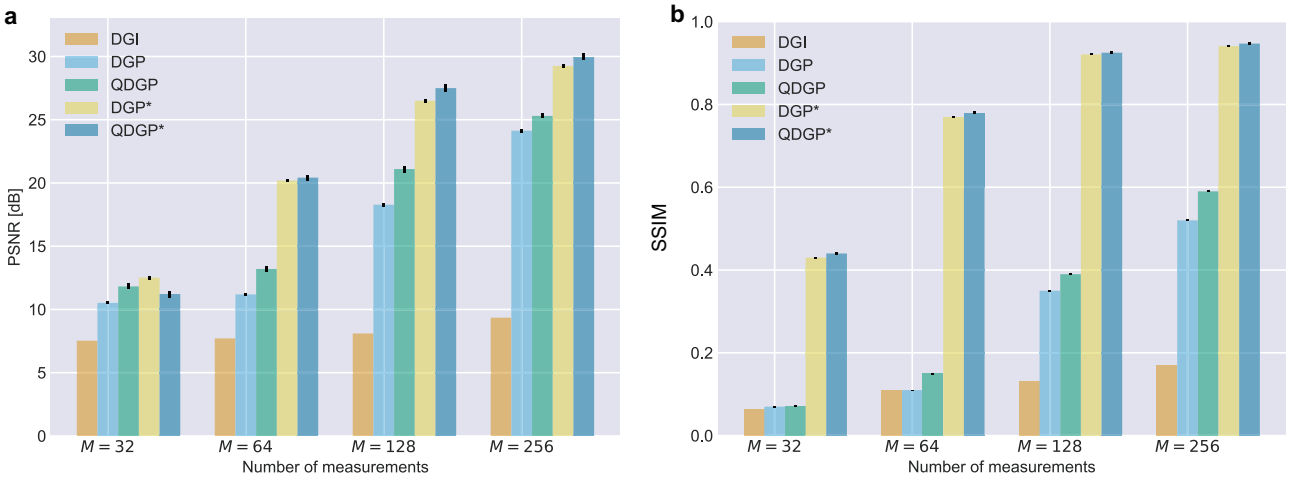
Supplementary Table II. Inpainting evaluation. Deep generative prior (DGP) and quantum deep generative prior (QDGP) share the same hyperparameter settings and both adopt feature and MSE loss. The reported mean and standard variance are obtained over independent runs.

	DIP	DGP	QDGP
PSNR [dB]↑	14.51 (±0.02)	16.90 (±0.03)	17.35 (±0.04)
SSIM [1e-2]↑	29.23 (±0.11)	45.76 (±0.12)	47.21 (±0.13)

We evaluate the performance of colorization of QDGP and DGP on the 1k ImageNet validation set. The numerical results can



Supplementary Figure 2. The experimental results of QDGP under noisy quantum circuits. (a) The reconstructed image of butterfly with $64 * 64$ resolution under pretrained generator (QDGP) and random initialized generator (QDGP*). (b) and (c) are the reconstruction metrics of PSNR and SSIM for ideal and noisy quantum circuits. (d) The reconstruction results of wing with $64 * 64$ resolution under the same setting. (e) and (f) are the reconstruction metrics of PSNR and SSIM under the noisy and ideal quantum circuits for QDGP and QDGP*.



Supplementary Figure 3. The numerical results of QDGP, DGP and DGI methods on ghost imaging for MNIST dataset. (a) PSNR results; (b) SSIM results. The error bars are the standard deviation of the repeated runs.

be found in Supplementary Table I. The numerical results on 1k validation set also showcase higher PSNR and SSIM value of QDGP over DGP method. In addition, in Supplementary Note 4, we analysis the performance of DGP under different learning rate ratios both in inpainting and colorization tasks. The learning rate ratios are measured by the typical learning rate of DGP and QDGP. We find that tuning the learning rate including increasing and decreasing does not fill the performance gap between the QDGP and DGP. Controlling the latent code to be more or less sensitive is not necessary to obtain the optimal performance. However, fine-tuning the learning rate indeed shrinks the gap to some extent. Noteworthy, carefully tuning the learning rate of quantum parameters will also lead to better performance.

We further conduct experiments on the 1k ImageNet validation set to assess the performance of QDGP. The results are presented in Supplementary Table III. For the NIQE metric [10], DGP exhibits smaller values than both DIP and QDGP. However, in terms of metrics such as PSNR and RMSE, QDGP outperforms DIP and DGP. NIQE is designed to measure the quality of natural images without a reference image. The superior quality metrics of DGP with a Gaussian prior suggest that DGP, equipped with a Gaussian laten code, is well-suited for super-resolving natural images. In contrast, QDGP demonstrates larger PSNR and lower RMSE values, indicating a stronger fitting ability than DGP when utilizing a quantum latent space code. Both of these metrics

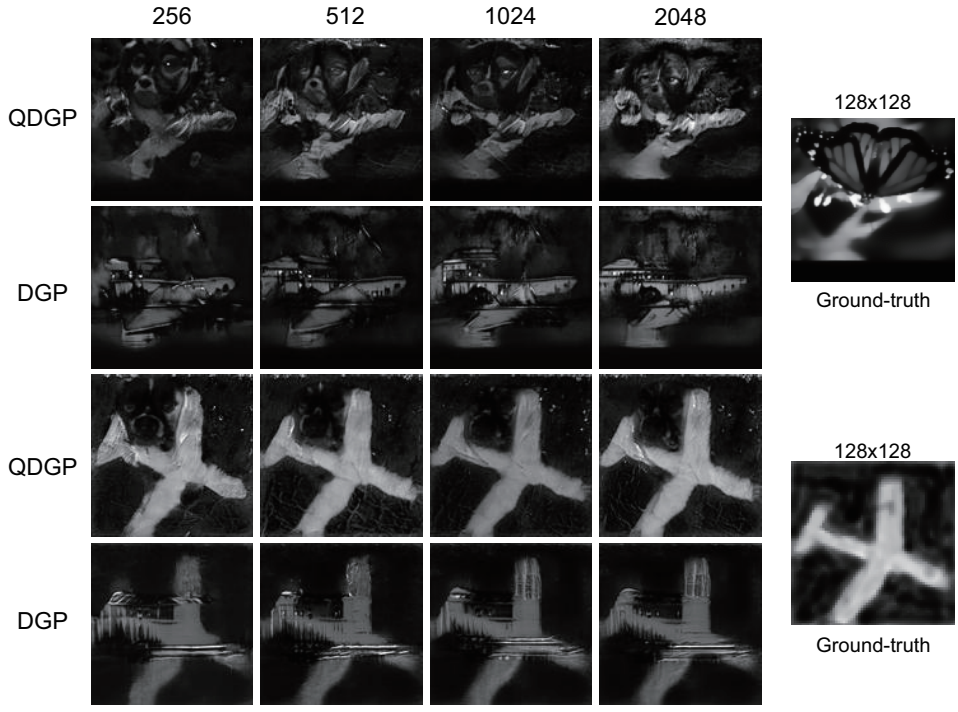
Supplementary Table III. Super-Resolution ($\times 4$) performance with different metrics. Deep generative prior (DGP) and quantum deep generative prior (QDGP) share the same hyperparameter settings and both adopt feature and MSE loss. The reported mean and standard variance are obtained from independent 20 runs.

	DIP	DGP	QDGP
NIQE↓	6.10 ($\pm$0.01)	4.98 ($\pm$0.02)	5.03 ($\pm$0.03)
PSNR↑	22.71 ($\pm$0.02)	23.02 ($\pm$0.03)	23.40 ($\pm$0.03)
RMSE↓	17.83 ($\pm$0.01)	17.58 ($\pm$0.01)	17.50 ($\pm$0.02)

focus on pixel differences, and PQC can be trained to generate a latent code that aligns with the degradation process. In scenarios emphasizing pixel differences, QDGP exhibits a potential advantage over DGP.

We extend our analysis by randomly sampling 200 images from the MNIST test dataset and performed optical ghost imaging experiments to gather the bucket signals along with their corresponding illumination speckles. These MNIST samples consist of handwritten digits not present in the ImageNet dataset. We examine the PSNR and SSIM metrics across various measurements $32, 64, 128, 256$ for the DGI, DGP, and QDGP techniques. Each measurement involved running QDGP and DGP with 20 independent iterations, initialized differently, to compute the mean and standard deviation of the metrics. The numerical outcomes are detailed in Supplementary Figure 3.

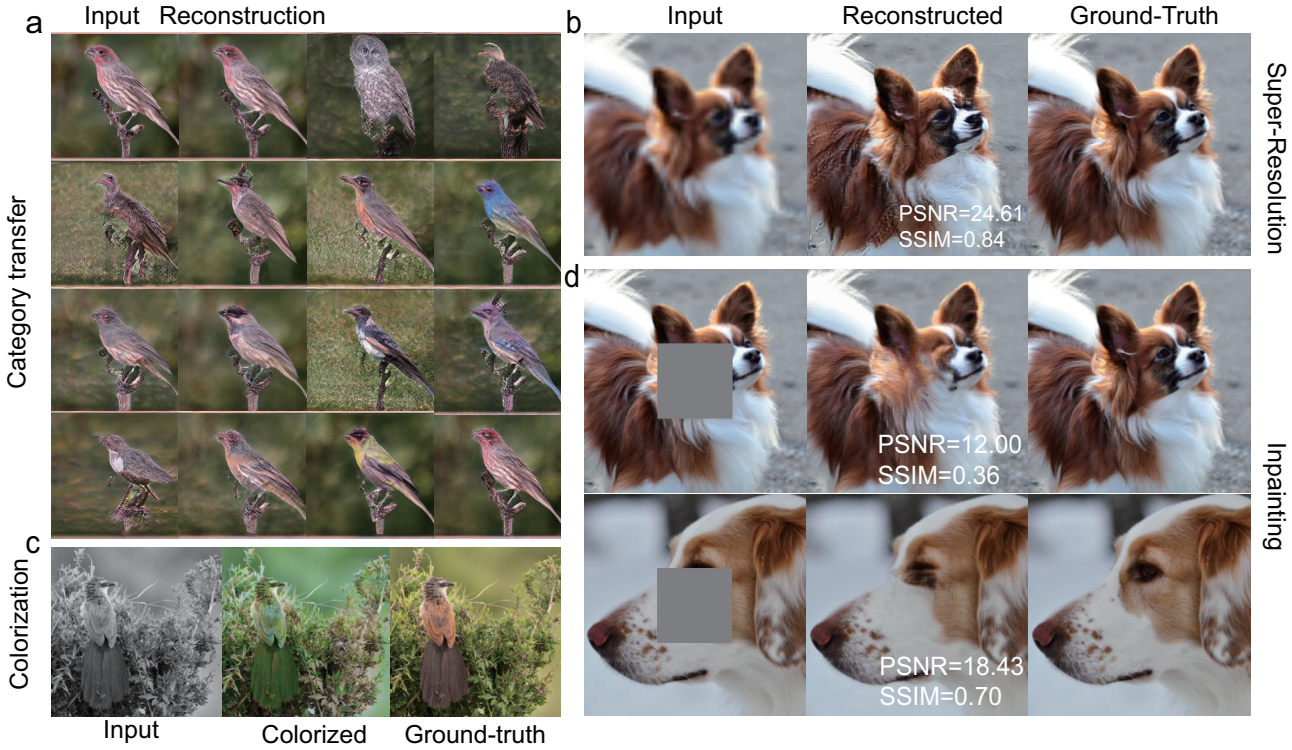
From the results, both QDGP and DGP shows significant advantage over DGI in terms of imaging reconstruction quality. When using pretrained GAN, QDGP performs better than DGP for $M = 64, 128, 256$ both in terms of PSNR and SSIM metrics. For $M = 32$, QDGP achieves nearly the same SSIM value with DGP. With randomly initialized GAN, QDGP weights performs worse than DGP for $M = 32$. For $M = 64, 128, 256$, QDGP shows competitive performance with DGP. Therefore, QDGP has no tangible improvement over DGP for different sampling rates when using randomly initialized GANs. The performance of recovering images from pretrained BigGAN models demonstrates the potential of PQC-induced distributions to enhance the out-of-distribution generalization ability of pretrained models.



Supplementary Figure 4. The reconstruction performance of QDGP and DGP with pretrained generator. The reconstructed images are based on the same number optimization iterations and the same pretrained generator. QDGP and DGP have different latent space.

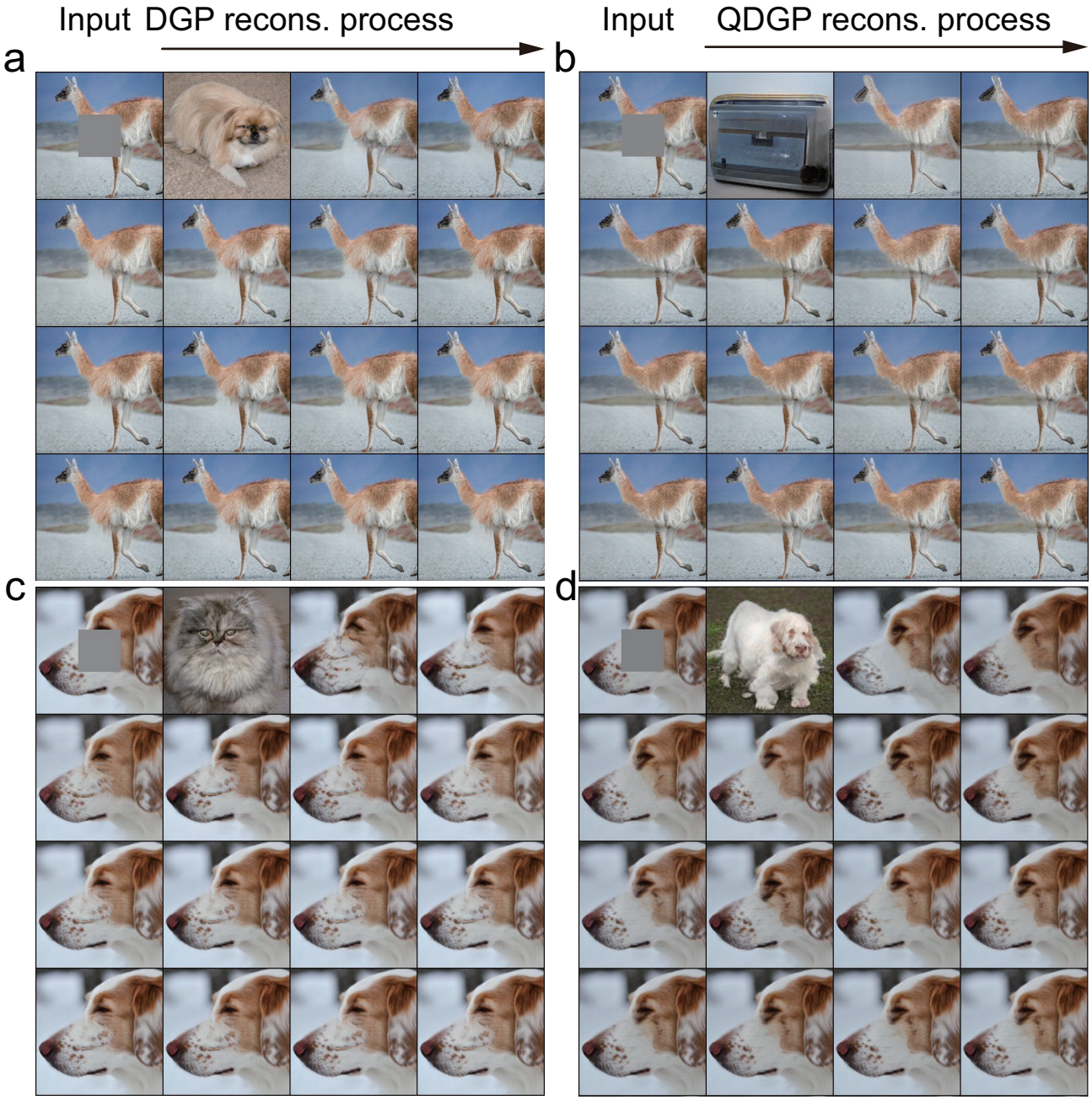
In this section, we present additional results from a noisy quantum generative model applied to ghost imaging and image restoration. The reconstruction results of QDGP with noisy PQCs are illustrated in Figure 2. Specifically, we focus on reconstructing samples labeled as 'butterfly' and 'wing' to showcase the robustness of QDGP based on noisy PQCs. Consistent with the main text notation, QDGP and QDGP* denote algorithms based on pretrained and randomly initialized generators, respectively. Figures 2a and 2d clearly depict degraded reconstruction performance when quantum circuits are noisy. Noisy quantum

circuits can introduce bias and large estimation variance to observables. While expectation bias does not affect algorithm convergence due to compensatory adjustments in successive classical neural networks, large observable variance hampers optimization convergence by yielding different expectations in each iteration. Figures 2b and 2c present numerical metrics (PSNR and SSIM) for reconstructed images under various sampling rates. Under noisy PQCs, QDGP with a pretrained generator experiences performance degradation. However, for randomly initialized QDGP, performance remains unaffected, achieving nearly the same results as with ideal PQCs. For unseen images, the generalization ability of the pretrained generator is impacted, despite the quantum latent space's potential to enhance it. Randomly initialized QDGP employs the neural network structure as an inductive bias to compensate for variance and bias in quantum latent distribution, making it resilient against quantum noise. The recommendation is to use a randomly initialized generator when quantum latent space requires extensive iterations for adjusting well-initialized parameters, as seen in noisy cases. Notably, for reconstructing the 'wing' sample, experiments under noisy and ideal conditions demonstrate the noise effects for QDGP. As anticipated, quantum noise in PQCs decreases QDGP's performance, particularly for pretrained QDGP. Surprisingly, randomly initialized QDGP exhibits strong resilience against quantum noise, with larger PSNR values and smaller SSIM values in noisy cases compared to ideal ones. The physical forward model's use in QDGP as a loss function, combined with correct bucket signals, mitigates the impact of neural network uncertainties on optimization convergence, countering noise effects. Moreover, pretrained QDGP under noisy PQCs demonstrates nearly the same PSNR value as the noise-free case, despite a smaller SSIM value. This result is surprising compared to the 'butterfly' reconstruction, highlighting the influence of noisy quantum latent space on the generalization of a pretrained generator. Overall, the results underscore the robustness of QDGP against quantum noise.



Supplementary Figure 5. The image restoration and manipulation results of QDGP under noisy parameterized quantum circuits. Note that the images are not lies in the ImageNet dataset. (a) The category transfer results of QDGP under noisy quantum circuits. The chosen image is the class of bird and transfers to other categories. (b) The image super-resolution results under noisy quantum circuits.(c) The image colorization results of QDGP with noisy quantum circuits. (d) The image inpainting results for both two dogs by using QDGP under noisy circuits.

We present experimental results for various samples at a 128^2 resolution using a ghost imaging system. The measurement times are configured as $256, 512, 1024, 2048$. As illustrated in Supplementary Figure 4, QDGP consistently outperforms DGP across different measurement scenarios. In the second row of Supplementary Figure 4, the reconstructed image exhibits a clear background corresponding to other categories. Achieving optimal performance requires a larger loss constraint (higher dimension). QDGP possesses a latent space that deviates significantly from a Gaussian distribution, unlike the pretrained pattern based on a Gaussian distribution. Consequently, leveraging physical forward loss constraints, the PQC-generated latent space aligns with the generator pattern to reconstruct high-resolution images. Results in Supplementary Figure 4 further showcase the ability of quantum latent space to enhance the out-of-distribution generalization capability of the pretrained generator, as discussed in the main text.



Supplementary Figure 6. The inpainting reconstruction process of QDGP and DGP for two typical natural images is depicted below. Figures (a) and (c) show the reconstruction results of DGP with random and known category labels, respectively. On the other hand, figures (b) and (d) display the reconstruction results of QDGP with random and known category labels, respectively.

We present results for image restoration and manipulation under noisy quantum circuits, focusing on the outcomes of noisy QDGP. In Supplementary Figure 5a, it is evident that noisy QDGP successfully reconstructs images from different categories with similar postures as the input images. Figure 5b displays the results of super-resolution with noisy quantum latent space, showcasing minimal influence on performance, indicated by a recovered image with significantly high Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM). The main text confirms that the final super-resolution performance remains the same as in the ideal case. In Supplementary Figures 5c and 5d, experimental results of noisy PQC-generated latent space for colorization and inpainting are presented. Similar outcomes are observed for category transfer and super-resolution across these tasks. In all four tasks, we conclude that the presence of quantum noise in PQCs for QDGP has almost no effect on the final results compared to QDGP with noise-free PQCs. In essence, QDGP with PQCs enhanced quantum latent space proves to be robust to quantum noise, significantly enhancing the feasibility of PQCs in near-term quantum devices.

We explore the potential reasons why quantum noise has a limited impact on QDGP. Our speculation revolves around the role of the quantum latent space, primarily serving as a source of random sampling and diversity for the GAN generator. Quantum noise, to some extent, enhances the randomness and diversity of the latent space. In the early optimization stages, the generated images may appear non-natural. However, in the late optimization stage, under the constraints of feature and MSE loss, both quantum and classical parameters can be fine-tuned to generate highly natural images. Additionally, the quantum latent space enables the GAN generator to represent a complex data distribution in a more compressed and structured form. Despite a potential reduction in distribution complexity with noisy PQCs, it remains sufficiently high to recover a single image based on the pretrained generator and discriminator.

It is crucial to emphasize that image restoration and manipulation tasks differ from ghost imaging (GI), particularly concerning QDGP for GI. The main disparity lies in the optimization constraints. In GI, the loss constraint involves experimental bucket signals and total variation loss, without discrimination loss. GI essentially aims to solve an undersampling inverse problem, presenting a more challenging task than image processing. This distinction also highlights the sensitivity of GI to quantum noise, given its limited prior information to serve as optimization constraints.

Finally, we present the inpainting reconstruction process in Supplementary Figure 6. It should be noted that the final inpainting results for both QDGP and DGP can be found in the main text. In Supplementary Figures 6a and 6b, it is evident that QDGP restores the image with a more natural appearance under the same optimization process, despite the substantial deviation of the randomly generated label (embedding) from the true image. This observation suggests that the parameterized quantum latent distribution possesses a robust self-healing capability and can generalize well to unseen images. When category labels are known, both methods exhibit competitive performance; however, as detailed in the main text, the final reconstructed image of QDGP achieves a higher PSNR and SSIM value. Moreover, the recovered region appears more natural compared to DGP, where speckles from the dog's nose area are displaced to the face.

SUPPLEMENTARY NOTE 3: HYBRID PQC WITH DEEP LAYERS

We present the outcomes of employing PQCs with deeper layers in a ghost imaging task enhanced by a physical forward model. The objective is to illustrate the influence of increasing quantum layers on the reconstruction efficacy. Initially, optical experiments were performed to capture the bucket signals, with the sampling rate set at 1024 and the image resolution at 64×64. By maintaining consistent sampling rates and resolution, we aim to isolate the impact of quantum layers on reconstruction performance. The number of quantum layers was varied between 6, 8, and 10 for PQCs utilizing 20 qubits. Additionally, for comparison purposes, MPS-based PQCs with a low bond dimension of 16 were simulated, enabling the emulation of a larger number of qubits. Furthermore, numerical experiments were conducted for both pre-trained and randomly initialized models.

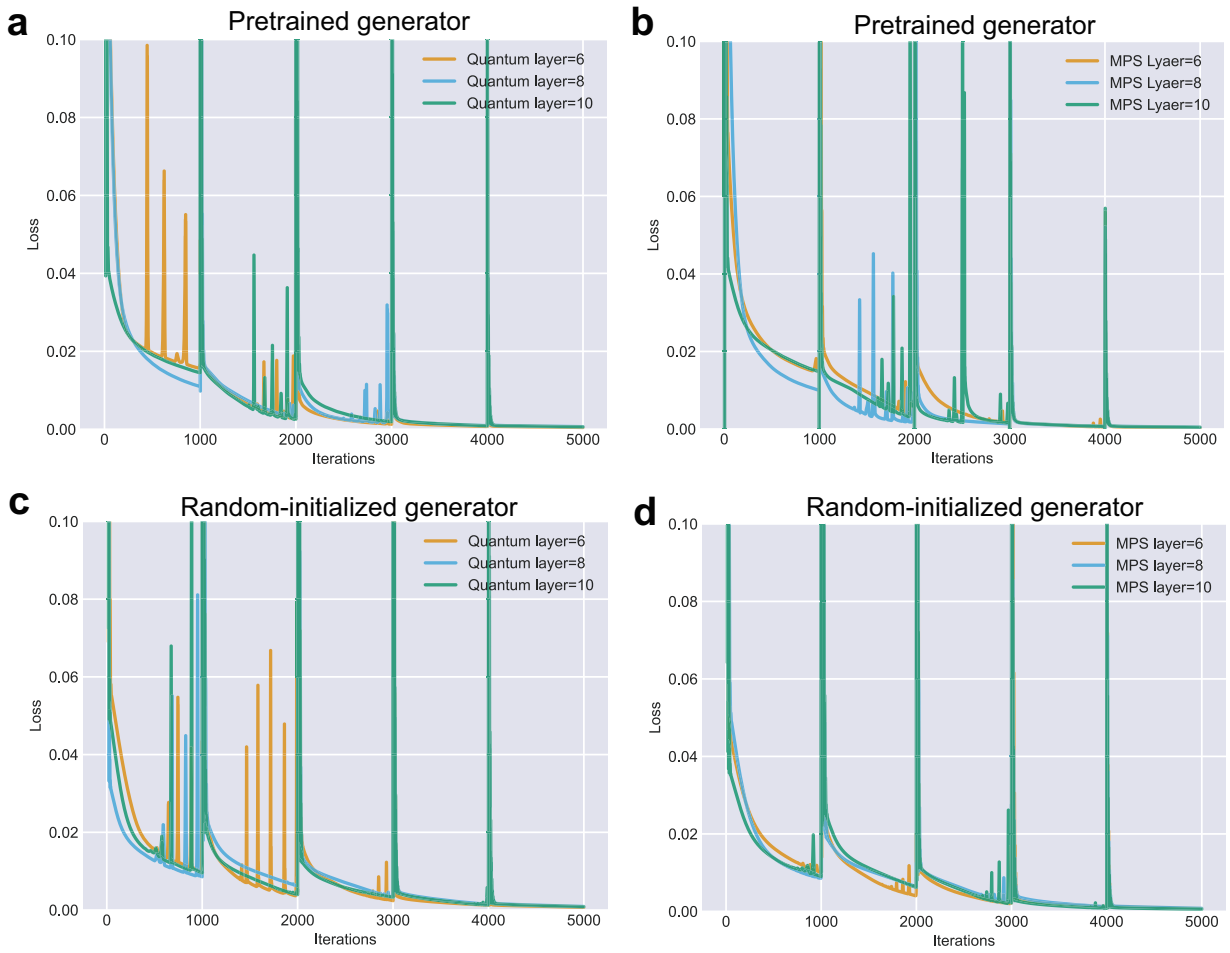
The loss curves of the optimization process were recorded for both MPS- and state-vector-based PQCs. Figure 7 illustrates that increasing the layers of PQCs, whether MPS-based or state-vector-based, has no impact on convergence. When the number of layers reaches 10, both types of PQCs exhibit a more noticeable shift in loss when switching the learning rate during optimization. This is anticipated as deeper PQCs involve more quantum parameters, leading to a greater variance in the induced distribution. Consequently, adjusting the learning rate prompts a larger variance in the quantum distribution compared to the distribution induced by previous parameters. These observations hold true for both MPS and state-vector-based PQCs. Importantly, these findings suggest that PQC-based QDGP is not significantly affected by the Barren Plateau problem.

There are two potential reasons for this outcome. Firstly, the PQC functions to generate the latent code for the subsequent BigGAN model rather than serving as the backbone of the machine learning model. Secondly, our proposed algorithm follows a hybrid structure, where the quantum latent code generated is subsequently processed by a classical neural network. Evidence indicates that the classical neural network can alter the loss landscape of quantum circuits [11, 12]. These two factors may elucidate why our hybrid deep quantum model remains unaffected by the Barren Plateau.

While the loss curves may appear similar, there are differences in the reconstruction performances of MPS- and state-vector-based PQC-induced BigGAN models. Figure 8 illustrates the PSNR values of images reconstructed by different methods across various quantum layer configurations.

The PSNR values depicted in Figure 8 reveal that increasing the layers of PQCs in QDGP leads to a slight decrease in the PSNR value. Conversely, the PSNR value of QDGP-MPS remains unaffected by additional quantum layers. However, the PSNR value of the QDGP-MPS algorithm is marginally lower than that of QDGP. Notably, as the quantum layers reach 10, QDGP-MPS surpasses the performance of QDGP. This suggests that while deep quantum layers diminish the performance of QDGP, QDGP-MPS exhibits a higher layer threshold before encountering the Barren Plateau phenomenon. Excessive entanglement, which can induce the Barren Plateau, is mitigated in QDGP-MPS due to its incapacity to precisely characterize quantum states with high quantum entropy, stemming from its low bond dimension. Consequently, the Barren Plateau phenomenon does not manifest in the QDGP-MPS algorithm.

When employing a randomly initialized generator, the PSNR outcomes align with those detailed in the main text. Notably, randomly initialized QDGP outperforms pretrained models when reconstructing out-of-distribution samples. In this scenario, QDGP also exhibits performance deterioration with increasing quantum layers. In contrast, QDGP-MPS, by constraining the



Supplementary Figure 7. The loss curves of the optimization process for both MPS-based and state-vector simulator-based PQCs induced in the BigGAN model for ghost imaging are depicted. Specifically, Figure (a) shows the loss curve of QDGP with a pretrained generator, while Figure (b) illustrates the loss curve of QDGP-MPS with a pretrained generator. Figure (c) displays the loss curve of QDGP with a randomly initialized generator, and Figure (d) presents the loss curve of QDGP-MPS with a randomly initialized generator.

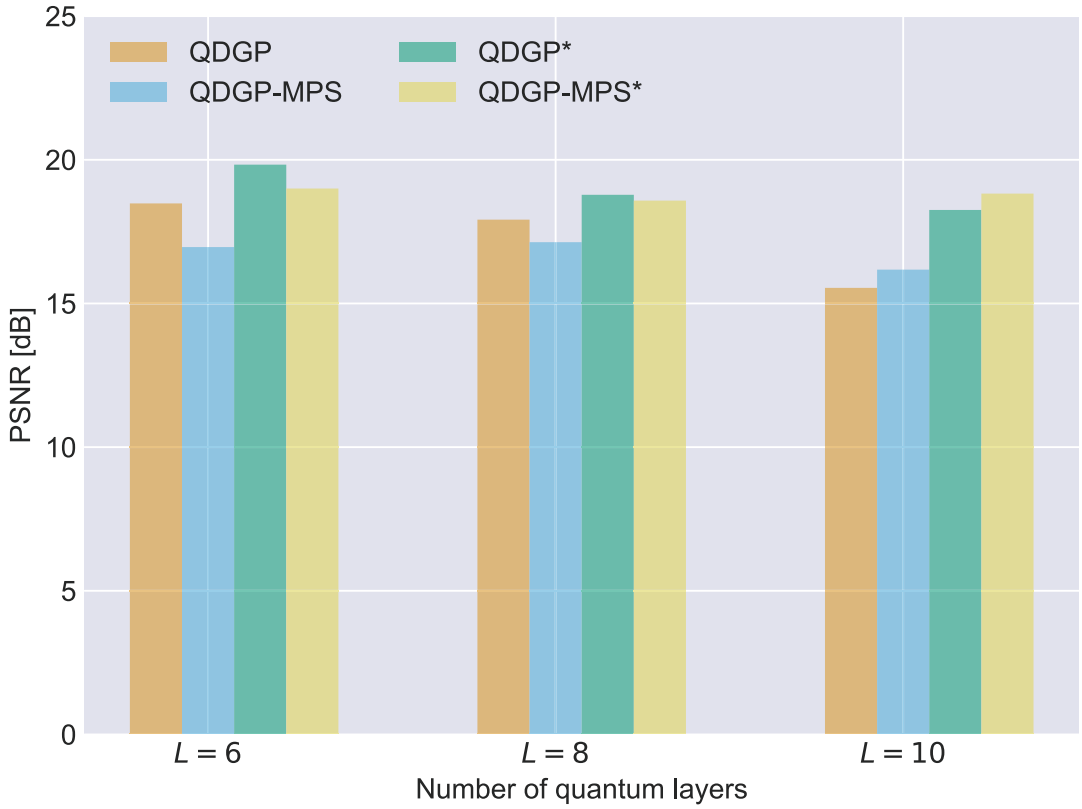
bond dimension and mitigating the entanglement of quantum states, avoids performance degradation with additional quantum layers. These findings imply that, in practical applications, programmable quantum circuits with appropriate quantum layers are preferable over deep quantum layers, which pose greater training challenges. MPS can facilitate the pretraining of deep quantum circuits, subsequently fine-tuned using actual quantum hardware. Moreover, MPS with optimized bond dimensions can emulate the characteristics of programmable quantum circuits, bolstering classical machine learning algorithms. Furthermore, MPS can serve as an indicator of entanglement. Analyzing performance with MPS at various quantum layers enables the inference of the requisite quantum layers for real quantum computers based on the bond dimension.

SUPPLEMENTARY NOTE 4: LEARNING RATE ANALYSIS

In this analysis, we examine the performance of QDGP and DGP under varying learning rates. Initially, we determine the typical learning rates for both QDGP and DGP in the colorization task. Subsequently, we compute the learning rate ratios between these two algorithms. Lastly, we maintain the initial latent code as z_0, which remains constant throughout optimization. Additionally, we introduce another trainable latent code initialized to zero and denoted as d_z. We update the latent code using the following equation:

$$z = z_0 + k \times d_z, \tag{3}$$

where k denotes the learning rate ratios. We observed that the rate ratio k can adjust the sensitivity of the latent code by scaling it k times, either more or less sensitive compared to the previous latent code. Through numerical experiments in colorization tasks

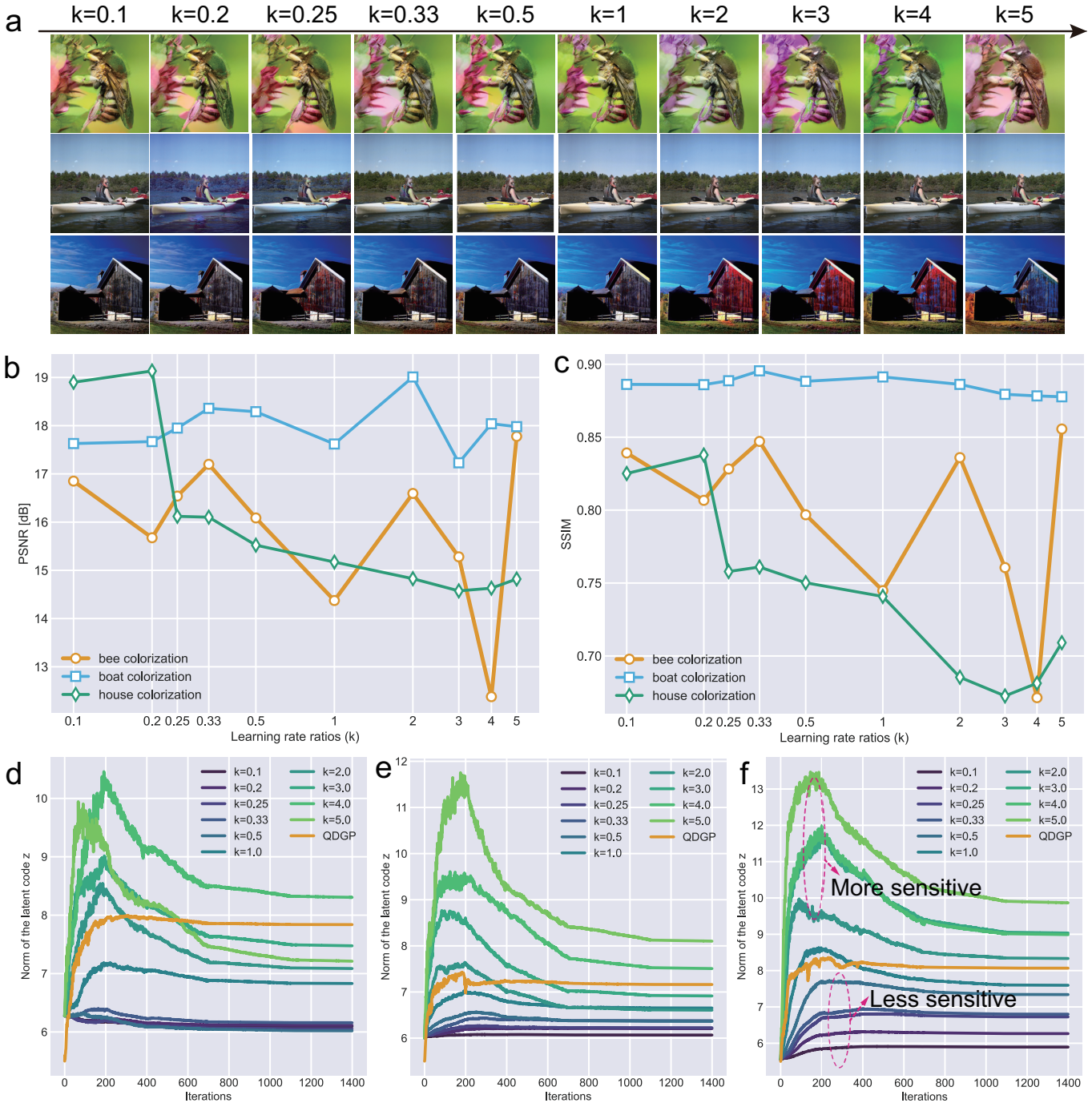


Supplementary Figure 8. The PSNR values of images reconstructed by the QDGP and QDGP-MPS algorithms with both pretrained and randomly initialized generators are depicted.

involving three images named ”bee,” ”boat,” and ”house,” we analyze the impact of the latent code’s sensitivity on performance. Figure 9a showcases the colorized images using DGP under varying learning rate ratios. Figures 9b and 9c display the PSNR and SSIM values calculated from the colorized images at different k values. Figures 9d, 9e, and 9f illustrate the norm of the latent code for different k values across iterations. The colorized images of these three objects by QDGP are provided in the main text, along with their corresponding PSNR and SSIM values: 19.06 and 0.8918 for the bee, 20.51 and 0.8917 for the boat, and 21.13 and 0.8625 for the house. Comparison of the results in Figures 9b and 9c reveals that adjusting the learning rate does not bridge the performance gap between QDGP and DGP. Notably, the colorization performance for the ”boat” appears less sensitive to the rescaling factors. Conversely, for the ”house” object, performance deteriorates with increasing learning rate ratios, while for the ”bee”, it oscillates. Figures 9d, 9e, and 9f indicate the rescaling effect on the norm of the latent code, displaying a consistent pattern of initially increasing to a peak value and then stabilizing. Particularly for small ratios ($k < 1$), the latent code’s norm exhibits minimal variation compared to the initial value z_0. The yellow curves in Supplementary Figures 9d, 9e, and 9f represent the quantum latent code curve, which aligns with the learning rate ratio trend at $k = 1$. However, unlike the norm of the latent code, the norm of the quantum latent code does not decrease after reaching the peak value. From the results in Supplementary Figures 9(b-f), we observe that colorization performance does not strictly correlate with the norm of the latent code. Both large and small variations in the norm can lead to improved performance, indicating that the pattern of the latent distribution may play a crucial role in influencing the QDGP algorithm’s performance. Consequently, while the norm of the latent code reflects the distribution’s global property, it may not fully capture its characteristics.

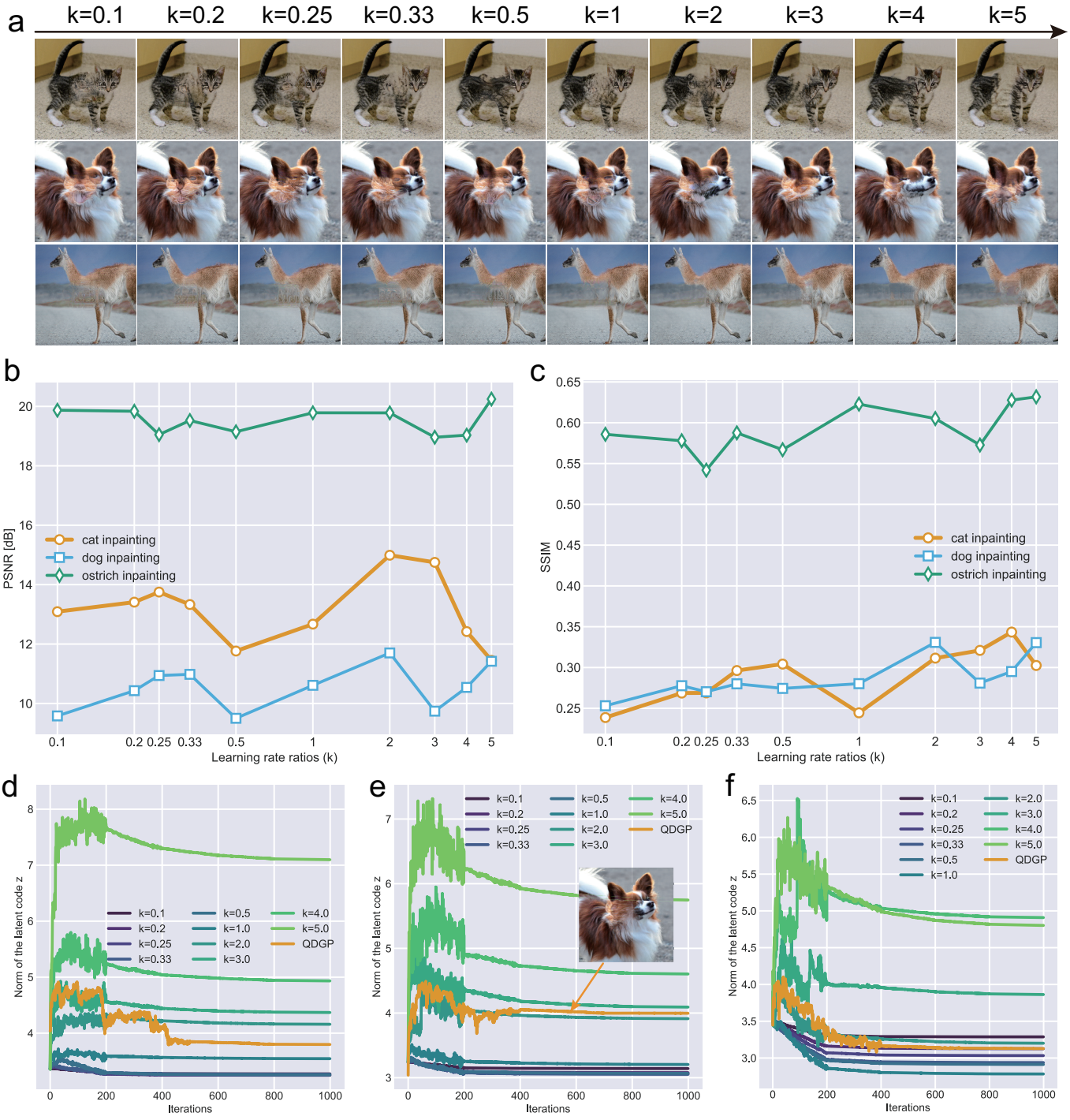
To analyze the inpainting performance of QDGP and DGP under typical learning ratios. We also conduct numerical experiments for randomly chosen three images. The results can be shown in Supplementary Figure 10.

From the inpainting results of Supplementary Figure 10a, we can see that the inpainted images can have a good visual appearance. For ”ostrich” image, the inpainting performance is the best compared to other two images. In fact, painted area of the ”ostrich” image has no complex features. Using DGP, the area can be well recovered. From the PSNR and SSIM values in Supplementary Figure 10b,c, it also validates our inference from visual appearance. Besides, we can find that when changing the learning ratio, these three images has no clear performance increase or drop. Note that it does not conclude that the learning rate of latent code is not important. Conversely, when changing the learning rate, we always can find a best one to achieve the largest PSNR and SSIM. Compared to the results of colorization, the learning rate ratios have no large impact on the final performance. We suspect that the inpainting task is much harder than colorization task. Search an proper latent code for the



Supplementary Figure 9. The learning rate analysis hyperparameters for the classical DGP are presented. The learning rate ratio k is computed between the standard learning rates of QDGP and DGP. A smaller k implies less sensitivity to the latent code, and vice versa. Subsequently, the following components are illustrated: (a) Colorized images based on DGP with varying learning rate ratios. (b,c) PSNR and SSIM measurements corresponding to the reconstructed images across different learning rate ratios. (d,e,f) Norm measurements of the latent code during the optimization process, where (d) pertains to the bee, (e) to the boat, and (f) to the house.

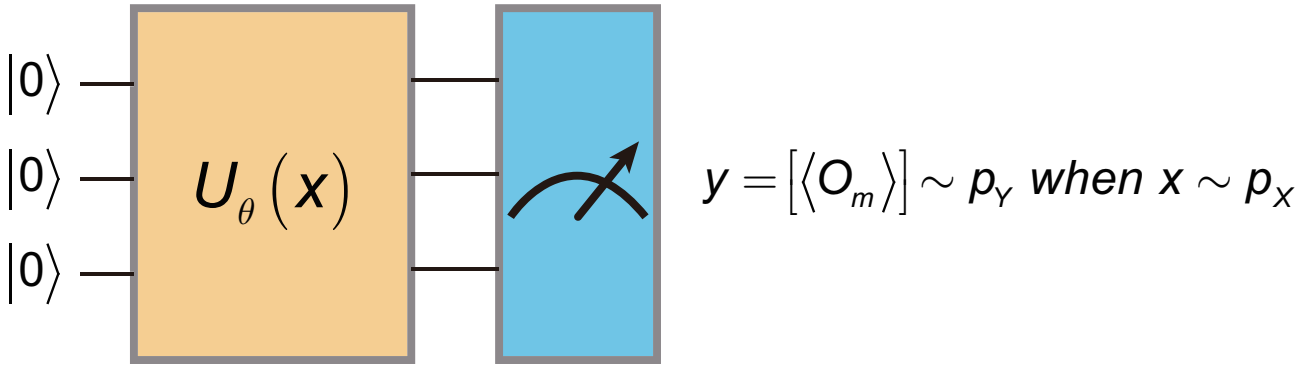
optimization is also much harder than colorization. Tuning the learning rate of DGP can influence the inpainting performance although the significance is limited. Moreover, from the results of Supplementary Figure 10d,e,f, we also can find that the norm of the latent code increases rapidly at the early stage of training for learning rate ratio $k \geq 1$. Then, the norm of the latent code drops to a stable value. With large learning ratio, the norm has a large increment. We note that the yellow curves denote the norm of the quantum latent code in inpainting the respective images. The PSNR and SSIM for cat images are 16.66 and 0.3497, for dog are 13.99 and 0.2854 (see the insert figure), for ostrich are 20.95, 0.6522. We note that for comparison with the results



Supplementary Figure 10. The numerical inpainting results of DGP under different learning rate ratios. (a) The reconstructed images of DGP under typical learning rate ratios. (b,c) The calculated PSNR and SSIM from DGP with various learning rate ratios. (d,e,f) The norm of the latent code varying with the iterations under different k, where (d) is for "cat" inpainting, (e) is for "dog" inpainting, (f) is for "ostrich" inpainting.

of random labels, the inpainting results of ostrich image is with random label. The corresponding norm curve is also recorded in Supplementary Figure 10f. We can find that for random label, the inpainting performance is consistent with the results reported in the main text. QDGP still outperforms DGP. Since inpainting ostrich is relatively a simpler task than inpainting cat and dog, the gap of DGP and QDGP are not large which may be filled by tuning the learning rate ratios carefully. Similar to the results of colorization, changing the learning rate ratios can influence the algorithm performance but cannot fill the performance gap of the DGP and QDGP.

SUPPLEMENTARY NOTE 5: EXPECTATION VALUE SAMPLING MODEL



Supplementary Figure 11. The Expectation Value Sampling Model involves encoding a randomly sampled vector into a quantum state through a parameterized quantum circuit. Subsequently, the expectation values of fixed observables are computed and returned as another random vector.

Here, we briefly introduce the expectation value sampling model based on parameterized quantum circuits. Firstly, we present the main result of reference [13]. The definition of Generalized Polynomial Chaos Expansion (GPCE) is provided as follows:

Definition 1. *A generalized chaos expansion is characterized by a probability density function p_X defined on the support $\mathcal{X} \subseteq \mathbb{R}^M$ with finite moments (usually chosen as standard distributions, such as Gaussian or uniform). This choice defines an inner product for functions in $\{f : \mathcal{X} \to \mathbb{R}\}$:*

$$\langle f|g\rangle_{p_X} = \int_{\mathcal{X}} f^*(x)g(x)p_X(x)dx. \tag{4}$$

This choice of inner product comes with the choice of an order family of functions, usually polynomials, that are orthonormal with respect to the above inner product:

$$\Phi_{p_X} = \{\phi_l : \mathcal{X} \to \mathbb{R}, \forall(k,l), |\phi_l\rangle\langle\phi_k| = \delta_{k,l}\}. \tag{5}$$

A generalized chaos expansion is a representation of a random variable Y with probability density p_Y as a vector α in this Hilbert space, such that:

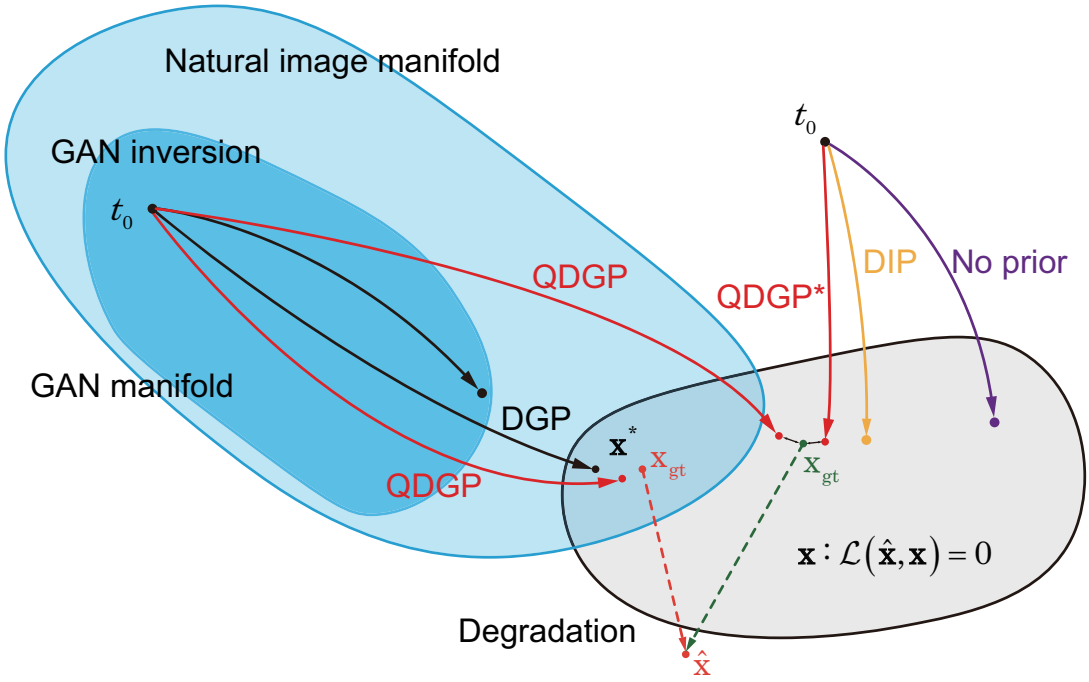
$$Y = \sum_{t=0}^{\infty} \alpha_l \phi_l(X) \sim p_Y, \quad X \sim p_X. \tag{6}$$

This definition establishes a Hilbert space as a potential structure for studying mappings of random variables. Specifically, one of the primary outcomes of polynomial chaos expansion is their status as universal generative models. Utilizing data re-uploading quantum circuits with an integer-valued spectrum denoted as $U_\theta(x)$, PQCs can also be interpreted as a GPCE. Consequently, we have the following theorem:

Theorem 1. *Any expectation value sampling model using a quantum re-uploading model with integer-valued spectrum $U_\theta(x)$, together with the uniform distribution on $[0, 2\pi)^M$ outputs a random variable Y that has an exact finite polynomial chaos expansion for any choice of observables.*

Our proposed PQC also adopts a data re-uploading encoding structure with an integer-valued spectrum, thus it can also be viewed as a GPCE, making it a universal generative model. In contrast, the latent space of the DGP comprises 120 independent

trainable parameters initialized or sampled from a standard Gaussian distribution. During training, these 120 parameters are independently optimized, potentially leading to a mismatch with the pretrained manifold. Therefore, in practice, one may add an L_2 constraint to regularize the latent code to maintain a specific distribution. However, PQC introduces a novel approach to distribution generation or learning. Since PQC is universal for generating multivariate continuous functions, its generated latent code naturally adheres to an intrinsic constraint of parametric distribution. Thus, there is no need to add any constraint to the latent code. The multivariate distribution can be trained or optimized to match the manifold of the pretrained or randomly initialized BigGAN. From this perspective, the PQC-induced distribution can be seen as a generalized distribution compared to the distribution induced by 120 independent parameters. Therefore, we can confidently conclude that PQC enhances the expressiveness of the latent space model without significantly altering the manifold of BigGAN. Moreover, the feature of PQC enables more universal distribution learning or modeling. On one hand, using PQC as the latent code generator may enhance the generalization ability of the generative model. On the other hand, a PQC-based sampling model can be combined with classical generative models to learn the data distribution.



Supplementary Figure 12. Image restoration with priors in the image manifold visualization. We illustrate the image restoration process with different image priors when reconstructing an image $\mathbf{x}_{gt}$ from a degraded image $\hat{\mathbf{x}}$. The gray region belongs to a manifold with zero distance to $\mathbf{x} : \mathcal{L}(\hat{\mathbf{x}}, \mathbf{x}) = 0$. For natural images, the ground-truth both belongs to the natural image and the zero-distance manifold ($\mathbf{x}_{gt}$ in red). Pretrained QDGP has potential to obtain the competitive or better recovery compared to DGP. For non-natural images ($\mathbf{x}_{gt}$ in green), QDGP is likely to obtain the optimal reconstruction when using quantum latent space. GAN inversion generally cannot reach the zero distance manifold. DIP and methods without priors cannot reach the natural image manifold. We note that QDGP with random initialization (QDGP*) has potential to outperform DIP method (DGP with fixed latent space and random initialization).

The illustration of different image priors with classical or quantum latent space is shown in Supplementary Figure 12. QDGP can be view as an integration of both internal learning and external learning where internal learning uses the prior structural information of GAN generator such as DIP and external learning uses the fixed pretrained GAN inversion to recover images. In this work, we study three functions of QDGP shown in origin lines of Supplementary Figure 12. Through the simulation results, we find that quantum latent space can extend the boundaries of GAN manifold better than classical Gaussian code. For non-natural image restoration, QDGP is able to breakthrough the boundary of natural image manifold using PQC generated latent code. The fact can benefit for optimizing non-natural images such as biological samples. PQC enhanced quantum latent space can increase the out-of-distribution generalization of pretrained models and the diversity of latent space to prevent mode collapse.

SUPPLEMENTARY NOTE 6: HYPERPARAMTERS

The total number of parameters in the BigGAN generator is 82,108,355. For random initialization, we train the generator from scratch using the physical forward loss. The prior is inherent in the structure of the generative model, as the generator is naturally

aligned with the image manifold. In the case of a pretrained generator, the model parameters encode information from a large dataset. Fine-tuning the model weights based on feature loss and MSE loss can be beneficial for specific tasks. Both data prior and model prior are utilized.

In both tasks, when using the pretrained model, fine-tuning of the model weights of the classical generator is necessary. We employ BigGAN with a 256^2 resolution, where the discriminator consists of 7 blocks. The output features of each block are utilized as the discriminator loss (feature loss). Mathematically, the feature loss function is expressed as:

$$\mathcal{L}(\mathbf{x}_1, \mathbf{x}_2) = \sum_{i \in \mathcal{I}} \|D(\mathbf{x}_1, i), D(\mathbf{x}_2, i)\|_1 \,, \tag{7}$$

where $\mathbf{x}_1, \mathbf{x}_2$ are two images, corresponding to $\hat{\mathbf{x}}$ and $\Phi(G(\mathbf{M}(\boldsymbol{\theta}), \boldsymbol{\xi}))$ as shown in the main text. D denotes the pretrained discriminator coupled with the generator. $D(\mathbf{x}, i)$ returns the output feature of $\mathbf{x}$ at ith block of D, and $\mathcal{I}$ is the index set of used blocks. Additionally, we employ the progressive reconstruction technique for image recovery. Traditionally, generator parameters are fine-tuned simultaneously during optimization. However, in this work, to mitigate the issue of "information lingering," we utilize progressive reconstruction. Specifically, we initially fine-tune the shallowest block of the generator and gradually proceed with blocks at deeper depths. This approach allows QDGP and DGP to focus first on high-level or global features before gradually incorporating low-level details. Reference [14] has demonstrated that progressive reconstruction better preserves consistency between missing and existing semantics compared to non-progressive reconstruction.

For the ghost image task, the hyperparamters are presented in Supplementary Table IV. Notably, we do not utilize the feature loss as an optimization constraint; instead, we rely on the physical forward model-induced loss function as the primary constraint. Additionally, we incorporate total variation as an extra constraint. The reconstruction process is divided into 5 stages, each with different learning rates for the latent code and the generator. In the case of QDGP, the learning rate for the parameterized quantum circuit (PQC) differs from that of the learnable Gaussian prior. However, the generator learning rate remains consistent for both quantum and classical DGP. It's worth noting that we opt for a larger learning rate in the early stages of optimization for QDGP due to the increased freedom of PQC in generating the latent code. Furthermore, the pretrained generator parameters are tailored to Gaussian distributions, and a larger learning rate may expedite the matching process of the quantum latent code with the pretrained generator.

Supplementary Table IV. The hyperparamters setting for ghost imaging. *Block number* denote the number of generator blocks to be fine tuned. *Iterations* denote that the total number of optimization iterations in each stage. *G lr* and *z lr* denote the learning rate of the generator and latent code, respectively. *PQC lr* denotes the learning rate of PQC generated latent code.

Stage	1	2	3	4	5
Block number	7	7	7	7	7
Iterations	1000	1000	1000	1000	1000
G lr	1.4e-5	1.4e-5	1.4e-5	0.7e-5	0.35e-5
z lr	7e-4	7e-5	7e-5	7e-6	7e-7
PQC lr	7e-3	7e-4	7e-5	7e-6	7e-7

The total number of quantum parameters is determined by the number of layers L, the number of qubits D and the structure. In Supplementary Figure 1, the total number of quantum parameter of the hardware-efficient PQC is $3NL+(N-1)L = 4NL-L$. For **CNOT** entangling gate, the total number of parameters is $3NL$. Each ResBlock requires one PQC to generate the latent code. Therefore, the final number of quantum parameters is $O(NLk)$ with k denoting the number of ResBlock. In fact, we have $Nk = d$, where d is the dimension of latent code. We set $L = 4, N = 20, k = 6$ to construct the quantum circuits to generate the quantum latent code.

For image restoration and manipulation tasks, the hyperparameters setting is presented in Supplementary Table V. For the four typical tasks, most hyperparameters remain consistent between DGP and QDGP, except for the learning rate. Due to the distinct features of the PQC-generated latent space compared to a Gaussian latent code (which has only one trainable parameter for variance), the learning rate also needs to vary. Here, the learning rate steps differ from those in Supplementary Table IV, where the learning rate pertains to the i-th coefficient of the gradients. *lrs* denotes the learning rate before multiplying the learning rate ratio for each stage *lrs* represents the learning rate before multiplying the learning rate ratio for each stage. For ith generator block, the learning rate is

$$lr = lrs \times lr\_ratio(i)^i. \tag{8}$$

The default warm-up steps are set to 0. For super-resolution, the weights of the MSE loss and feature loss are configured to ensure that there is no bias towards either of them. In reality, one can set different weights for MSE and feature loss. For large MSE weights, the recovered image has large PSNR metric. Otherwise, the recovered image has NIQE metric.

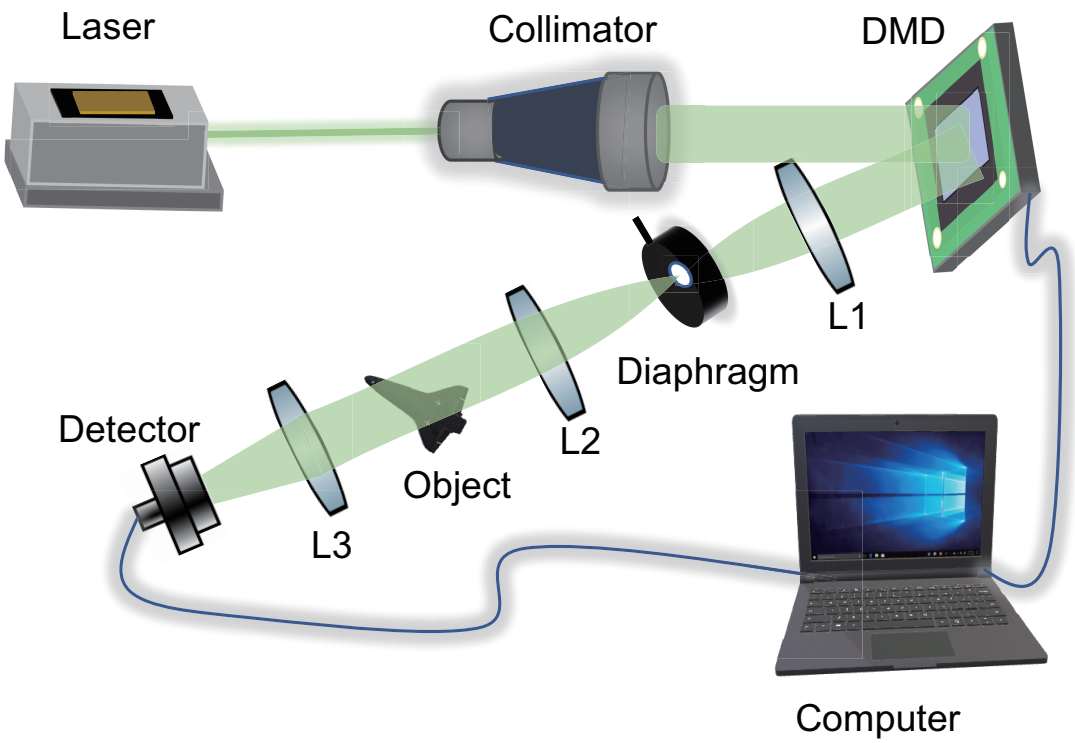
Supplementary Table V. The hyperparamters setting for image restoration and manipulation. *D loss weight* denotes the weight of the feature loss. *MSE loss weight* denotes the weight of the MSE loss. *ftr num* denotes the number of blocks for calculating feature loss. *ft num* is the number of parameter groups in the generator to be fine tuned. *lr ratio* is the decreasing ratio for learning rate in blocks. *G lrs* is the learning rate steps of the generator. *z lrs* and *PQC lrs* denote the learning rate steps of the classical and quantum latent code, respectively.

Stage (Inpainting)	1	2	3	4	5
D loss weight	1	1	1	1	0.5
MSE loss weight	1	1	1	1	10
ftr num	8	8	8	8	8
ft num	7	7	7	7	7
Iterations	200	200	200	200	200
lr ratio	1.0	1.0	1.0	1.0	1.0
G lrs	5e-5	5e-5	2e-5	2e-5	1e-5
z lrs	2e-3	1e-3	2e-5	2e-5	1e-5
PQC lrs	5e-2	5e-3	1e-3	5e-4	1e-4
Stage (Colorization)	1	2	3	4	5
D loss weight	1	1	1	1	1
MSE loss weight	0	0	0	0	0
ftr num	7	7	7	7	7
ft num	2	3	4	5	6
Iterations	200	200	300	400	300
lr ratio	0.7	0.7	0.8	0.9	1.0
G lrs	5e-5	5e-5	5e-5	5e-5	2e-5
z lrs	2e-3	1e-3	5e-4	5e-5	2e-5
PQC lrs	2e-2	1e-2	5e-3	1e-3	1e-4
Stage (Super-resolution)	1	2	3	4	5
D loss weight	1	1	1	1	0.5
MSE loss weight	1	1	1	1	10
ftr num	8	8	8	8	8
ft num	7	7	7	7	7
Iterations	200	200	200	200	200
lr ratio	1.0	1.0	1.0	1.0	1.0
G lrs	5e-5	5e-5	2e-5	2e-5	1e-5
z lrs	2e-3	1e-3	2e-5	2e-5	1e-5
PQC lrs	1e-2	1e-3	2e-5	2e-4	1e-4
Stage (Category transfer)	1	2	3	/	/
D loss weight	1	1	1	/	/
MSE loss weight	0	0	0	/	/
ftr num	8	8	8	/	/
ft num	7	7	7	/	/
Iterations	125	125	100	/	/
lr ratio	1	1	1	/	/
G lrs	2e-7	2e-5	2e-6	/	/
z lrs	1e-1	1e-2	2e-4	/	/
PQC lrs	1e-1	1e-1	2e-2	/	/

SUPPLEMENTARY NOTE 7: GHOST IMAGING EXPERIMENTAL SETUP

We employ a conventional single-beam computational ghost imaging setup to acquire data, illustrated in Supplementary Figure 13. The procedure commences with a pulsed laser beam (SC-OEM, YSL) passing through a fiber and a filter (AOTF, YSL), which narrows the wavelength to $\lambda = 680$ nm. The monochromatic light, expanded by a collimator, illuminates the Digital Micromirror Device (DMD) (DLC9500P24, 1080×1920) with a sequence of M patterns. The reflected light from the DMD undergoes spatial modulation and travels to the object through a 4f system (L1 and L2) and a diaphragm.

Following interaction with the object, the light is gathered by lens L3, detected by a Single Photon Avalanche Diode (SPAD), and directed to the photon counter (HydraHarp 400) to generate bucket signals. The focal lengths of lenses L1, L2, and L3 are all configured to 100 mm. Post data acquisition, the bucket signals, comprising M measurements, together with the corresponding



Supplementary Figure 13. The experimental setup of the optical ghost imaging system. The light from the laser is modulated by the DMD and then the modulated light interacts with the object and finally detected by a single pixel camera. The computer controls the pattern of the DMD and analysis the results of the detection signals.

illumination patterns, are utilized for image reconstruction.

[1] Junyu Liu, Khadijeh Najafi, Kunal Sharma, Francesco Tacchino, Liang Jiang, and Antonio Mezzacapo. Analytic theory for the dynamics of wide quantum neural networks. *Physical Review Letters*, 130(15):150601, 2023.

[2] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, 4(4):043001, 2019.

[3] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *nature*, 549(7671):242–246, 2017.

[4] Christa Zoufal, Aurélien Lucchi, and Stefan Woerner. Variational quantum boltzmann machines. *Quantum Machine Intelligence*, 3:1–15, 2021.

[5] Chenfeng Cao and Xin Wang. Noise-assisted quantum autoencoder. *Physical Review Applied*, 15(5):054012, 2021.

[6] Jonathan Romero, Jonathan P Olson, and Alan Aspuru-Guzik. Quantum autoencoders for efficient compression of quantum data. *Quantum Science and Technology*, 2(4):045001, 2017.

[7] Samuel Yen-Chi Chen, Chao-Han Huck Yang, Jun Qi, Pin-Yu Chen, Xiaoli Ma, and Hsi-Sheng Goan. Variational quantum circuits for deep reinforcement learning. *IEEE Access*, 8:141007–141024, 2020.

[8] Tailong Xiao, Jingzheng Huang, Hongjing Li, Jianping Fan, and Guihua Zeng. Quantum generative adversarial imitation learning. *New Journal of Physics*, 25(3):033034, 2023.

[9] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[10] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal processing letters*, 20(3):209–212, 2012.

[11] Lucas Friedrich and Jonas Maziero. Avoiding barren plateaus with classical deep neural networks. *Physical Review A*, 106(4):042433, 2022.

[12] Ali Rad, Alireza Seif, and Norbert M Linke. Surviving the barren plateau in variational quantum circuits with bayesian learning initialization. *arXiv preprint arXiv:2203.02464*, 2022.

[13] Alice Barthe, Michele Grossi, Sofia Vallecorsa, Jordi Tura, and Vedran Dunjko. Expressivity of parameterized quantum circuits for generative modeling of continuous multivariate distributions. *arXiv preprint arXiv:2402.09848*, 2024.

[14] Xingang Pan, Xiaohang Zhan, Bo Dai, Dahua Lin, Chen Change Loy, and Ping Luo. Exploiting deep generative prior for versatile image restoration and manipulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7474–7489, 2021.