KI - Künstliche Intelligenz
https://doi.org/10.1007/s13218-024-00858-5

TECHNICAL CONTRIBUTION

Quantum Hybrid Diffusion Models for Image Synthesis

Francesca De Falco[1] · Andrea Ceschini[1] · Alessandro Sebastianelli[2] · Bertrand Le Saux[2] · Massimo Panella[1]

Received: 29 February 2024 / Accepted: 2 July 2024

Abstract
In this paper, we propose a new methodology to design quantum hybrid diffusion models, derived from classical U-Nets with ResNet and Attention layers. Specifically, we propose two possible different hybridization schemes combining quantum computing's superior generalization with classical networks' modularity. In the first one, we acted at the vertex: ResNet convolutional layers are gradually replaced with variational circuits to create Quantum ResNet blocks. In the second proposed architecture, we extend the hybridization to the intermediate level of the encoder, due to its higher sensitivity in the feature extraction process. In order to conduct an in-depth analysis of the potential advantages stemming from the integration of quantum layers, images generated by quantum hybrid diffusion models are compared to those generated by classical models, and evaluated in terms of several quantitative metrics. The results demonstrate an advantage in using hybrid quantum diffusion models, as they generally synthesize better-quality images and converges faster. Moreover, they show the additional advantage of having a lower number of parameters to train compared to the classical one, with a reduction that depends on the extent to which the vertex is hybridized.

Keywords Quantum hybrid diffusion model · Variational quantum circuit · Quantum hybrid U-Net · Efficient quantum simulation.

Mathematics Subject Classification MSC 68 · MSC 81 · MSC 94

1 Introduction

Quantum Machine Learning (QML) has recently emerged as a promising framework for generative Artificial Intelligence (AI). In fact, over the years, QML algorithms have been developed for both supervised [1–3] and unsupervised [4, 5] learning tasks. The implementation and performance analysis of these machine learning algorithms have demonstrated that quantum computing can bring numerous advantages. In particular, the benefits come from the exponentially large space that a quantum system can express, as well as from the ability to represent mappings that are classically impossible to compute [6, 7]. Additionally, an investigation into effective dimension is presented in [8] by a data-dependent capacity measure, revealing that quantum computing can offer benefits by achieving a better effective dimension than comparable classical neural networks.

In classical AI, Diffusion Models (DMs) have established themselves as the leading candidates for data and image generation [9–11], showcasing superior quality and stability in training when compared to state-of-the-art Generative Adversarial Networks (GANs) [12]. DMs rely on an iterative diffusion process that effectively models complex distributions by progressively refining the data distribution through a sequence of diffusion steps. However, DMs may encounter notable challenges, including high computational requirements and the need for extensive parameter adjustments [13].

Considering generative QML, there have been various implementations of Quantum Generative Adversarial Networks (QGANs) that have demonstrated superior performance in capturing the underlying data distribution. They have also shown better generalization properties, allowing for a significantly lower number of trainable parameters compared to classical GANs [14–17]. One of the first

✉ Massimo Panella
massimo.panella@uniroma1.it

[1] Department of Information Engineering, Electronics and Telecommunications, University of Rome "La Sapienza", 00184 Rome, Italy

[2] ESA Φ-Lab, European Space Agency, 00044 Frascati, Italy

Published online: 09 August 2024

implementations of a QGAN is described in [14]. In this implementation, only the generator is realized using quantum circuits, offering two different solutions: the first one, called Quantum Patch GAN, involves dividing an image into sub-images, which are then generated by sub-generators and later recombined to form the complete image, whereas in the second one, called Quantum Batch GAN, there is no longer a division into sub-images.

Another possible implementation of QGANs is proposed in [15], where a novel architecture called style-qGAN is presented. In this scenario, the Generator is entirely quantum whereas the Discriminator remains entirely classical. The novelty of this model consists in the embedding of the latent variable into every quantum gate of the network, whereas previous qGAN models introduced them only at the beginning of the network. The results obtained by testing the architecture on toy data, namely 1D gamma and 3D correlated Gaussian distributions, as well as on data for realistic quantum processes at the Large Hadron Collider (LHC), demonstrate the effectiveness of such an architecture and its potential use for data augmentation, as it is capable of reproducing known reference distributions from small sample sets.

Concerning DMs, there have been only some initial and simple attempts to develop a quantum version of a DM. While in [18] a sole theoretical discussion of a potential quantum generalization of diffusion models is provided, with results tied only to very basic and simplified scenarios, two different architectures are proposed in [19]: the first one works on downsized images from the MNIST dataset; the second model suggests operating on a latent space using a pretrained autoencoder. Both architectures use amplitude encoding to load data into quantum circuits, which is a memory-efficient approach suitable for current quantum devices, but requires an exponential number of circuit runs to fully reconstruct the entire output distribution of bitstrings.

In this paper, we propose an efficient methodology to design Hybrid Quantum Diffusion Models (HQDMs) by incorporating variational quantum layers, with novel circuit designs within a classical U-Net [20]. Namely, we employ a state-of-the-art U-Net architecture, proposed in [11] and composed of ResNet [21] and Attention blocks as the foundation for our approach. Therefore, the U-Net is hybridized in two different modes: the first involves inserting Variational Quantum Circuits (VQCs) only at the vertex, while the second implies the insertion of VQCs on both the encoder side, which is more sensitive to feature extraction, and the vertex, which is responsible for compressed image processing. The rationale behind this approach is to leverage the strengths of both classical and quantum computing paradigms. By integrating variational quantum layers into the classical U-Net architecture, we aim to exploit the expressive power of quantum circuits for faster network convergence and superior generalization capabilities [8]. On the other hand, classical U-Net layers allow us to introduce modularity and nonlinearity in the computation, thus enabling complex image processing while mitigating the inherent challenges associated with quantum computing [22, 23]. Furthermore, by strategically placing VQCs at key points within the U-Net architecture, we ensure that quantum layers are properly allocated to areas where they can have the most significant impact on performance improvement, thanks to their expressivity and feature extracting capabilities [24–26].

We also propose to adopt an approach inspired by transfer learning, aiming at an overall efficient training time of the models. During the initial epochs, a classical model is trained and then, some of its weights are transferred to a hybrid model that is trained, in turn, for some further epochs. In this way, while still maintaining a limited training time, we achieve better results compared to the classical counterpart: on Fashion MNIST, we achieve + 2% FID, + 5% KID, and + 2% IS. On MNIST, the results are even better with an improvement in metrics of approximately + 8% on FID, + 11% on KID, and + 2% on IS.

Our innovative architecture seamlessly incorporates quantum elements through the employment of angle encoding as the primary encoding method, whereas the outputs of the circuits are retrieved through the expected value of the Pauli-Z observable. Since the VQCs are placed at specific points in the U-Net where the image dimensions are reduced, and thus the number of pixels to be encoded in the circuit through angle encoding is low, we achieve a crucial result: having a low number of qubits, which optimizes resource utilization and ensures practicality when deploying our model on Noisy Intermediate-Scale Quantum (NISQ) devices. Moreover, unlike amplitude encoding, angle encoding has the advantage of streamlining output computation without the need for an exponentially long circuit, which is unfeasible for current NISQ machines.

Additionally, the analysis of two different hybrid architectures allows to highlight how certain parts of the U-Net are more sensitive to the quantum integration than others. Not only do these enhancements lead to remarkable improvements across various performance metrics, but they also yield a substantial reduction in the number of parameters requiring training. In particular, depending on the degree of U-Net hybridization, we can achieve up to about an 11% reduction in parameters compared to the classical one. Furthermore, it is possible to achieve an improvement of almost 2% on the FID and KID metrics in the case of the Fashion MNIST dataset, while in the case of the MNIST dataset, we achieve an improvement of about 5% on the FID and more than 6% on the KID compared to the classical network.This reduction not only streamlines the overall training process, but also contributes to more efficient utilization of computational resources,

ultimately enhancing the model's scalability and applicability in real-world scenarios.

The rest of the paper is organized as follows. In Sect. 2 we provide an explanation of DMs and variational circuits. In Sect. 3 we present the employed methodology, while in Sect. 4 we discuss the obtained results. Finally, we draw our conclusions in Sect. 5.

2 Theoretical Background

2.1 Classical Diffusion Models

In recent years, DMs have proven to be an important class of generative models. A standard mathematical formulation for diffusion models is the one presented by [10] and here summarized to give to readers a general overview of its fundamentals.

DMs mainly consist of two distinct phases as shown in Fig. 1. The first one is the forward process, also called diffusion, involving a transformation that gradually converts the original data distribution $\mathbf{x}_0 \sim q$, where q is a probability distribution to be learned, by repeatedly adding Gaussian noise:

$$\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon}_t, \;\; t = 1 \dots T, \tag{1}$$

where $\boldsymbol{\epsilon}_1, \dots, \boldsymbol{\epsilon}_T$ are IID samples drawn from a zero-mean, unit variance Gaussian (normal) distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, and β_t determines the variance scale for the t-th step. This progression is underpinned by a Markov chain that can be represented as follows:

$$q(\mathbf{x}_{0:T}) = q(\mathbf{x}_0) \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \tag{2}$$

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}\Big(\sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}\Big), \tag{3}$$

being $\mathbf{I}$ the identity matrix.

The goal of the forward process is to add incremental noise to the initial sample $\mathbf{x}_0$ over a certain number of steps, until at the final time step T all traces of the original distribution $\mathbf{x}_0 \sim q$ are lost so as to obtain $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Through the application of the 'reparameterization trick', a closed-form solution becomes available for calculating the total noise at any desired step using the cumulative product:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, \tag{4}$$

where $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the Gaussian noise.

The second phase is the reverse process or backward diffusion, where the transformations gradually restore the initial noise distribution and reconstruct a noise-free version of the original data. If we could successfully reverse the aforementioned process sampling from $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$, we would gain the ability to recreate the true sample starting from the Gaussian noise input $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; it is also noteworthy that when β_t is sufficiently small, $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is close to a Gaussian distribution. Regrettably, estimating $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is complex due to its reliance on the entire dataset and hence, a data-driven learning model like a neural network must be

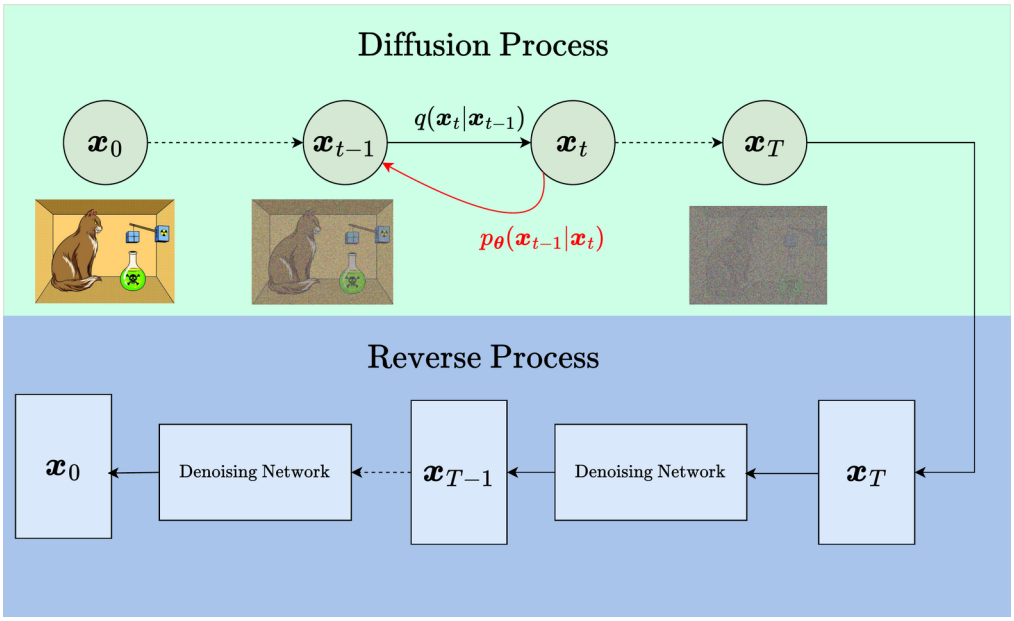


Fig. 1 Diffusion process and reverse process of a DM

used in order to approximate these conditional probabilities, enabling the execution of the reverse diffusion process.

Let p_θ be the mathematical model depending on some parameters $\boldsymbol{\theta}$ that represents the estimated distribution of the backward diffusion process:

$$p_\theta(\mathbf{x}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{5}$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}\big(\boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\big), \tag{6}$$

where $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$ and $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)$ are the general outputs of the adopted neural network, which takes as inputs $\mathbf{x}_t$ and t.

A simplified approach based on variational inference assumes a fixed covariance matrix, such as for instance $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t) = \beta_t \mathbf{I}$, and the direct estimation by the neural network of the noise $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ at time step t. Then, using reparameterization and the normal distributions of conditional data, we obtain:

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right). \tag{7}$$

The neural network producing $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ is usually trained by stochastic gradient descent on an even more simplified loss function like:

$$L_t^{\text{simple}} = \mathbb{E}_{\mathbf{x}_0 \sim q, t, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})}\left[\left\|\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) - \boldsymbol{\epsilon}\right\|^2\right]. \tag{8}$$

2.2 Variational Quantum Circuits

Variational Quantum Algorithms (VQAs) are the most common QML algorithm that are currently implemented on today's quantum computers [27], they make use of parametrized quantum circuits known as ansatzes. Ansatz circuits are composed of quantum gates that manipulate qubits through specific parametrized unitary operations. However, these operations depend on parameters denoted as $\boldsymbol{\theta}$, which are the parameters to be trained during the training process.

The training workflow of a VQC, shown in Fig. 2, can be summarized as follows:

1. Classical data $\boldsymbol{x} \in \mathbb{R}^n$ is appropriately encoded in a quantum state of the Hilbert space $\mathbb{H}^{2^n}$ through the unitary $U_\phi(\boldsymbol{x})$, to be used by the quantum computer;
2. An ansatz $U_W(\boldsymbol{\theta})$ of $\boldsymbol{\theta}$-parametrized unitaries with randomly initialized parameters and fixed entangling gates is applied to the quantum state $|\phi(\boldsymbol{x})\rangle$ obtained after the encoding;
3. Upon completion, measurements are taken to obtain the desired outcomes. The expected value with respect to a given observable $\hat{O}$ is typically computed, and the resulting prediction is given by:

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = \langle \phi(\boldsymbol{x})| U_W(\boldsymbol{\theta})^\dagger \hat{O} U_W(\boldsymbol{\theta}) |\phi(\boldsymbol{x})\rangle, \tag{9}$$

4. Finally, a suitable loss function is evaluated, and a classical co-processor is used to properly update the parameters $\boldsymbol{\theta}$.

This cycle is repeated until a termination condition is met. To update $\boldsymbol{\theta}$ and train the VQC, gradient-based techniques can be used; gradients in a parametrized quantum circuit are calculated via the parameter-shift rule:

$$\nabla_\theta f(\boldsymbol{x}, \theta) = \frac{1}{2}\left[f(\boldsymbol{x}, \theta + \frac{\pi}{2}) - f(\boldsymbol{x}, \theta - \frac{\pi}{2})\right], \tag{10}$$

where $f(\boldsymbol{x}, \theta)$ is the output of the quantum circuit and θ is the parameter to be optimized.

3 Proposed Methodology

In the following, we introduce the quantum hybrid architectures proposed in this paper.

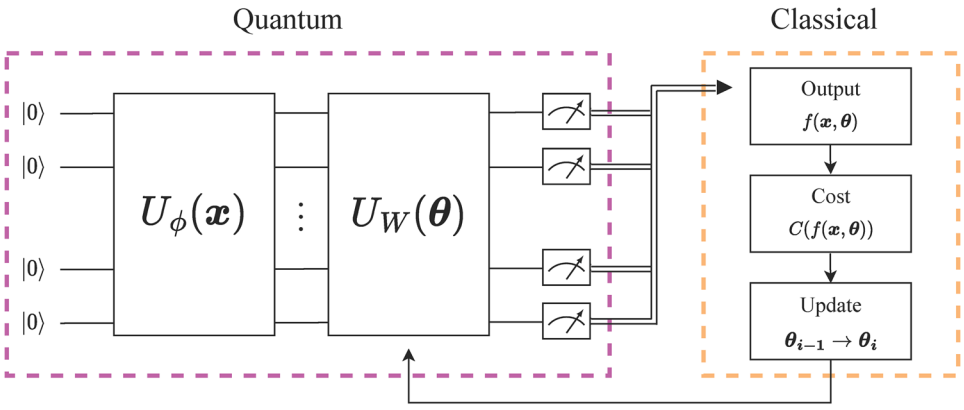


Fig. 2 Scheme of a hybrid quantum-classical VQC

3.1 Quantum Vertex U-Net Hybrid Architecture

Starting from the classical U-Net architecture, which is an extremely parameter-rich and expressive model, we propose a hybrid architecture capable of harnessing quantum capabilities to achieve better performance and, furthermore, to reduce the total number of parameters to be trained. The classical U-Net is initially implemented according to the proposal by [28], aiming at working with a state-of-the-art model: it is composed of ResNet blocks for residual connections and Attention blocks for feature aggregation; specifically, we use Multi-head Attention with four attention heads, as suggested in [28]. The ResNet and Attention layers are hence applied at various resolution levels in the U-Net. The first hybrid architecture we propose, which we name Quantum Vertex U-Net (QVU-Net), uses this U-Net as its reference architecture and efficiently integrates quantum layers within its structure, as shown in Fig. 3.

We use angle encoding as the data encoding method in order to achieve an architecture that efficiently incorporates quantum layers. This approach avoids the inefficiencies associated with amplitude encoding, which requires exponentially long circuits, and an exponential number of circuit runs to generate statistically valid outputs from the quantum states distribution with the same dimensionality as the inputs. In our angle encoding, input data $\boldsymbol{x} \in \mathbb{R}^n$ is encoded into n qubits via unitary transformation $U_{\phi(\boldsymbol{x})}$ made up of R_x rotation gates; each qubit encodes one feature of our input data, resulting in $\mathcal{O}(1)$ circuit depth.

The underlying idea in our hybrid model is to integrate the quantum components strategically, in such a way that the number of qubits used is kept limited while ensuring streamlined data processing efficiency. Specifically, considering that the classical U-Net initially processes a $28 \times 28 \times 1$ image and progressively scales it within the network's encoder until reaching a vertex with dimensions of $2 \times 2 \times 40$, we introduce the quantum elements precisely at the vertex of the network.

We propose the Quantum ResNet (QResNet) layer at the vertex of the QVU-Net, as shown in Fig. 4. The QResNet is analogous to the ResNet, as it is characterized by skip connections and two processing layers, whose output is then added to the input; the difference with the classical ResNet layer is that some of the Convolutional layers used in the classical ResNet are replaced with VQCs. As we are working with images scaled down to 2×2 dimensions, the only viable choice is to use a VQC instead of convolutional layers, which effectively amounts to a single filter pass over the entire image. Not all Convolutional layers of the ResNet are replaced with VQCs; rather, the replacement is done gradually. We initially analyze a hybrid architecture where the percentage of channels processed by VQCs in QResNet is set at 10%. Subsequently, we examine an architecture in which 50% of the input channels in QResNet are processed by VQCs. Finally, we explore an architecture in which all 40 channels are processed solely by VQCs instead of Convolutional layers. Gradually hybridizing the architecture allows us to better and more comprehensively analyze the impact

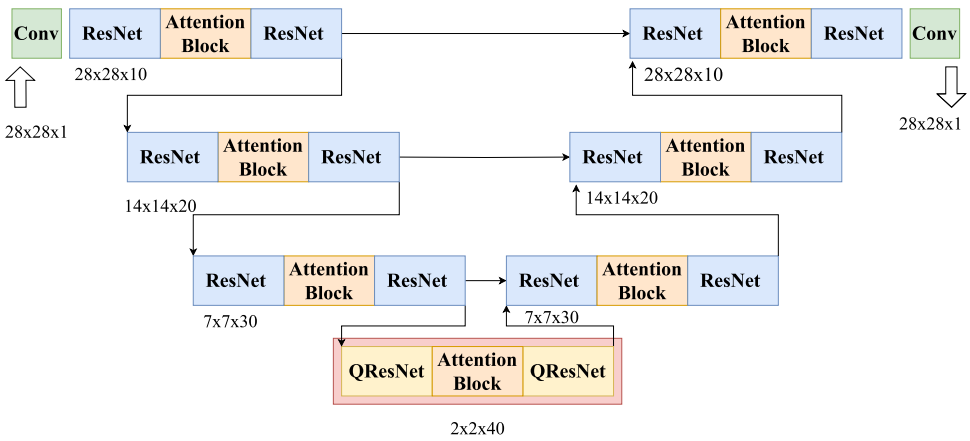


Fig. 3 First proposed hybrid U-Net architecture named QVU-Net, where the quantum part is incorporated at the vertex of the network

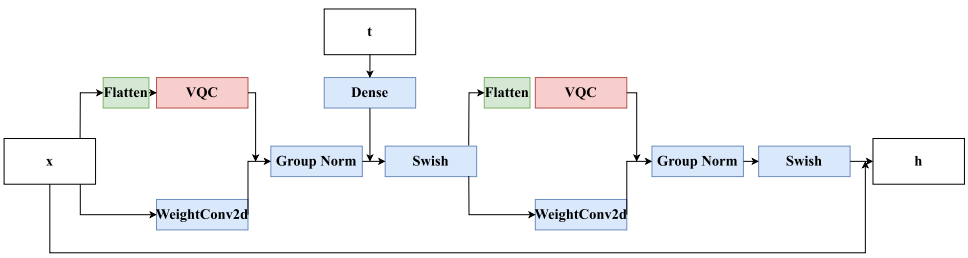


Fig. 4 Architecture of the QResNet block, where Convolutional layers are replaced with VQCs. The QResNet takes as input x, which is the information coming from the image, and t, which is the temporal information, and finally returns h

of incorporating quantum elements into the network's structure. To the best of our knowledge, this is the first time a quantum version of the ResNet block within a hybrid U-Net is proposed in the literature.

The choice of the ansatz in the VQCs was made considering that the maximum channels at the vertex are 40. Since the information in the vertex is distributed across multiple channels rather than confined to one, we want to adopt an ansatz configuration capable of efficiently capturing and leveraging the correlations among these channels. Starting from these considerations, we try two distinct ansatzes structures, inspired by [29]. In fact, their choice is driven by the fact that they work on three channels simultaneously, aiming to process not only local information related to a single channel, but especially intra-channel information.

The Hierarchical Quantum Convolutional Ansatz (HQConv) in Fig. 5 extracts local information separately among the initial channels, followed by additional controlled gates used to encode intra-channel information. Initially, controlled gates are used to extract information within each channel first. In particular, as shown in Fig. 5, the A blocks, acting on qubits belonging to the same channel, can be expressed mathematically as:

$$\left| q_p^2, q_{p+s}^2 \right\rangle = \left[CR_x(\theta_{x,p}) \circ CR_z(\theta_{z,p}) \right] \left| q_p^1, q_{p+s}^1 \right\rangle$$

where symbol $\circ$ represents the composition of the CR_x gate and CR_z gate, with CR_z being implemented first. Additionally, q_p^1 is the control qubit and q_{p+s}^1 is the target qubit, moreover the subscript p indicates the pixel to which the qubit refers and ranges from 0 to 3 as the images has dimensions of 2×2 for each channel. The symbol s represents instead the value of hyperparameter stride, i.e. indicates the distance between the control qubit and the target qubit. The stride used in this case is equal to 1. The superscript 1 indicates the initial state of the qubit immediately after encoding, while the superscript 2 indicates the state of the qubit after the application of the considered block A. The second part of the ansatz is instead characterized by intra-channel information processing. As seen in Fig. 5, the B blocks are aimed at working on qubits belonging to two different channels, and in particular, we can express them in mathematical terms as:

$$\left| q_0^3, q_4^3 \right\rangle = \left(CR_x(\theta_x) \circ CR_z(\theta_z) \right) \left| q_0^2, q_4^2 \right\rangle$$

where q_0^2 is the control qubit, i.e. the first qubit of the first channel considered in the block B and q_4^2 is the target qubit, i.e. the first qubit of the second channel considered in the block B. The subscript 3 in this case indicates the state of the qubit after the application of the block B.

On the other hand, the Flat Quantum Convolutional Ansatz (FQConv) in Fig. 6 immediately incorporates both intra-channel and inter-channel information, giving its structure a flat form. The blocks C and D, as shown in Fig. 6, are characterized by the presence of gates controlled by qubits belonging to another channel. In mathematical terms, the block C can be expressed as:

$$\left| q_p^2, q_{p+s}^2 \right\rangle = \left(CR_z(\theta_{z,p}) \right) \left| q_p^1, q_{p+s}^1 \right\rangle$$

while the block D as:

$$\left| q_p^3, q_{p+s}^3 \right\rangle = \left(CR_x(\theta_{x,p}) \right) \left| q_p^2, q_{p+s}^2 \right\rangle$$

where as before q_p is the control qubit and q_{p+s} is the target qubit. The stride used in this case is equal to 4. Once again,

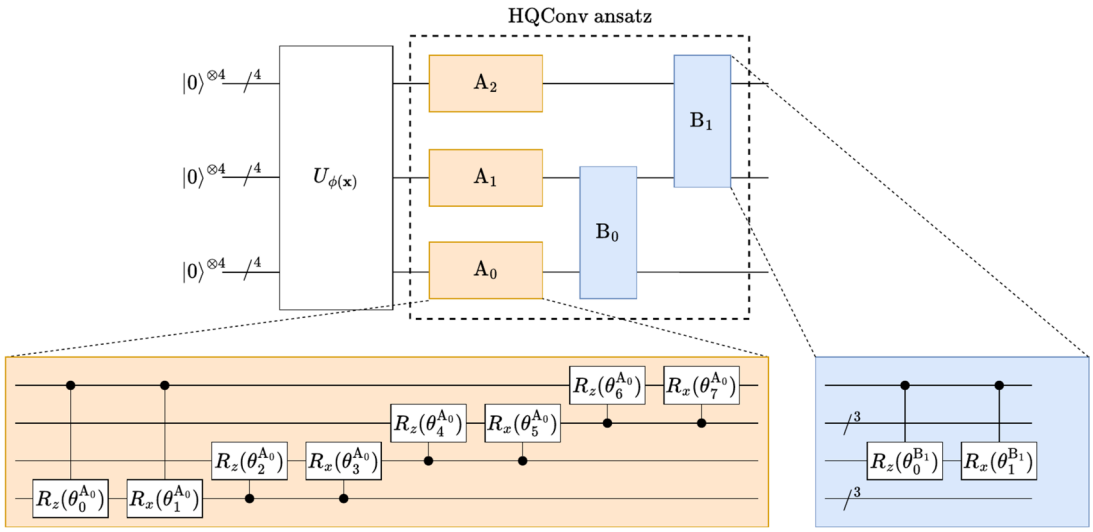


Fig. 5 HQConv ansatz proposed in [29] initially extracts local information, then uses additional controlled gates to encode intra-channel information

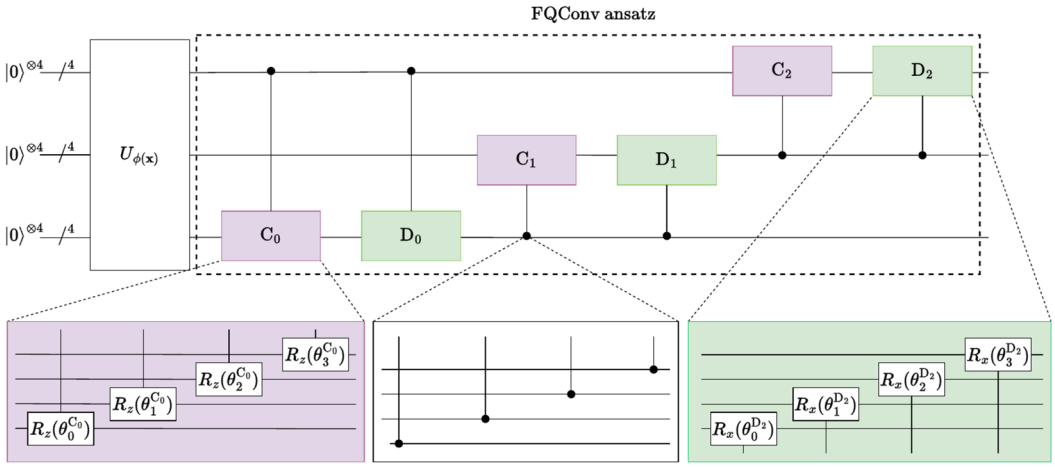


Fig. 6 FQConv ansatz proposed in [29] is capable of immediately incorporating both intra-channel and inter-channel information

the superscript 1 indicates the initial state of the qubit after encoding, 2 after the application of the first block (block C), and 3 the state of the qubit after the application of the second block (block D). These ansatzes operate on 3 channels at a time; since the compressed image at the vertex is 2×2 in size, we need 4 qubits per channel, and hence the total number of qubits used in our VQCs is 12. When considering the architecture where the entire vertex is processed with quantum circuits, as there are a total of 40 channels to be processed, the first 39 are processed with the previously described ansatzes, HQConv or FQConv, while the last one is processed with the PennyLane Basic Entangling Layer, which involves Rx rotations acting on 4 qubits only. The number of layer inside each VQC is equal to 3. At the end of the circuit, a measurement operation carried out in the Pauli-Z basis, which has $+1$ and -1 as eigenvalues, is performed on each qubit to extract the outcome of the quantum circuit as the expected value with respect to the Pauli-Z operator. The output of each VQC is thus restricted to the $[-1, 1]$ range.

3.2 Quanvolutional U-Net Hybrid Architecture

In addition to the QVU-Net, where only the vertex is hybridized, we also propose a hybridization in a part of the U-Net network dedicated to feature extraction. The purpose is to assess whether quantum feature extraction can indeed bring further improvements in terms of the quality of the generated images. However, we decide not to act on the first level of the U-Net because the images still have dimensions of $28 \times 28 \times 10$, making the efficient use of a VQC computationally challenging. Therefore, we consider the second level of the encoder, where the images are $14 \times 14 \times 20$, as shown in Fig. 7; we call this architecture QuanvU-Net.

In order to keep angle encoding, we process the image in the second level of the encoder with an idea inspired by the Quanvolutional method [30]. We employ Quanvolutional filters that, similarly to classical Convolutional filters, process one subsection of the image at a time, until they have traversed the entire image. In doing so, they produce a feature map by transforming spatially-local subsections of the input tensor. Unlike the straightforward element-wise matrix multiplication operation performed by a classical Convolutional filter, a Quanvolutional filter alters input data through the utilization of a quantum circuit, which may have a structured or random configuration. In our case, the Quanvolutional approach is employed within the ResNet block, thereby creating the QuanResNet block, as depicted in Fig. 8. Specifically, we consider only 3 channels out of the total 20 in the 14×14 image to make the approach practically feasible, as it requires a large number of quantum circuits executions. In particular, 4 pixels are taken from each channel and processed by a 12-qubits variational circuit; the variational circuit remains always the same when passing over the entire image and acts as if it was a classical Convolutional filter.

As for the previous QVU-Net architecture, the idea with the QuanvU-Net is to work on 3 channels at a time to process both intra-channel and inter-channel features. Therefore, the ansatzes used in the QuanResNet block are the same as the ones used at the vertex, namely HQConv and FQConv shown in Figs. 5 and 6, respectively. Also in this case, the use of Quanvolutional filters in a ResNet block marks a novel advancement compared to the literature.

3.3 Transfer Learning Approach

In addition to standard training for both the hybrid and classical networks, we propose an approach inspired by transfer learning. In fact, both the training phase and the subsequent

inference phase prove to be significantly time consuming in the case of the hybrid QVU-Net and QuanvU-Net, becoming even longer compared to the classic U-Net as the percentage of variational circuits at the vertex increases. It is therefore worth reducing the time required for the training phase.

For this reason, we propose to adapt classical-to-quantum transfer learning [31], which proves to be one of the most appealing transfer learning approaches. As illustrated in Fig. 9, our idea is to initially train the classic U-Net for a certain number of epochs. The weights obtained in this way are then transferred to the QVU-Net, except for the weights at the vertex. So the QVU-Net is trained for a very limited number of epochs. In this way, it is expected that the final fine-tuning with the hybrid networks can bring a significant improvement in performance compared to using only the classical network, while still maintaining a low training time.

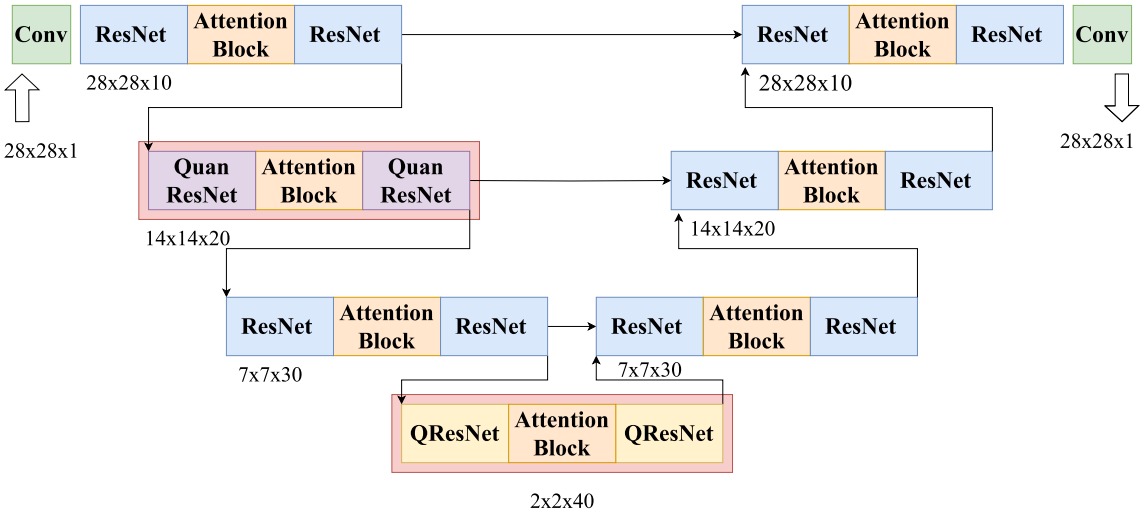


Fig. 7 Second proposed U-Net hybrid architecture named QuanvU-Net, where the quantum part is incorporated not only at the vertex but also at the second level of the encoder block

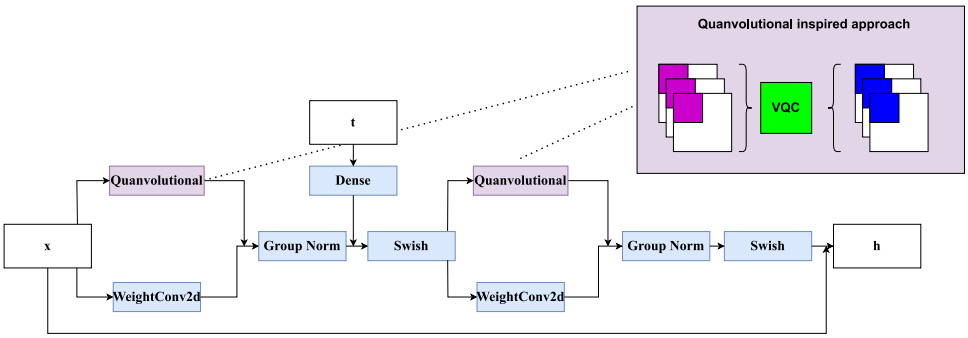


Fig. 8 Architecture of the QuanResNet block, where the Convolutional layer of the classical ResNet is replaced with a Quanvolutional filter. The QuanResNet takes as input **x**, which is the information coming from the image, and **t**, which is the temporal information, and finally returns **h**

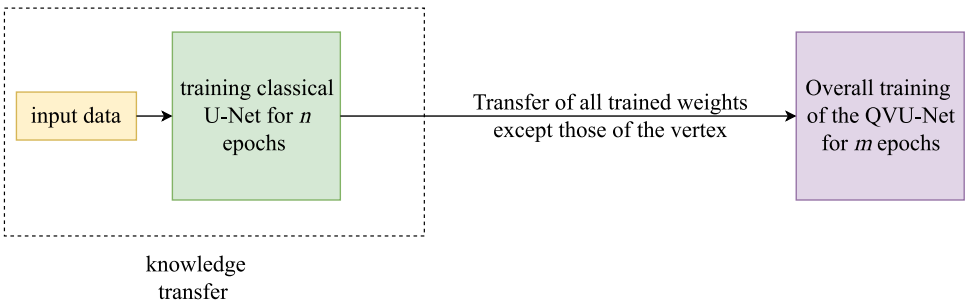


Fig. 9 Outline of the proposed transfer learning approach. The classical model is trained for n epochs. All the weights, except those of the vertex, are then transferred to the hybrid model. The latter is further trained for additional m epochs

4 Experimental Results

In this section, we analyze the results obtained by using the different architectures proposed in the paper:

- the 1HQConv QVU-Net and 1FQConv QVU-Net with quantum circuits in the QResNet that act only on 10% of the channels at the vertex and use the HQConv or FQConv the ansatz, respectively;
- 7HQConv QVU-Net and 7FQConv QVU-Net, with quantum circuits in the QResNet that act only on 50% of the channels at the vertex and use the HQConv or FQConv the ansatz, respectively;
- FullHQConv QVU-Net and FullFQConv QVU-Net, with quantum circuits in the QResNet that act on all channels at the vertex and uses the HQConv or FQConv the ansatz, respectively;
- QuanvU-Net with the vertex ibridized at 10% and the QuanResNet at the second level of the encoder.

4.1 Experimental Settings

The implementation is carried out in Python 3.8 using PennyLane and Flax. PennyLane is a framework enabling local quantum circuits simulations and integration with classical neural networks, whereas Flax is an open-source framework providing a flexible and efficient platform for hybrid neural network execution via compilation. We use PennyLane for the implementation of quantum circuits, while the hybrid networks and the entire training process are carried out in Flax; the classical U-Net is implemented in Flax as well.

Regarding the experiments, the L2 loss is used with the P2 weighting [32]. We used an exponential moving average (EMA) over model parameters with a rate that depends on the training step, and the Adam optimizer [33] is used with a learning rate of 10^{-3}, β_1 of 0.9, and β_2 of 0.99. The training process consists of a total of 20 epochs.

We use two benchmark datasets, namely MNIST [34] and Fashion MNIST [35]. Both of them contain grayscale images belonging to 10 different classes, with a total of 60k training samples. The metrics used for evaluations are FID [17], Kernel Inception Distance (KID) [36], and Inception Score (IS) [37], assessed on 7000 generated images. We utilize the TorchMetrics library [38], replicating each channel of the generated images three times to make the dimensions compatible with those required by InceptionV3 network backbone. For the KID calculation, the subset size for computing mean and variance was set to 100, while for the IS calculation the dataset was divided into 10 splits for mean and variance computation.

A machine equipped with an AMD Ryzen 7™ 5800X 8-Core CPU at 3.80 GHz and with 64 GB of RAM is used for the experiments.

4.2 Fashion MNIST Dataset

Let us initially consider the images generated by Fashion MNIST. As there are no significant differences between the use of HQConv and FQConv, we specifically examine the results obtained by HQConv with three layers. As seen in Fig. 10, at the first epoch hybrid networks demonstrate better performance compared to the classical one, achieving a slightly lower FID and a higher IS value. Indeed, the images generated by the FullHQConv QVU-Net achieve an FID of 295.7863 and an IS of 1.4454 ± 0.0057, while those obtained from the classical architecture have a higher FID of 296.869 and a lower IS of 1.4164 ± 0.0122. This is in line with our expectations, as quantum models with a few epochs are generally more adept at extracting features and processing than classical models. By the tenth epoch, all hybrid networks show significantly better values in terms of FID and KID, with the 1HQConv QVU-Net Architecture in Fig. 10f having a FID of 52.5332, which is more than seven points lower than the classical network in Fig. 10e, which has a FID of 60.1476.Therefore, the 1HQConv QVU-Net Architecture is able to achieve an FID that is nearly 13% better than the classical architecture. The IS of hybrid networks at the tenth epoch is comparable to that achieved by the classical network. However, at the twentieth epoch, a gradual deterioration in the performance of hybrid networks in Figs. 10j–l is observed, progressively worsening with an increase in the level of hybridization. The results are still comparable to those obtained by the classical network, as the classical architecture generates images with an FID of 39.4563, while the worst-performing quantum architecture in this case, the FullHQConv QVU-Net, has an FID of 41.3882. But the most important aspect is that there is a significant reduction in parameters as the percentage of vertex hybridization increases: indeed, the FullHQConv QVU-Net has more than 11% fewer parameters than the classical architecture.

Considering the images generated by the second possible hybridization of the U-Net, the QuanvU-Net architecture, which involves not only the hybridized vertex but also the use of quanvolutional on outer layers of the U-Net, it can be observed in Fig. 11 how this yields a better performance. Considering the case of only 10% of the hybridized vertex, which leads to more satisfactory performance, we outline that if only the vertex is hybridized at the first epoch, the metrics are more or less similar to those of the classical network. However, if we also consider the QuanvU-Net from the first epoch, as shown in Fig. 11c, the performance is greatly improved with a FID lower than that of the classical network by about 10 points, as the images generated by the

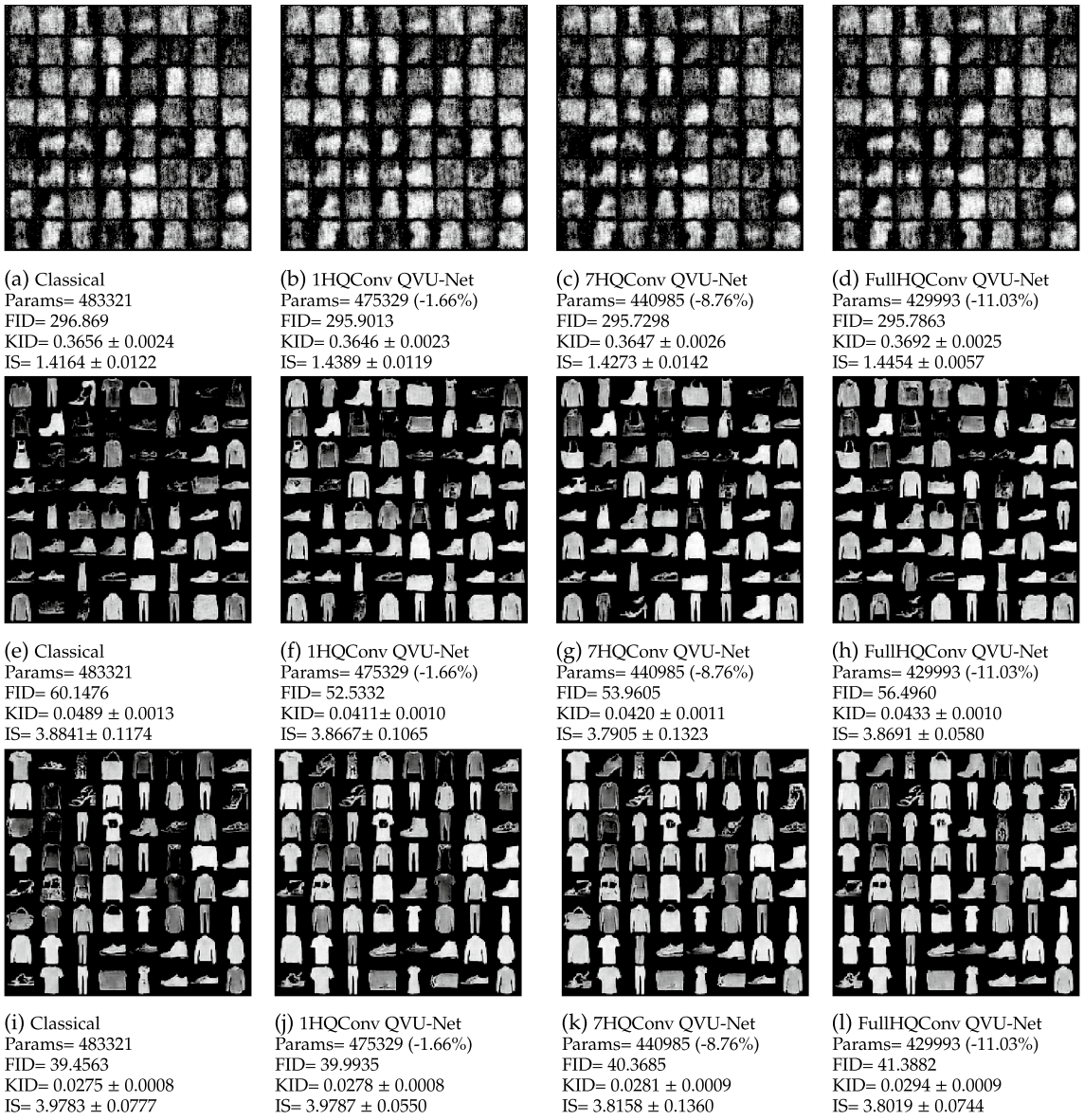
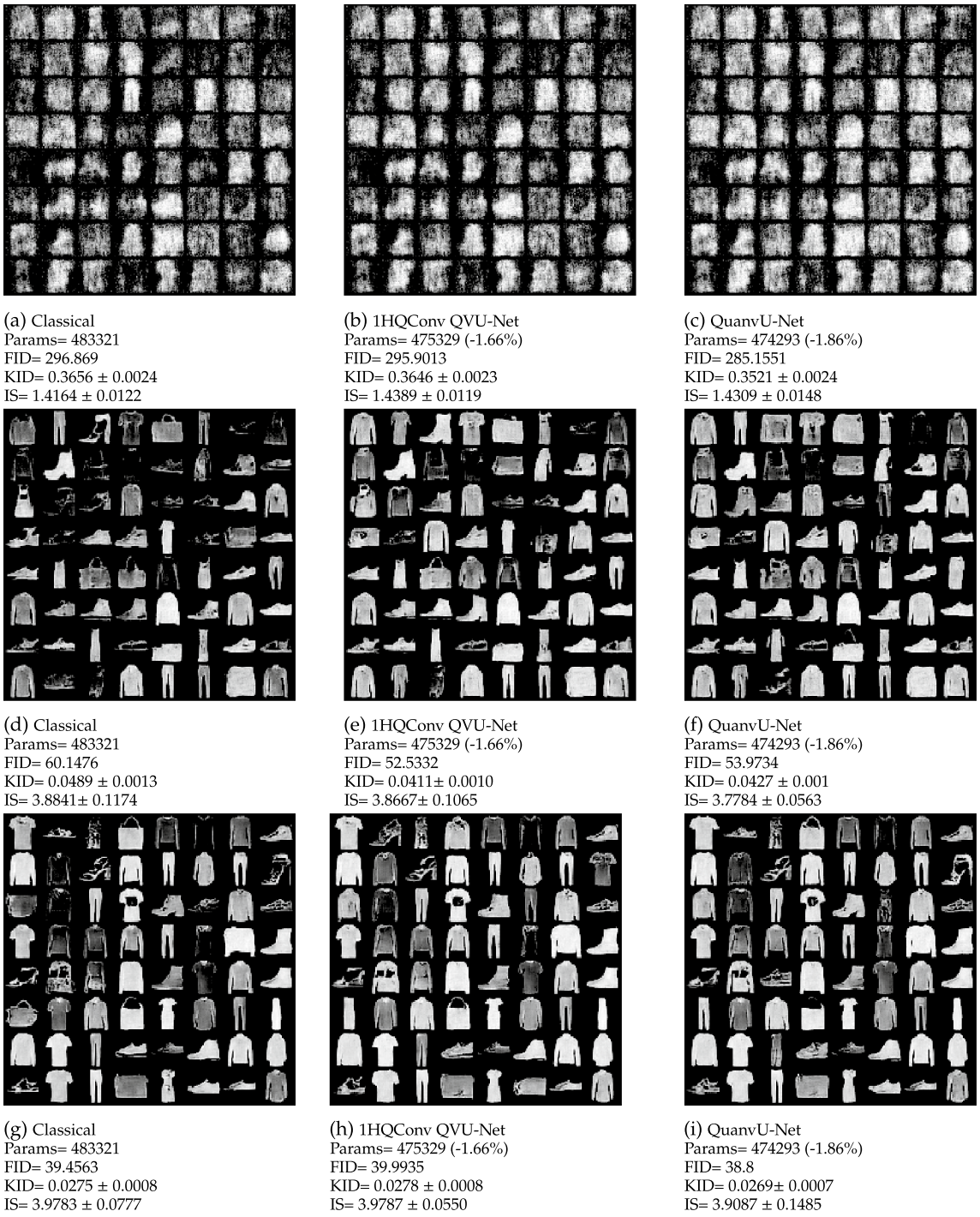


Fig. 10 Fashion MNIST dataset results using the first hybrid architecture. The figure shows in the first column the images generated by the classical U-Net network, while in the second column the images generated by the 1HQConv QVU-Net, in the third column by the 7HQConv QVU-Net, and in the last column the images generated by the FullHQConv QVU-Net. The row-wise division considers in the first row the images generated after the networks are trained for just one epoch, the second row after training for ten epochs, and the third row after the complete training of twenty epochs

QuanvU-Net achieve an FID of 285.1551, in contrast to the classical architecture which has an FID of 296.869. By the tenth epoch, there are no significant differences between the two possible implementations. More interesting is the case of the last epoch Fig. 11i, where initially the introduction of the quantum did not bring any improvement. Now, however, better performance in terms of FID and KID is achieved, as the images of the QuanvU-Net reach an FID of 38.8 and a KID of 0.0269 ± 0.0007, which is better than the FID of 39.4563 and KID of 0.0275 ± 0.0008 obtained from the classical architecture. This is in line with our expectations, given that we have now incorporated a quantum component into a much more critical area of the U-Net, which not only processes but, more importantly, extracts features.

4.3 MNIST Dataset

We then consider the MNIST dataset, whose results are shown in Fig. 12. In this case, we report the results obtained from the first hybridization, the QVU-Net, with the HQConv ansatz implemented with three layers only, as the results obtained using FQConv or the second hybridization, the



(a) Classical
Params= 483321
FID= 296.869
KID= 0.3656 ± 0.0024
IS= 1.4164 ± 0.0122

(b) 1HQConv QVU-Net
Params= 475329 (-1.66%)
FID= 295.9013
KID= 0.3646 ± 0.0023
IS= 1.4389 ± 0.0119

(c) QuanvU-Net
Params= 474293 (-1.86%)
FID= 285.1551
KID= 0.3521 ± 0.0024
IS= 1.4309 ± 0.0148

(d) Classical
Params= 483321
FID= 60.1476
KID= 0.0489 ± 0.0013
IS= 3.8841± 0.1174

(e) 1HQConv QVU-Net
Params= 475329 (-1.66%)
FID= 52.5332
KID= 0.0411± 0.0010
IS= 3.8667± 0.1065

(f) QuanvU-Net
Params= 474293 (-1.86%)
FID= 53.9734
KID= 0.0427 ± 0.001
IS= 3.7784 ± 0.0563

(g) Classical
Params= 483321
FID= 39.4563
KID= 0.0275 ± 0.0008
IS= 3.9783 ± 0.0777

(h) 1HQConv QVU-Net
Params= 475329 (-1.66%)
FID= 39.9935
KID= 0.0278 ± 0.0008
IS= 3.9787 ± 0.0550

(i) QuanvU-Net
Params= 474293 (-1.86%)
FID= 38.8
KID= 0.0269± 0.0007
IS= 3.9087 ± 0.1485

Fig. 11 Fashion MNIST dataset results using the second hybrid architecture. The figure shows in the first column the images generated by the classical U-Net network, while in the second column the images generated by the 1HQConv QVU-Net, in the third column by the QuanvU-Net. The row-wise division considers in the first row the images generated after the networks are trained for just one epoch, the second row after training for 10 epochs, and the third row after the complete training of 20 epochs

QuanvU-Net, that involves the use of quantum in layers other than the vertex are very similar. At the first epoch, hybrid networks have significantly better results than the classical one in terms of all metrics, as the FullHQConv QVU-Net, which is the best-performing hybrid architecture in this case, achieves an FID of 299.7292, a KID of 0.3974 ± 0.0028, and an IS of 1.5930 ± 0.0239. These metrics are better than those obtained from the classical architecture, which has an

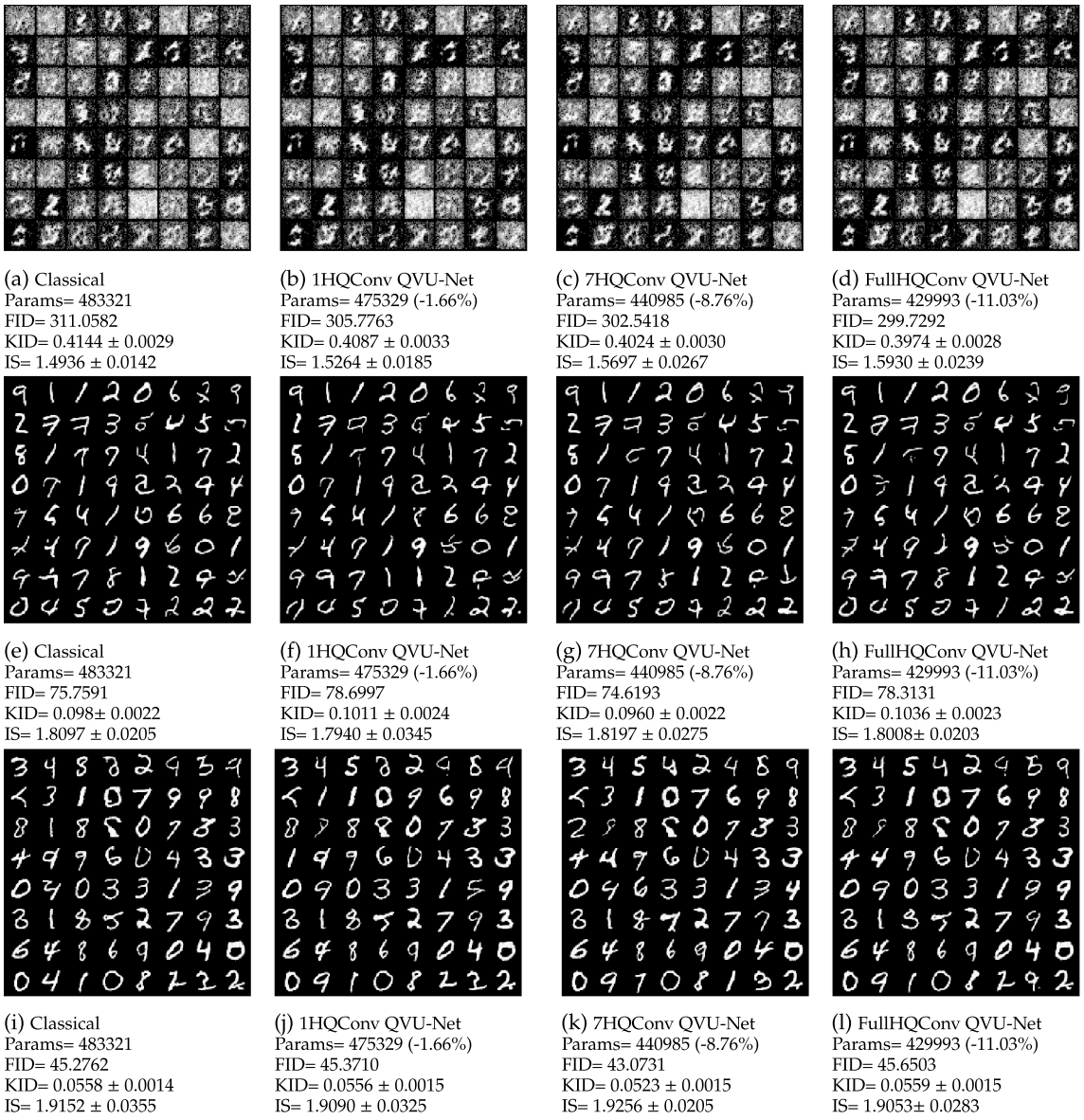


Fig. 12 MNIST dataset results using the first hybrid architecture. The figure shows in the first column the images generated by the classical network, while in the second column the images generated by the 1HQConv QVU-Net, in the third column by the 7HQConv QVU-Net, and in the last column the images generated by the FullHQConv QVU-Net. The row-wise division considers in the first row the images generated after the networks are trained for just one epoch, the second row after training for 10 epochs, and the third row after the complete training of 20 epochs

FID of 311.0582, a KID of 0.4144 ± 0.0029, and an IS of 1.4936 ± 0.0142. Similarly to what we observed with the Fashion MNIST dataset, this once again demonstrates the ability of quantum models to perform better than classical models when training epochs are limited. By the tenth epoch, the advantage diminishes somewhat, except for the 7HQConv QVU-Net architecture shown in Fig. 12g that still shows better values in terms of FID, KID, and IS. At the last epoch, once again the 7HQConv QVU-Net architecture shows better results as shown in Fig. 12k, obtaining an FID of 43.0731, a KID of 0.0523 ± 0.0015, an IS of 1.9256 ± 0.0205, with an FID therefore better by almost 2 points compared to that of the classical architecture, which has an FID of 45.2762, a KID of 0.0558 ± 0.0014, an IS of 1.9152 ± 0.035. We note that all hybrid networks have a lower number of parameters than the classical one.

4.4 Transfer Learning

We can now analyze the results obtained with the transfer learning approach. We compare the metrics of the MNIST dataset images obtained by first training a classical U-Net

network for all the 20 epochs and then training a classical network for 19 epochs and transferring the weights except for the vertex to another classical network that is fully retrained for an additional epoch. Finally, we compare the results obtained by training a classical network for 19 epochs and performing transfer learning on the hybrid network 1FQConv QVU-Net, which is retrained for a single epoch. The results obtained are shown in Fig. 13, demonstrating how the transfer learning approach from classical to hybrid works very well.

What is noticeable is that, by this approach, images are obtained with FID of 41.6646, KID of 0.0493 ± 0.0013, IS of 1.9556 ± 0.0335, which are significantly better even than the best-performing hybrid architecture previously presented, the results of which are shown in Fig. 12k. Thus, by using the transfer learning approach the results obtained from the classical model are significantly improved, with an approximately 8% improvement on FID and 12% on KID for the MNIST dataset.

Similarly, the same approach was taken for the Fashion MNIST dataset, with the results reported in Fig. 14. Instead of having 19 training epochs on the classical network and just one on the network to which the weights have been transferred, we now have 18 epochs on the classical network and 2 on the network to which the weights have been transferred. Also in this case, better results are obtained compared to the end-to-end training of previously proposed hybrid architectures. Indeed, the images shown in Fig. 14c are obtained with FID of 38.6835, KID of 0.0261 ± 0.0007, IS of 4.0527 ± 0.0890, which are superior to those of the best-performing hybrid architecture previously presented, the results of which are shown in Fig. 11i.

Finally, all the numerical results obtained in the previous experiments are summarized in Tables 1 and 2 for

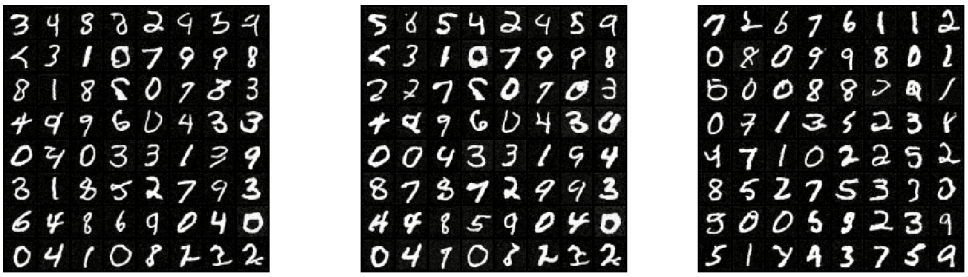


Fig. 13 Transfer learning results on the MNIST dataset where generated images are: **a** classical architecture; **b** transfer learning (19 + 1) classical-classical; **c** transfer learning (19 + 1) classical-1FQConv QVU-Net

(a) Classical
FID= 45.2762
KID= 0.0558 ± 0.0014
IS= 1.9152 ± 0.0355

(b) Transfer learning
FID= 48.9228
KID= 0.0600 ± 0.0014
IS= 1.9094 ± 0.0241

(c) Transfer learning
FID= 41.6646
KID= 0.0493 ± 0.0013
IS= 1.9556 ± 0.0335

Fig. 14 Transfer learning results on the Fashion MNIST dataset where generated images are: **a** classical architecture; **b** transfer learning (18 + 2) classical-classical; **c** transfer learning (18 + 2) classical-1FQConv QVU-Net

(a) Classical
FID= 39.4563
KID= 0.0275 ± 0.0008
IS= 3.8076 ± 0.0741

(b) Transfer Learning
FID= 40.6769
KID= 0.0273 ± 0.0008
IS= 4.0764 ± 0.0945

(c) Transfer Learning
FID= 38.6835
KID= 0.0261 ± 0.0007
IS= 4.0527 ± 0.0890

Table 1 Numerical results for the images generated from the Fashion MNIST dataset

Metrics	Classical	1HQConv QVU-Net	7HQConv QVU-Net	FullHQConv QVU-Net	QuanvU-Net	Transf. learning
Params	483,321	475,329	440,985	429,993	474,293	483,321
FID	39.4563	39.9935	40.3685	41.3882	38.8	**38.6835**
KID	0.0275 ± 0.0008	0.0278 ± 0.0008	0.0281 ± 0.0009	0.0294 ± 0.0009	0.0269 ± 0.0007	$\mathbf{0.0261 \pm 0.0007}$
IS	3.9783 ± 0.0777	3.9787 ± 0.0550	3.8158 ± 0.1360	3.8019 ± 0.0744	3.9087 ± 0.1485	$\mathbf{4.0527 \pm 0.0890}$

Table 2 Numerical results for the images generated from the MNIST dataset

Metrics	Classical	1HQConv QVU-Net	7HQConv QVU-Net	FullHQConv QVU-Net	QuanvU-Net	Transf. learning
Params	483,321	475,329	440,985	429,993	474,293	483,321
FID	45.2762	45.3710	43.0731	45.6503	44.5144	**41.6646**
KID	0.0558 ± 0.0014	0.0556 ± 0.0015	0.0523 ± 0.0015	0.0559 ± 0.0015	0.0545 ± 0.0015	**0.0493 ± 0.0013**
IS	1.9152 ± 0.0355	1.9090 ± 0.0325	1.9256 ± 0.0205	1.9053 ± 0.0283	1.9284 ± 0.0425	**1.9556± 0.0335**

Fashion MNIST and MNIST datasets, respectively, where bold values represent the best ones obtained according to each metric.

5 Conclusions

The use of quantum computing in generative machine learning models can bring numerous advantages, both in terms of performance and in terms of reducing the parameters to be trained. In this paper, we proposed an efficient integration of quantum computing in diffusion models, presenting for the first time two possible hybrid U-Nets. The first, the QVU-Net, involves replacing the convolutional layers that form the ResNet with variational circuits only at the vertex, while the second, the QuanvU-Net, involves the replacement in the second block of the encoder part as well, leveraging in this case an approach inspired by quanvolutional. We also attempted to exploit an approach inspired by transfer learning to reduce the overall training times compared to what we would have had with the complete training of a hybrid architecture.

The obtained results confirm the real advantage in using quantum in extremely complex networks, such as the U-Net of DMs. The approach of hybridizing the U-Net confirms that integrating variational circuits into a classical network can yield certain benefits. Through numerous tests, we proved that quantum allows for further enhancement of network performance. For MNIST, the use of the 7HQConv QVU-Net yielded the best performance. Furthermore, not only at the twentieth epoch did the use of quantum lead to better performance, but in general all hybrid networks show significantly more positive metric values compared to the classical network they are consistently compared against from the first epoch.

On Fashion MNIST, the first possible hybridization of the U-Net, the QVU-Nets, which involves only the hybridized vertex, fails to yield better results than the classical one, despite having a significantly lower number of parameters than the classical network. However, the quantum advantage in this case lies in a faster learning rate, as when we analyze the results at the tenth epoch, all hybrid networks still demonstrate metric values much more advantageous than those observed from the classical network. It is with the second implementation, the QuanvU-Net, which also involves the use of quanvolutional in more expressive layers, that better results are achieved. This demonstrates that having the introduction of quantum in areas dedicated to feature extraction is more effective than introducing it only at the vertex, which primarily operates on processing features already extracted earlier.

Finally, the idea behind using transfer learning between a classical network and a hybrid network is driven by the desire to keep simulation times limited while still achieving better performance than the classical network. This is observed both in the case of MNIST and Fashion MNIST, where we indeed obtain the best results. It is essential to note that the main goal of this paper is to demonstrate that quantum outperforms or performs equally to a classical network with more parameters. It is worth emphasizing that all hybrid networks have significantly fewer parameters.

The possible future developments of this work involve expanding hybridization to other parts of the U-Net, replacing convolutional layers with variational circuits even in regions where the image is less downscaled. This aims to achieve further reduction in the number of trainable parameters, in addition to the potential for improved performance. Furthermore, the goal is to extend the testing to more complex datasets beyond MNIST and Fashion MNIST.

Acknowledgements The contribution in this work of M. Panella, A. Ceschini and F. De Falco was in part supported by the "NATIONAL CENTRE FOR HPC, BIG DATA AND QUANTUM COMPUTING" (CN1, Spoke 10) within the Italian "Piano Nazionale di Ripresa e Resilienza (PNRR)", Mission 4 Component 2 Investment 1.4 funded by the European Union - NextGenerationEU - CN00000013 - CUP B83C22002940006.

Funding Open access funding provided by Università degli Studi di Roma La Sapienza within the CRUI-CARE Agreement.

Data Availibility Statement The data and code that support the findings of this study are openly available in "NesyaLab/Quantum-Hybrid-Diffusion-Models" at https://github.com/NesyaLab/Quantum-Hybrid-Diffusion-Models.

References

1. Farhi E, Neven H (2018) Classification with quantum neural networks on near term processors. arXiv: QuantumPhysics (2018). https://api.semanticscholar.org/CorpusID:119037649
2. Johri S, Debnath S, Mocherla A, Singh A, Prakash A, Kim J, Kerenidis I (2021) Nearest centroid classification on a trapped ion quantum computer. npj Quantum Inf. https://hal.science/hal-03432449
3. Schuld M, Fingerhuth M, Petruccione F (2017) Implementing a distance-based classifier with a quantum interference circuit. EPL (Europhys Lett) 119(6):60002. https://doi.org/10.1209/0295-5075/119/60002
4. Aïmeur E, Brassard G, Gambs S (2013) Quantum speed-up for unsupervised learning. Mach Learn 90(2):261–287. https://doi.org/10.1007/s10994-012-5316-5
5. Benedetti M, Lloyd E, Sack S, Fiorentini M (2019) Parameterized quantum circuits as machine learning models. Quantum Sci Technol 4(4):043001. https://doi.org/10.1088/2058-9565/ab4eb5
6. Bravyi S, Gosset D, König R (2018) Quantum advantage with shallow circuits. Science 362(6412):308–311. https://doi.org/10.1126/science.aar3106
7. Bravyi S, Gosset D, König R, Tomamichel M (2020) Quantum advantage with noisy shallow circuits. Nat Phys 16(10):1040–1045. https://doi.org/10.1038/s41567-020-0948-z
8. Abbas A, Sutter D, Zoufal C, Lucchi A, Figalli A, Woerner S (2021) The power of quantum neural networks. Nat Comput Sci 1(6):403–409. https://doi.org/10.1038/s43588-021-00084-1
9. Sohl-Dickstein J, Weiss E, Maheswaranathan N, Ganguli S (2015) Deep unsupervised learning using nonequilibrium thermodynamics. In: Bach F, Blei D (eds) Proceedings of the 32nd international conference on machine learning, vol 37 of Proceedings of machine learning research, PMLR, Lille, France, pp 2256–2265. https://proceedings.mlr.press/v37/sohl-dickstein15.html
10. Ho J, Jain A, Abbeel P (2020) Denoising diffusion probabilistic models. Adv Neural Inf Process Syst 33:6840–6851
11. Rombach R, Blattmann A, Lorenz D, Esser P, Ommer B (2021) High-resolution image synthesis with latent diffusion models. In: 2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 10674–10685. https://api.semanticscholar.org/CorpusID:245335280
12. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Ghahramani Z, Welling M, Cortes C, Lawrence N, Weinberger K (eds) Advances in neural information processing systems, vol 27. Curran Associates, Inc., pp 1–9. https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf
13. Dhariwal P, Nichol A (2021) Diffusion models beat GANs on image synthesis. Adv Neural Inf Process Syst 34:8780–8794
14. ...Huang H-L, Du Y, Gong M, Zhao Y, Wu Y, Wang C, Li S, Liang F, Lin J, Xu Y, Yang R, Liu T, Hsieh M-H, Deng H, Rong H, Peng C-Z, Lu C-Y, Chen Y-A, Tao D, Zhu X, Pan J-W (2021) Experimental quantum generative adversarial networks for image generation. Phys Rev Appl. https://doi.org/10.1103/physrevapplied.16.024051
15. Bravo-Prieto C, Baglio J, Cè M, Francis A, Grabowska DM, Carrazza S (2022) Style-based quantum generative adversarial networks for Monte Carlo events. Quantum 6:777. https://doi.org/10.22331/q-2022-08-17-777
16. Tsang SL, West MT, Erfani SM, Usman M (2022) Hybrid quantum-classical generative adversarial network for high-resolution image generation. IEEE Trans Quantum Eng 4:1–19
17. Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S (2017) GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in neural information processing systems 30. https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf
18. Parigi M, Martina S, Caruso F (2023) Quantum-noise-driven generative diffusion models. arXiv:2308.12013
19. Cacioppo A, Colantonio L, Bordoni S, Giagu S (2023) Quantum diffusion models . arXiv:2311.15444
20. Ronneberger O, Fischer P, Brox T (2015) U-Net: convolutional networks for biomedical image segmentation. In Navab N, Hornegger J, Wells WM, Frangi AF (eds) Medical image computing and computer-assisted intervention—ICCAI 2015, Springer International Publishing, Cham, pp 234–241. https://doi.org/10.1007/978-3-319-24574-4_28
21. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: IEEE conference on computer vision and pattern recognition (CVPR) 2016, pp 770–778. https://doi.org/10.1109/CVPR.2016.90
22. Sebastianelli A, Zaidenberg DA, Spiller D, Le Saux B, Ullo SL (2021) On circuit-based hybrid quantum neural networks for remote sensing imagery classification. IEEE J Select Top Appl Earth Observ Remote Sens 15:565–580. https://doi.org/10.1109/JSTARS.2021.3134785
23. Ceschini A, Rosato A, Panella M (2022) Hybrid quantum-classical recurrent neural networks for time series prediction. In 2022 International joint conference on neural networks (IJCNN), IEEE, pp 1–8. https://doi.org/10.1109/IJCNN55064.2022.9892441
24. Henderson M, Shakya S, Pradhan S, Cook T (2019) Quanvolutional neural networks: powering image recognition with quantum circuits. arXiv:1904.04767
25. Du Y, Hsieh M-H, Liu T, Tao D (2020) Expressive power of parametrized quantum circuits. Phys Rev Res 2(3):033125. https://doi.org/10.1103/PhysRevResearch.2.033125
26. Wu Y, Yao J, Zhang P, Zhai H (2021) Expressivity of quantum neural networks. Phys Rev Res 3(3):L032049. https://doi.org/10.1103/PhysRevResearch.3.L032049
27. Cerezo M, Arrasmith A, Babbush R, Benjamin SC, Endo S, Fujii K, McClean JR, Mitarai K, Yuan X, Cincio L, Coles PJ (2021) Variational quantum algorithms. Nat Rev Phys 3(9):625–644. https://doi.org/10.1038/s42254-021-00348-9
28. Nichol AQ, Dhariwal P (2021) Improved denoising diffusion probabilistic models. In: Meila M, Zhang T (eds) Proceedings of the 38th international conference on machine learning, vol 139 of Proceedings of machine learning research. PMLR, pp 8162–8171. https://proceedings.mlr.press/v139/nichol21a.html
29. Jing Y, Li X, Yang Y, Wu C, Fu W, Hu W, Li Y, Xu H (2022) RGB image classification with quantum convolutional ansatz. Quantum Inf Process. https://doi.org/10.1007/s11128-022-03442-8
30. Henderson MP, Shakya S, Pradhan S, Cook T (2019) Quanvolutional neural networks: powering image recognition with quantum circuits. Quantum Mach Intell 2. https://api.semanticscholar.org/CorpusID:104291950
31. Mari A, Bromley TR, Izaac J, Schuld M, Killoran N (2020) Transfer learning in hybrid classical-quantum neural networks. Quantum 4:340. https://doi.org/10.22331/q-2020-10-09-340

32. Choi J, Guh J, Zhou Z, Wang Z (2022) Perception prioritized training of diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 2870-2879. https://openaccess.thecvf.com/content/CVPR2022/papers/Choi_Perception_Prioritized_Training_of_Diffusion_Models_CVPR_2022_paper.pdf
33. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. CoRR arXiv:abs/1412.6980 (2014). https://api.semanticscholar.org/CorpusID:6628106
34. LeCun Y, Cortes C, Burges C (2010) MNIST handwritten digit database. ATT Labs [Online] 2. http://yann.lecun.com/exdb/mnist
35. Xiao H, Rasul K, Vollgraf R (2017) Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. CoRR arXiv:abs/1708.07747
36. Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X, Chen X (2016) Improved techniques for training GANs. In: Lee D, Sugiyama M, Luxburg U, Guyon I, Garnett R (eds) Advances in neural information processing systems, vol 29. Curran Associates, Inc., pp 1–9. https://proceedings.neurips.cc/paper_files/paper/2016/file/8a3363abe792db2d8761d6403605aeb7-Paper.pdf
37. Betzalel E, Penso C, Navon A, Fetaya E (2022) A study on the evaluation of generative models. arXiv:2206.10935
38. Detlefsen NS, Borovec J, Schock J, Jha AH, Koker T, Liello LD, Stancl D, Quan C, Grechkin M, Falcon W (2022) Torchmetrics—measuring reproducibility in pyTorch. J Open Source Softw 7(70):4101. https://doi.org/10.21105/joss.04101