Quantum-Noise-Driven Generative Diffusion Models

*Marco Parigi, Stefano Martina, and Filippo Caruso**

Generative models realized with Machine Learning (ML) techniques are powerful tools to infer complex and unknown data distributions from a finite number of training samples in order to produce new synthetic data. Diffusion Models (DMs) are an emerging framework that have recently overcome Generative Adversarial Networks (GANs) in creating high-quality images. Here, is proposed and discussed the quantum generalization of DMs, i.e., three Quantum-Noise-Driven Generative Diffusion Models (QNDGDMs) that could be experimentally tested on real quantum systems. The idea is to harness unique quantum features, in particular the non-trivial interplay among coherence, entanglement, and noise that the currently available noisy quantum processors do unavoidably suffer from, in order to overcome the main computational burdens of classical diffusion models during inference. Hence, the suggestion is to exploit quantum noise not as an issue to be detected and solved but instead as a beneficial key ingredient to generate complex probability distributions from which a quantum processor might sample more efficiently than a classical one. Three examples of the numerical simulations are also included for the proposed approaches. The results are expected to pave the way for new quantum-inspired or quantum-based generative diffusion algorithms addressing tasks as data generation with widespread real-world applications.

1. Introduction

In Machine Learning (ML), diffusion probabilistic models, or briefly Diffusion Models (DMs), are an emerging class of generative models used to learn an unknown data distribution in order to produce new data samples. They has been proposed for the first time by Sohl-Dickstein et al.[1] and take inspiration from diffusion phenomena of non-equilibrium statistical physics. The underlying core idea of the DMs is to gradually and slowly destroy the information encoded into the data distribution until it became fully noisy, and then learn how to restore the corrupted information in order to generate new synthetic data. More precisely, the generic structure of diffusion models consists of two stages: i) a *diffusion* (or *forward*) process and ii) a *denoising* (or *reverse*) process. In the former phase, a training data is progressively perturbed by adding noise, typically Gaussian, until all data information is destroyed. The increasing perturbation of information due to the systematically and progressive injections of noise can be physically understood as if the noise propagates inside the data structure, as shown in **Figure 1** from left to right. Let us highlight the fact that in this first stage the training of any ML model is not required. In the second phase, the previous diffusive dynamics is slowly reversed in order to restore the initial data information. The goal of this phase is to learn how to remove noise correctly and produce new data starting from uninformative noise samples as in Figure 1 from right to left. In contrast to the forward diffusion process, the noise extraction—and as a result the data information retrieval—is implemented training a ML model typically based on a so-called U-Net Neural Network (NN) architecture.[2] In detail, U-Net models are structured in a succession of convolutional layers followed by an equal number of deconvolutional layers, where each deconvolution takes as input the output of the previous deconvolution and also the copy of the output of the corresponding convolutional layer in reverse order. The procedure described above allows DMs to succesfully address the main complication in the design of probabilistic models, i.e., being *tractable* and *flexible* at the same time.[1,3] In fact, alternatively to DMs there are other generative probabilistic models, for instance, Autoregressive Models (ARMs) that are generally tractable but not flexible, or Variational Auto Encoders (VAEs)[4] and Generative Adversarial Networks (GANs)[5] that are flexible but not tractable.

Diffusion models find use in computer vision for several image processing tasks,[6] such as, inpainting,[7] super-resolution,[8] image-to-image translation,[9] and image generation.[10–12] They are also successfully adopted in several applications, for instance: Stable diffusion[13] that is an open source model for high resolution image syntesis;[10] DALL-E 2 that is a platform implemented by OpenAI[14] to generate photorealistic images from text prompts;[11] Google Imagen[15] that combines

M. Parigi, S. Martina, F. Caruso
Department of Physics and Astronomy
University of Florence
Via Sansone 1, Sesto Fiorentino 50019, Florence, Italy
E-mail: filippo.caruso@unifi.it

S. Martina, F. Caruso
LENS - European Laboratory for Non-Linear Spectroscopy
University of Florence
Via Nello Carrara 1, Sesto Fiorentino 50019, Florence, Italy

F. Caruso
Istituto Nazionale di Ottica del Consiglio Nazionale delle Ricerche (CNR-INO)
Sesto Fiorentino I-50019, Italy

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/qute.202300401

DOI: 10.1002/qute.202300401

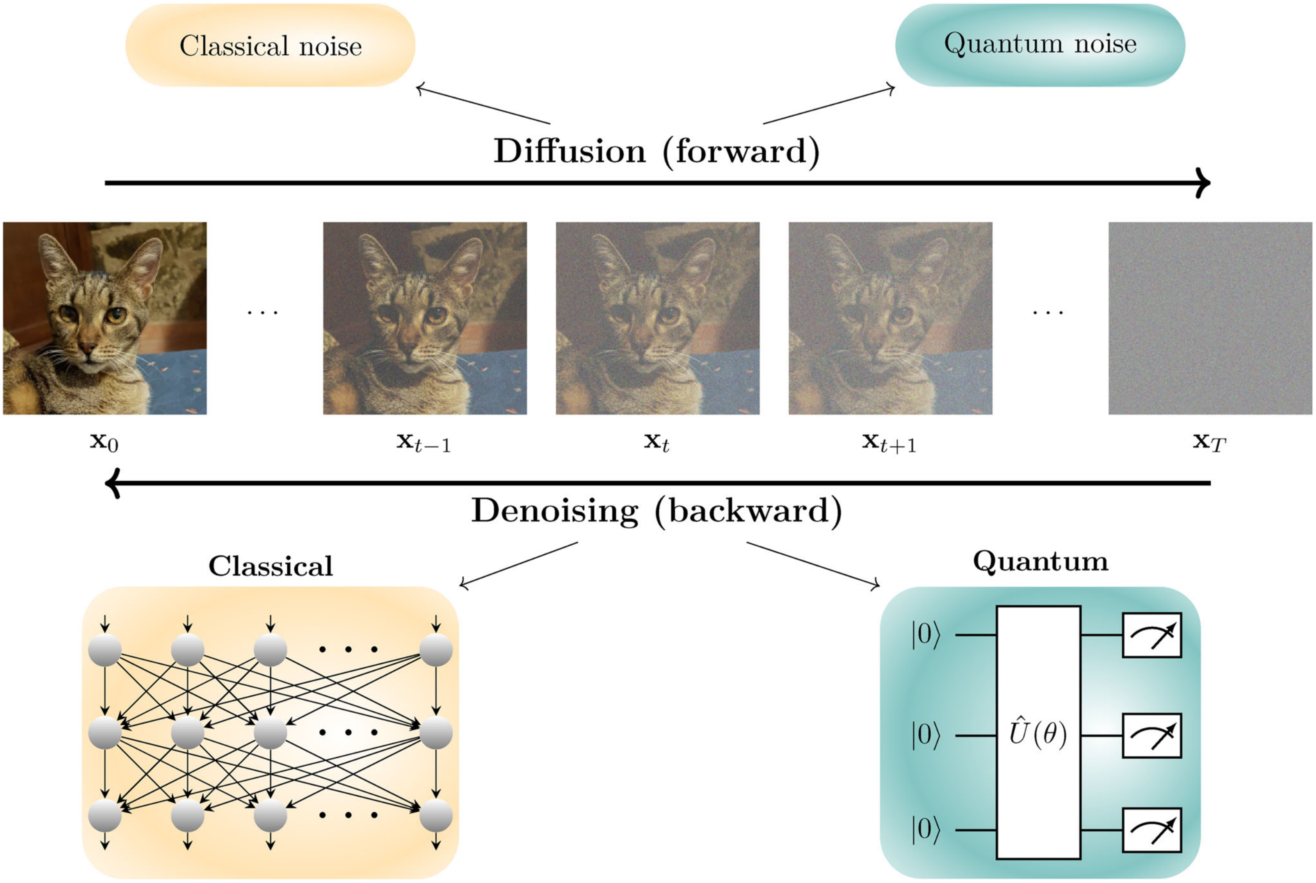

Figure 1. Depiction of the diffusion (from left to right) and denoising (from right to left) processes within a diffusion probabilistic model framework. The original image $\mathbf{x}_0$ sampled from the unknown data distribution $p(\mathbf{x}_0)$ is progressively perturbed ($t \to t+1$) by adding noise to obtain a latent variable $\mathbf{x}_T$ from a known and tractable distribution where the information is completely destroyed. In our framework the diffusion process can be implemented with a classical or a quantum stochastic dynamics. The denoising process is trained to approximate the structure of the data distribution in order to generate new samples. The latter is implemented step by step, using a classical (on the left in orange) or quantum (on the right in green) parameterized model $\hat{U}(\theta)$ in order to approximate the backward mapping. The standard diffusion models implement both the diffusion and the denoising processes in a classical framework. We propose three different new approaches for the other cases: i) classical diffusion and quantum denoising (CQGDM); ii) quantum diffusion and classical denoising (QCGDM); iii) quantum diffusion and quantum denoising (QQGDM). A similar picture can be applied to time series.

transformer language models with diffusion models also in the context of text-to-image generation.[12] Moreover, it has recently been shown that diffusion models perform better than GANs on image synthesis.[16]

Furthermore, diffusion models can also be applied to other contexts, for instance in text generation[17,18] and time-series related tasks.[19–21] For instance, time series forecasting is the task of predicting future values from past history and diffusion models can be employed to generate new samples from the forecasting distribution.[22,23] Diffusion models can also be used in time series generation, which is a more complex task involving the complete generation of new time-series samples from a certain distribution.[24,25]

On the other side, very recently, we are witnessing an increasing interest in quantum technologies. Near-term quantum processors are called Noisy Intermediate-Scale Quantum (NISQ) devices[26] and they represent the-state-of-the-art in this context. NISQ computers are engineered with quantum physical systems using different strategies. For instance, a commonly used technology employs superconductive-circuits-based platforms[27,28] realized with transmon qubits.[29,30] This technology is exploited, for instance, by IBM,[31] Rigetti Computing,[32] Google.[33] Moreover D-Wave[34] exploits superconducting integrated circuits mainly as quantum annealers.[35] Xanadu[36] is instead a company employing photons as information units within the linear optical quantum computing paradigm[37] to realize their devices. Finally, the quantum computation can be realized directly manipulating the properties of single atoms. For instance, IonQ[38] realizes quantum devices with trapped ions,[39,40] while Pasqal[41] and QuEra[42] realize analog quantum computers with Rubidium Rydberg neutral atoms held in optical tweezers.[43] All the mentioned devices can be in principle integrated in computational pipelines that can involve also classical computation. In this context they can be referred with the term Quantum Processing Unit (QPU) that can make some computational task much faster than its classical counterpart (CPU) harnessing the quantum properties of particles at the atomic scale. The main reason for building a quantum processor is the possibility of exploiting inherent and peculiar resources of quantum mechanical systems such as *superposition*, *coherence*, and *entanglement* that, in some cases, allow to perform computational tasks that are impossible or much more difficult via a classic supercomputer.[44,45] In particular, the peculiar properties of quantum systems could lead to *quantum speedup* on some tasks compared to their classical counterpart.[46–48]

One of the most promising applications of NISQ devices is represented by Quantum Machine Learning (QML) that is a recent interdisciplinary field merging ML and quantum computing fields in a way such that data to be processed and/or learning algorithms are quantum.[44,49–51] Indeed, it involves the integration of ML techniques and quantum computers in order to process and subsequently analyze/learn the underlying data structure. QML can involve the adoption of classical ML methods with quantum data or environments, for instance to analyze noise in quantum devices[52–55] or to control quantum systems.[56] Alternatively, QML can consider the implementation of novel ML techniques using quantum devices, for instance to implement visual tasks[57] or generative models like Quantum Generative Adversarial Network (QGAN)[58–62] that are the quantum implementation of classical GAN or in Natural Language Processing (NLP) context to generate text.[63] In fact, quantum devices are capable of processing information in ways that are different from the classical computation. Thus, the implementation of QML models can offer an advantage over the corresponding classical ML models,[64,65] especially in the context of generative models.[66] However, NISQ devices are indeed still very noisy and thus they do not perform the ideal (pure) dynamics. Therefore, the system evolution is affected and driven by *quantum noise* due to the undesired interactions with external environment and has to be described by the more general open quantum system formalism.[67]

In order to generalize DMs with quantum computing ideas, a crucial role is played by noise. In classical information theory, noise is usually modeled by the framework of probability theory, and in general via Markovian processes. Accordingly, the main features of classical noise are a linear relationship among successive steps of the dynamics whose evolution depends only on the current state. Formally, noise is represented by a transition matrix that has the properties of *positivity* (non-negative entries) and *completeness* (columns summing to one). In particular, Gaussian noise is a type of random noise that is very often added to the input data of a DM in order to help its learning to generate new data that is as similar as possible to the training data, also in the case when the input is not perfect.

In the quantum domain, noise can be generated also by quantum fluctuations that are typical of quantum systems, hence going much beyond the classical noise sources. Mathematically, quantum noise is described by the more general formalism of *quantum operations* or *quantum maps*,[67] where, for instance, the decoherence is the typical noise affecting the phase coherence among the quantum states. The noise is, in fact, the main enemy to fight in order to build up quantum processors that are more powerful than the classical counterpart. Indeed, recent works show that the noise can even destroy quantum computation making the dynamic more classically simulable.[68] But what about if such noise is not only detrimental for the quantum computation but it is instead actually beneficial for some ML tasks? Even considering the aforementioned detrimental effects, there are theoretical and experimental evidences[56,69,70] that quantum noise can improve the efficiency of information transport and a noisy quantum dynamics can diffuse faster than the noiseless equivalent. Quantum noise might allow, for instance, to generate more complex (due to the presence of entanglement) probability distributions that would be difficult, or even impossible, to express classically and from which is possible to sample more efficiently via a quantum processor than via a classical supercomputer.

Compared to other generative models, classical DMs require a large number of steps, both for the diffusion and the denoising phases. This means that, when used in data generation, the sampling is computationally expensive because it requires to iterate through all such steps. Inspired by the aforementioned physical systems where the quantum noise accelerates the diffusion, in this article we therefore introduce and formalize the quantum versions of DMs, in particularly based on Denoising Diffusion Probabilistic Models (DDPMs) and Score Stochastic Differential Equations (Score SDEs) in the context of QML. More precisely, we propose three potential Quantum-Noise-Driven Generative Diffusion odels (QNDGDMs) that can be both computationally simulated in the NISQ devices and implemented *experimentally* due to the naturally occurring noise effects in open quantum systems. The three algorithms are: i) Classical-Quantum Generative Diffusion Model (CQGDM) in which the forward diffusion process can be implemented in the classical way, while the backward denoising with a Quantum Neural Network (QNN) (that can be either a Parametrized Quantum Circuit (PQC) or an hybrid quantum-classical NN); ii) Quantum-Classical Generative Diffusion Model (QCGDM) in which the noise diffusion process can be implemented in a quantum way, while in the denoising process classical NNs are used; iii) Quantum-Quantum Generative Diffusion Model (QQGDM) where both the diffusion and the denoising dynamics can be implemented in a quantum domain.

2. Results

2.1. Classical-Quantum Generative Diffusion Model (CQGDM)

In this section we propose a model where the diffusion process is classical while the denoising phase is implemented with a quantum dynamics. Moreover, as a result of this setting, the training dataset is necessarily classical, for instance, images, videos, time series, etc.

Formally, given an initial training data $\mathbf{x}_0$ sampled from a generic and unknown probability distribution $p(\mathbf{x}_0)$, the procedure consists in a progressive destruction of the information encoded in the initial data via a diffusive stochastic process. At the end, the data is degraded to a fully noisy state $\mathbf{x}_T$ sampled from a classical closed form and tractable *prior* distribution $p(\mathbf{x}_T)$ that represents the latent space of the model. Here, tractable stands for the fact that the distribution can be computationally calculated. The implementation of this process can be obtained with different ways. For instance, in DDPMs the dynamics of forward diffusive process is implemented by a classical Markov chain,[1,3] while in Score SDEs the stochastic evolution is determined by a differential equation.[71] In detail, the former approach considers a discrete-time stochastic process whose evolution, at every step, depends only on the previous state and the transition relies on hand-designed kernels $p(\mathbf{x}_t|\mathbf{x}_{t-1})$,

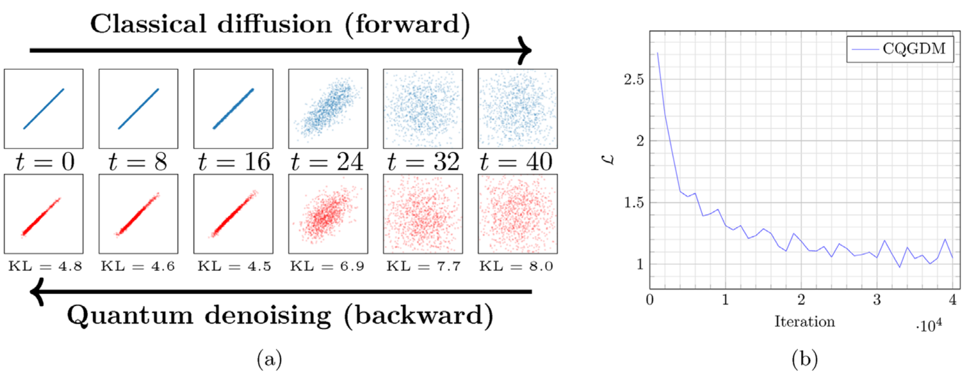

Figure 2. a) Evolution of the data distribution for a trained simulated CQGDM and b) its Kullback-Leibler divergence loss function $\mathcal{L}$ during the training, averaged every 1 000 iterations. The initial data distribution consists of 2D points distributed in a line segment between −1 and 1. The diffusion process is implemented via a classical diffusion process that transforms the initial data distribution $p(\mathbf{x}_0)$ at time $t = 0$ to the prior $p(\mathbf{x}_T)$ that is a normalized Gaussian distribution at the final time $t = 40$. Meanwhile, the denoising is implemented via a (noiseless) simulated PQC to reconstruct the initial data distribution ($t = 0$) from the Gaussian prior ($t = 40$). In the top row of (a), we show the forward process (from left to right) for a sample of 1 000 points at different discrete time steps $t = 0, 8, \dots, 40$. In the bottom row, we display the denoising (from right to left) of a different sample of 1 000 points. Under the figure is reported also the Kullback-Leibler divergence between the data distribution $p(\mathbf{x}_0)$ and the reconstructed distribution at the corresponding time.

$t = 1, 2, \dots, T$ (see Section Classical Methods for more details). Alternatively, in Score SDEs the evolution is a continuous-time process within a close time interval $t \in [0, T]$ and determined by the stochastic differential equation: $\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$, where $\mathbf{f}(\mathbf{x}, t)$ is the drift coefficient, $g(t)$ is the diffusion term, and $\mathbf{w}$ is the Wiener process (also known as standard Brownian motion) that models the stochastic process.[72] The solutions of this equation lead to the tractable prior distribution $p(\mathbf{x}(T))$.

Afterwards, in order to generate new data samples, the objective is to learn how to reverse the diffusion process starting from the prior latent distribution. In case of DDPMs, calculating $p(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is not computationally tractable and it is classically approximated by a model parameterized with θ (e.g., a NN): $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ (see Section Classical Methods). In the case of Score SDE models, the quantity to be estimated is $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$, where $p_t(\mathbf{x})$ is the density probability of $\mathbf{x}(t)$.[71] Here, for either DDPMs and Score SDE diffusion processes, we propose to implement the denoising process with a classical method aided by a QNN that can be fully quantum, via a PQC, or even a classical-quantum hybrid NN model. The results of a simulation of this type of algorithm on a dataset composed of 2D points distributed along a line segment in the interval $[-1, 1]$ is shown in **Figure 2**a. At the best of our knowledge, this is the first implementation of a hybrid classical-quantum diffusion model, and indeed represents a starting point for more in-depth future studies. The model is capable to reconstruct the initial data distribution $p(\mathbf{x}_0)$ with a good approximation that is quantified by the Kullback-Leibler (KL) divergence $\mathrm{KL}(p(\mathbf{x}_0)||p_\theta(\mathbf{x}_{t:T}))$ between the data distribution and the one reconstructed starting from the Gaussian distribution $p(\mathbf{x}_T)$ until time t. In detail, we estimate the KL from the 1 000 samples using the method described in ref. [73]. In Figure 2b we show the evolution of the loss during the training of the model averaged every 1 000 iterations (more details on the model and the implementation in Section Classical-Quantum Generative Diffusion Model (CQGDM)).

In this context, the main advantage of using the quantum denoising process instead of the classical one can be the possibility of using the trained quantum model to efficiently generate highly dimensional data (e.g., images) taking advantage of the peculiar quantum mechanical properties, such as quantum superposition and entanglement, to speed up data processing.[74–76] Indeed, QPU devices could be very effective to overcome the main computational burdens of classical diffusion model during this inference process. As shown in Figure 4, the denoising process for CQGDM crosses the border between classical and quantum distribution spaces, this could take advantage of the quantum speedup in order to accelerate the training of the model. Moreover, in literature there are evidences of a quantum advantage of using PQCs instead of deep NNs. For instance, it is proved that PQCs outperform classical NNs in generative tasks[77] and it is shown that PQCs have an exponential advantage in model size respect to NNs in function approximation of high dimensional smooth functions.[78] Therefore, it is plausible that we also can take advantage in using PQCs instead of NNs in our context.

2.2. Quantum-Classical Generative Diffusion Model (QCGDM)

In real experiments quantum systems are never perfectly isolated, but they are easily subjected to noise, e.g., interactions with the environment and imperfect implementations. Accordingly, we propose to physically implement the diffusion process via a noisy quantum dynamics.

In this setting a quantum dataset is considered, i.e., a collection of *quantum* data. Classical information can be embedded into the initial state of a quantum system, allowing to treat classical data as quantum.[51,79,80] Even better, we could avoid the encoding of the classical data if we consider quantum data as any result arising from a quantum experiment[81] or produced directly by a quantum sensing technology.[65] Formally, a quantum data is identified with the density operator ρ living in $\mathfrak{S}(\mathcal{H})$ being the set of non-negative operators with unit trace acting on the elements of the Hilbert space $\mathcal{H}$ where the quantum states live.

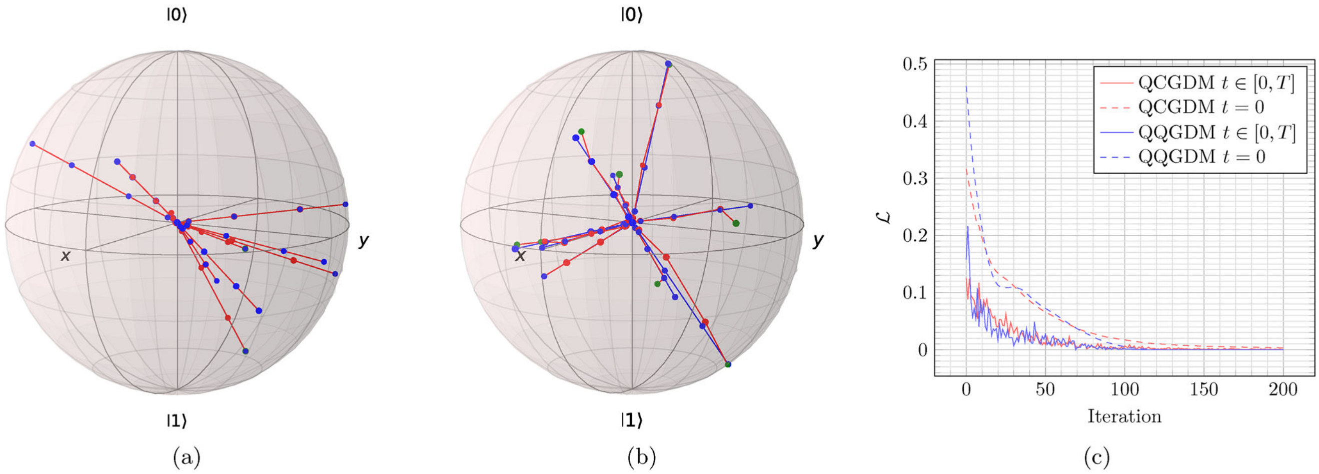

Figure 3. Reconstruction of ten random one-qubit pure states for trained a) QCGDMs and b) Quantum-Quantum Generative Diffusion Models (QQGDMs), and c) the evolution of the quantum infidelity loss for both models for one of the ten states. In (a) and (b), the blue segments represent the evolution of the forward dynamic implemented with a depolarizing quantum channel. In (a) the red segments are the evolutions of the backward process implemented with a neural network and in (b) with a parameterized quantum circuit. The green points are the final reconstructed states. The solid lines in (c) report the evolution of the loss used during the training while the dashed lines are the same loss but calculated only on the final reconstructed states. The red lines refers to the QCGDM and the blue lines to QQGDM.

We here propose two approaches to implement the diffusion process: i) *quantum* Markov chains generalizing their classical counterparts,[82] and ii) Stochastic Schrödinger Equation (SSE)[83–86] modeling the dynamics of an *open quantum system* subjected to an external noise source.

In the former approach (i), a quantum Markov chain can be described with a composition of Transition Operation Matrices (TOMs) mapping a density operator ρ to another density operator ρ'. TOMs are matrices whose elements are *completely positive maps* and whose column sums form a *quantum operation* (for more details refer to Section 4). A special case of TOMs are the Transition Effect Matrices (TEMs) whose columns are discrete Positive Operator Valued Measures (POVMs). A quantum Markov chains can be, thereby, implemented by a sequence of quantum measurements.[82]

The second approach (ii) employs SSEs to describe the physical quantum diffusion process. Given a system in the state $\rho(t)$, its stochastic evolution is determined by a SSE that takes the form $\dot{\rho}(t) = -i[H(t), \rho(t)]$, with $\hbar = 1$, and where the Hamiltonian $H(t) = H_s(t) + H_p(t)$ consists of the sum of the Hamiltonian of the system $H_s(t)$ and the stochastic term $H_p(t)$ representing the stochastic dynamics to which the quantum system is subjected. Arbitrary sources of noise applied to optimally controlled quantum systems were very recently investigated with the SSEs formalism by our group.[87]

As a practical implementation of the Quantum-Classical Generative Diffusion Model (QCGDM) model, we decide to realize the forward dynamic as a discrete quantum Markovian chain composed by the iteration at each time step of a *depolarizing* quantum channel $\rho_{t+1} = (1-p)\rho_t + p\frac{\mathbb{I}}{d}$ with $p \in [0, 1]$, $t \in [0, T]$ and d the dimension of the considered quantum system. In this way, the (quantum) information encoded in the initial quantum state ρ_0 is progressively degraded until we reach the maximally mixed state $\rho_T \equiv \frac{\mathbb{I}}{d}$. Formally, the loss of information on the quantum state of a system can be quantified by the von Neumann entropy $S(\rho) = -\mathrm{tr}(\rho \log_2 \rho)$ that is zero for *pure* states $\rho = |\psi\rangle\langle\psi|$, strictly positive for *mixed* states $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$, with $p_i \geq 0$ and $\sum_i p_i = 1$, and it is maximal and equal to $S(\frac{I}{d}) = \log_2 d$ for the maximally mixed state.[44] Regarding the implementation of the backward, we use classical neural networks that predict $\hat{\rho}_t$ from the input at time $t+1$. A specific neural network is trained for each time step t to maximize the fidelity between the ρ_t obtained during the forward pass and the $\hat{\rho}_t$ obtained from the NN when its input is ρ_{t+1} in turn from the forward. In generation after the training, the NN can be iteratively used to obtain $\hat{\rho}_0$ from the initial maximally mixed state $\frac{\mathbb{I}}{d}$ predicting, at each time step t, $\hat{\rho}_t$ from the previous prediction $\hat{\rho}_{t+1}$. In **Figure 3**a, we show the simulation on a single qubit system of ten QCGDMs and $T = 5$. The neural network is capable to reconstruct the information of the initial state ρ_0. The average reconstruction quantum fidelity on a sample of 100 different random states is equal to 0.997 ± 0.013. In Figure 3c is visible, with the red curves, the evolution of the infidelity for batches of data during the training for one of the single states and also the infidelity loss between $\hat{\rho}_0$ and the ρ_0 in dashed (see Section QCGDM for more details).

In our simulations, the forward process is classically simulated, therefore we obtain all the states ρ_t directly from the simulator. If we want to use real NISQ devices, we need to perform quantum state tomography to obtain the ρ_t, and classically calculate the infidelity loss between the reconstructed state and the $\hat{\rho}_t$ predicted by the NN. As a possible solution to avoid tomography, we could prepare the predicted $\hat{\rho}_t$ and use the *swap test*, which is a common method to obtain the fidelity between quantum states,[57] to compute the loss. In any case, both strategies are used only during the training of the model. After the training, the NN is a generative model that classically simulate the generation of quantum states starting from the maximally mixed state, representing the noisy state.

The implementation of diffusion dynamics on quantum systems during the forward stage can allows the processing of the data information not only by classically simulated noise but also with quantum physical noise. Here, as previously mentioned, let us remind that quantum noise is more general with respect to its classical counterpart. In particular, the noise distributions used in QCGDMs can be expressed (and more naturally arise by quantum dynamics) in more general and powerful forms respect to the typical Gaussian distributions that are commonly employed in classical DMs. In this set up, at the end of the diffusion process, it is possible to obtain non-classical prior distributions related to entangled state that do not exist in the classical information scenario. In other terms there are probability density distributions that are purely quantum. This can be used to implement diffusion processes that are not possible to be implemented classically. At the end, during the denoising phase, classical NNs can be used in order to remove noise and thus finally generate new samples. Moreover, if the obtained prior distribution is not classical, it is possible to consider the adoption of the denoising NN as a discriminator to identify probability distributions that are purely quantum. This could also be framed in a *security* context. One can imagine a channel where the communication of data takes place with the application of a quantum diffusion process that maps to a purely quantum probability distribution. In that case, the receiver can restore and so obtain the initial information only with the training of a QNN and thus only with a quantum device. This might be also exploited for quantum attacks/defence in cyber-security applications.

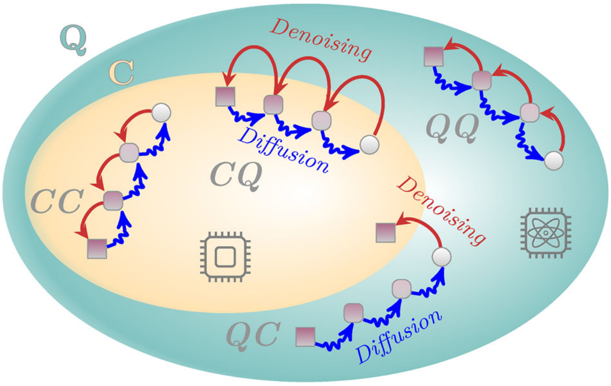

Figure 4. Relationship between the space of the probability distributions that are tractable with classical computation (**C**) and instead only with quantum computation (**Q**). We show the trajectories arising from the mappings between probability distributions (colored and white shapes) during the diffusion (blue wavy arrows) and denoising (red arrows) processes for the four different combination: **CC**, **CQ**, **QC** and **CC** indicating whether the diffusion (first letter) and the denoising (second letter) are classical or quantum. The initial data distribution (squares) is progressively transformed during diffusion (changing color and shape) to an uninformative distribution represented by the white circles, and vice versa during denoising. Completely classical models are limited to operate within the space of classically-tractable probability distributions, while completely quantum models can manipulate quantum-tractable probabilities. Models that have classical diffusion and quantum denoising are forced to work only with classical probabilities, but during the denoising phase they can exploit quantum properties within each step. Finally, models that have quantum diffusion and classical denoising can manipulate quantum probabilities during the forward, but in that case, it is not possible to train the classical backward to map those probability distributions.

2.3. Quantum-Quantum Generative Diffusion Model (QQGDM)

In this last section we describe diffusion models within a fully quantum physical framework. Precisely, the training data, the diffusion process and the denoising process have all a quantum mechanical nature. This scenario can be obtained by exploiting the quantum tools described above, namely, quantum Markov chain or SSE for the forward diffusion phase, and a PQC for the backward denoising phase.

Accordingly, all the advantages described in Sections 2.1, 2.2 hold. The adoption of a fully quantum pipeline for both the diffusion and denoising phases would allow the possibility to obtain purely quantum prior distributions that can be processed during the denoising phase with PQCs obtaining a generation process that is not feasible classically. As shown in **Figure 4**, the diffusion and denoising processes for QQGDM are entirely located in the space of quantum distributions. This might lead to the speedup already described previously for CQGDM and in addition to the possibility of exponentially reducing the computational resources for storing and processing of data information.[76] Finally, it is also possible to access to complex quantum probability distributions that are impossible or much more difficult to treat classically.

Similarly to QCGDM model, we propose a practical implementation of the forward quantum noise dynamic with a depolarizing channel to degrade the initial state ρ_0 to the maximally mixed state ρ_T. The quantum backward process should be an equally noisy dynamic, in fact an unitary dynamics is not sufficient to reconstruct any initial state different from the maximally mixed one: $\frac{\mathbb{I}}{d} \mapsto U\frac{\mathbb{I}}{d}U^{\dagger} = \frac{\mathbb{I}}{d}$. For this reason, we propose an implementation of the backward via the interaction of a system with an external environment that we trace out at each time step. In detail, in our numerical simulations we consider a single-qubit system coupled via a parameterized circuit to another single-qubit that acts as the environment. During training, at each time step t, the system is initialized with ρ_{t+1} and, after the environment tracing-out, the state is $\hat{\rho}_t$. In generation, it is possible to obtain $\hat{\rho}_0$, starting from the maximally mixed state $\frac{\mathbb{I}}{2}$, iteratively applying the denoising with the previous prediction as input. Similarly to the QCGDM, the loss is based on the infidelity between the quantum states ρ_t from the forward and $\hat{\rho}_t$ predicted during the backward (for more details see Section QCGDM). In our classical simulations we have access directly to the quantum states ρ_t and $\hat{\rho}_t$ at each time step and, therefore, the loss can be straightforwardly computed. On real NISQ devices, the loss could be obtained with the help of swap test. Practically, the forward process could be implemented in two parallel copies of the system. The first one stops at time t to prepare ρ_t while the second one makes a further step to obtain ρ_{t+1}. The backward system can be appended to the second copy to prepare $\hat{\rho}_t$ and trained using the swap test between the two states.

In Figure 3c in blue is reported the evolution of the loss during the training of one QQGDMs and in Figure 3b the resulting simulations for ten different initial states. For each one, we train the model to reconstruct a random pure state with $T = 5$. We can observe that the model is capable to learn the reverse dynamic from the maximally mixed state ρ_T at the center of the Bloch sphere to the state $\hat{\rho}_0$. Furthermore, we compute the average reconstruction quantum fidelity for 100 random states obtaining 0.996 ± 0.0086.

3. Conclusion

The entanglement is a crucial quantum mechanical phenomenon occurring only in the quantum domain (not classical analogue) when two or more quantum systems interact. It is detected by measurement correlations between the quantum systems that cannot be described with classical physics. Accordingly, quantum systems are capable of representing distributions that are impossible to be produced efficiently with classical computers.[49,88] For this reason, a quantum diffusion process is capable to explore probability density functions that are not classically tractable.

In Figure 4 we highlight the relationship between the space of the probability distributions that are tractable with classic computers, which we denote with *classical distributions* to be more concise, and the space of the probability distributions that are tractable with quantum devices, which we denote with *quantum distributions* hereinafter. Moreover, we can observe several possible trajectories that map probability distributions to other probability distributions during the diffusion and denoising of the classical DM and of the three proposed quantum approaches: CQGDM, QCGDM, and QQGDM.

The classic DM realizes maps from classical distributions to other classical distributions and the NN that implements the denoising are trained to realize the inverse maps, i.e., to match the distributions crossed during the diffusion.

In the CQGDM approach, the diffusion process is implemented classically. Thus, all the probability distributions are necessarily classical. However, during the denoising process, the quantum dynamics is free to explore also the quantum probability space within each one of the steps hence exploiting potential (noise-assisted and/or quantum-enhanced) shortcuts. This may give advantages for the training of the denoising model. Moreover, when we evolve quantum systems within a QPU it is possible to process and manipulate exponentially more information as compared to the classical case.

When we consider the fully quantum framework QQGDM we gain the advantage of exploring quantum distributions also during the diffusion phase. For this reason we could explore more complex noisy dynamics compared to the ones that can be simulated in classical computers. Moreover, the two processes can be experimentally implemented on real quantum processors. Furthermore, compared to the CQGDM approach, and provided that the initial distribution of the dataset is quantum, it is possible to design a QQGDM generative models that is capable of generating complex quantum data that are not analytically computable.

Besides, we would like point out that the QCGDM approach can be challenging to be implemented. In detail, if the diffusion process leads to an entagled quantum distribution it is impossible, for the previously mentioned reasons, to efficiently train a classical NN to perform the denoising. This context could be adopted as a proof of concept for the realization of a discriminator for the quantum distributions from the classical ones. In other words, if it is possible to train a model to perform the denoising, then the distribution is classical.

After the second version of our work was published on arXiv,[89] related works have appeared, showing indeed a remarkable research interest on this topic. Compared to ref. [89], the current version of our manuscript contains two more examples for QCGDM and QQGDM.

More precisely, in ref. [90] the authors propose an implementation of a quantum generative diffusion model where the forward phase is implemented with random unitaries iteratively applied at every time step t and the backward with trainable PQCs with ancilla systems that are measured after each step. The main difference with respect to our approach is that in their case all the states ρ_t and $\hat{\rho}_t$ during respectively the forward and backward phases are pure. Instead in our proposed method all the intermediate states, and in principle also the initial ρ_0 can be mixed states.

Another research group in ref. [91] has implemented an hybrid model where the classic forward degrades images similarly to DMs, while the quantum backward reconstructs the images with PQCs. The main difference with our example implementation is that we use the PQC as a real-valued function to predict the parameters of a classical Markov chain, while they use it as a quantum operator directly to reconstruct the state.

Finally, another recent work[92] has later proposed a fully quantum generative diffusion model where the forward phase iteratively degrades an initial quantum state ρ_0 using depolarizing channels to the maximally mixed state ρ_T and the backward restores ρ_0 with a parameterized quantum operation. Our QQGDM example approach is similar to their implementation, but with some differences. The first difference is that they use a cosine noise schedule in the forward, while we use a linear schedule. The second difference regards the denoising process where in order to reconstruct $\hat{\rho}_{t-1}$ from the state ρ_t they also use a quantum circuit to perform the embedding of time t on quantum state τ_t and then employ a PQC that acts on the state of total system $\tau_t \otimes \rho_t$. Next, they trace out the qubits of τ_t and obtain the state $\hat{\rho}_{t-1}$. Instead, in our implementation of the backward we use a single PQC with parameters specific to each timestep t without any embedding of time. Moreover, they also provide a second implementation that uses less resources (qubits) for the backward.

As a future outlook, we would like to realize the implementation of the QNDGDMs either computationally via NISQ and/or physically by using quantum sensing technologies. In particular, regarding QCGDMs and QQGDMs, we propose to implement the diffusion process exploiting naturally noisy quantum dynamics in order to take advantage of the possible benefits of the quantum noise. Instead, regarding CQGDMs and QQGDMs, we propose to use quantum implemented QML models, for instance QNNs and PQCs, to learn the denoising process.

A possible future work direction could be to study the applicability of other kinds of loss functions. In fact, there are evidences that the adoption of KL divergence loss in the context of quantum generative models leads to the formation of a new flavor of barren plateaus.[93] Related to this, we plan to deepen the study of the possible noise-induced speedup of the diffusion dynamic and the trainability of the quantum QNDGDMs. Moreover, other kinds of quantum channels could enhance the diffusion models by fully exploiting the quantum properties over all the diffusion and denoising dynamic. For instance, we can consider the adoption of coherent or non-unital noise such as the amplitude damping.

Finally, the design and realization of QNDGDMs, with respect to classical DMs, could alleviate and reduce the computational resources (e.g., space of memory, time, and energy) to successfully address ML applications such as generation of high-resolution images, the analysis and the prediction of rare events in time-series, and the learning of underlying patterns in experimental data coming also from very different fields as, among others, life, and earth science, physics, quantum chemistry, medicine, material science, smart technology engineering, and finance.

4. Experimental Section

In this section, some mathematical details are included on the classical and quantum tools for the diffusion and denoising processed discussed in the main text.

Classical Methods: Here, we formalize the classical methods used in the standard generative diffusion models and for the relevant part of the proposed CQGDM and QCGDM. In particular, classical Markov chains were considered for a Gaussian pertubation and the NNs were used for the classical denoising.

The classical diffusion process[1,3] starts from an initial data sample $\mathbf{x}_0$ drawn from an unknown generic distribution $p(\mathbf{x}_0)$. Gaussian noise is then iteratively injected for a number T of time steps to degrade the data to $\mathbf{x}_T$ sampled from a prior Gaussian distribution $\mathcal{N}(0, \mathbf{I})$. In detail, the used Gaussian transition kernel is in the form:

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

where $\beta_t \in (0, 1)$ is an hyperparameter (fixed or scheduled over the time) for the model at the time step t that describes the level of the injected noise, $\mathbf{I}$ is the identity matrix, and $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$ are the random variables at the time steps t and $t-1$, respectively. In this way it is possible to calculate a tractable closed form for the trajectory:

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

By obing so, for T sufficiently high, $p(\mathbf{x}_{1:T}|\mathbf{x}_0)$ converges to an isotropic Gaussian $p(\mathbf{x}_T) \approx \mathcal{N}(0, \mathbf{I})$. Moreover, given an initial data $\mathbf{x}_0$, a data sample $\mathbf{x}_t$ can be obtained by sampling a Gaussian vector $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} \tag{3}$$

where $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=0}^{t} \alpha_s$.

The denoising phase starts from the Gaussian prior distribution and the transition kernel that is implemented is in the form:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \tag{4}$$

and the closed form for the trajectory is:

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \tag{5}$$

Usually, a NN, specifically a U-Net architecture,[2] is used to estimate the mean $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$ and the covariance $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)$ in Equation (4). In principle, the approach to train the NN would be to find the parameters θ such that $p_\theta(\mathbf{x}_0)$ would be maximized for each training sample $\mathbf{x}_0$. However $p_\theta(\mathbf{x}_0)$ is intractable because it is impossible to marginalize over all the possible trajectories. For this reason, the common approach is to minimize the KL loss:

$$\begin{aligned}\mathcal{L} &= \mathrm{KL}(p(\mathbf{x}_{0:T})||p_\theta(\mathbf{x}_{0:T})) \\ &= -\mathbb{E}_p[\log p_\theta(\mathbf{x}_{0:T})] + \mathrm{const} \\ &= \mathbb{E}\left[-\log p(\mathbf{x}_T) - \sum_{t\geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{p(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] + \mathrm{const} \\ &\geq \mathbb{E}[-\log p_\theta(\mathbf{x}_0)] + \mathrm{const}\end{aligned} \tag{6}$$

Quantum Methods: Here, the use of the quantum Markov chain is formalized and introduced for the diffusion processes of QCGDM and QQGDM in Sections 2.2 and 2.3 and the QNNs used for the denoising of CQGDM and QQGDM in Sections 2.1 and 2.3.

Formally, a quantum Markov chain can be described by two elements: i) a directed graph G whose sites represent the possible state that the quantum system can occupy, ii) a TOM $\mathcal{E} = \mathcal{E}_{ij}$ whose elements are *completely positive maps*[67,94] and whose column sums form a *quantum operation*.[44,82] Formally, a positive a map is a linear transformation of one positive bounded operator into another. A completely positive map is a linear map $\phi : \mathcal{B}(\mathcal{H}) \rightarrow \mathcal{B}(\mathcal{H})$, where $\mathcal{B}(\mathcal{H})$ is the set of bounded linear operators acting on the Hilbert space $\mathcal{H}$, such that the map $\phi \otimes I$ is positive on the space $\mathcal{B}(\mathcal{H}) \otimes \mathcal{B}(\mathcal{H}')$ for any Hilbert space $\mathcal{H}'$. A quantum operation is a completely positive map ϕ preserving the trace, i.e., $\mathrm{tr}(\rho) = \mathrm{tr}(\phi(\rho))$, with $\rho \in \mathcal{B}(\mathcal{H})$. Physically, the elements E_{ij} describe the passage operation of the quantum system from site j to site i in one time step. Given a density operator ρ, representing the state of system, the quantity $\mathcal{E}(\rho)$ is again a density operator. Moreover, if $\mathcal{E}$ and $\mathcal{F}$ are two TOMs with the same size and acting on the same Hilbert space, then the $\mathcal{EF}$ is again a TOM by matrix multiplication. Accordingly, the dynamics of the quantum system after a discrete number of time steps n i described by the map $\mathcal{E}^n = \mathcal{E}\mathcal{E}^{n-1}$, with $n = 2, 3 \ldots$, and the initial state ρ is transformed in the final state $\mathcal{E}^n(\rho)$.

Let us introduce the concepts of QNN[51] in the QML framework and how they are trained. Formally, a QNN can be written as a product of layers of unitary operations:

$$\hat{U}(\boldsymbol{\theta}) = \prod_{\ell=1}^{L} \hat{V}_\ell \hat{U}_\ell(\boldsymbol{\theta}_\ell) \tag{7}$$

where $\hat{V}_\ell$ and $\hat{U}_\ell(\boldsymbol{\theta}_\ell)$ are fixed and parameterized unitary operations, respectively, for ℓ^{th} layer of QNN. The output of the QNN is:

$$f(\boldsymbol{\theta}) = \mathrm{tr}(\mathcal{M}\rho_\theta) \tag{8}$$

where $\mathcal{M}$ is an Hermitian operator representing the physical observable, $\rho_\theta = \hat{U}(\boldsymbol{\theta})\rho_0\hat{U}^\dagger(\boldsymbol{\theta})$ and ρ_0 is the initial state, which is the input of the QNN. The QNN is optimized minimizing the difference between its output and the desired value. Generally, the latter is performed with the gradient descent method with the adoption of the parameters shift rule.[95]

Simulations: Here, we describe in detail both the model and its implementation regarding the simulation of the CQGDM, QCGDM and QQGDM used to obtain the results of Figures 2, 3a,b, respectively.

CQGDM: For the training of the CQGDM model illustrated in Section 2.1, a dataset composed of points (x, y) distributed along a segment of the line $y = x$ in the interval $[-1, 1]$ was used. The model was trained on batches of 1 000 points uniformly random sampled for each step.

The diffusion process was implemented via a classical Markov chain composed of a sequence of Gaussian transition kernels as in Equation (1) in order to map the initial data distribution $p(\mathbf{x}_0)$ to an isotropic Gaussian $p(\mathbf{x}_T)$ with final time $T \equiv 40$. Furthermore, the data sampling at each time step t was computed by using Equation (3).

The denoising process was realized via a PQC and trained to estimate the mean $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$ and the covariance $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)$ of the kernel of the classical denoising Markov chain in Equation (4). The model was built and simulated with the help of the *Pennylane*[96] and *PyTorch*[97] libraries. More precisely, the PQC, whose output is in the form of Equation (8), consists of a four qubits circuit divided in two concatenated parts called *head* and *tail*. The parameters of the head were shared among all the values of t, while the parameters of the tail were specific for each value of $t = 0, \dots, 39$. In particular, the head takes as input the values of the coordinates of a single point and encode them in the state of the first two qubits with an *angle embedding*,[51] while the other two qubits were initialized to $|0\rangle$. After the embedding, the circuit was composed of 256 layers of parametric rotations on the three axes for all the four qubits alternated by layers of entangling controlled not gates (in particular, *StronglyEntanglingLayers* of *Pennylane* was used with default arguments[98]). At the end of the circuit, measurements were performed and the expectation values of the observable Pauli matrix σ_z on all four qubits were computed. The tail was similarly composed, except that the first operation was the angle embedding of the four expectation values previously obtained from the head. In order to simplify the model, it was assumed that the denoising kernel was uncorrelated among the features and therefore, the covariance matrix was diagonal and only two values for the variance were necessary. This assumption, analogous to the one of the classical DDPM, was justified by the fact that the denoising was implemented with a quantum-aided Markov chain and between each couple of time steps $t, t-1$ the data was classical and also the evolution of its distribution. Finally, the four expectation values measured from the tail were used for the predictions of the mean (the first two values) and variance (the second two values) of the classical kernel. In detail, the expectation values used for the mean were multiplied by a factor 3 in order to enlarge the possible range and the values for the variance were increased by 1 to force positivity. The model was trained for 40 000 epochs on random batches of 1 000 points to minimize the KL divergence loss of Equation (6) between the predicted and desired Gaussian distributions using *Adam*[99] with learning rate 10^{-4}. The evolution of such loss during the training is shown in Figure 2b averaged over every 1 000 iterations. The plots of Figure 2a were obtained, after the training of the model, using two different random batches of 1 000 points, one for the forward and another one for the backward. In detail, the top left plot in Figure 2a is the initial random batch of the data distributed with $p(\mathbf{x}_0)$ in the line segment. The batch was iteratively corrupted by the classical Markov chain with the kernel of Equation (1) and on the top part of the figure we report the evolution of the distribution at the intermediate time steps $t = 8, 16, 24, 32$ and at the final time $t = T \equiv 40$. The kernel parameter in Equation (1) was taken equal to $\beta_t = 0.3 \times \sigma(b_t)$ with $\sigma(x) = 1/(1+e^{-x})$ the sigmoid function and b_t equally spaced in $[-18, 10]$ for the 40 steps. The properties of the classical diffusion processes guarantee that for sufficiently large T any initial distribution converges to the isotropic Gaussian distribution $p(\mathbf{x}_T)$. The bottom right plot is another random batch sampled from $\mathcal{N}(0, \mathbf{I})$ that was iteratively denoised with a parametrized classical Markov chain with the kernel of Equation (4). The kernel parameters were predicted by the previously mentioned PQC and in the figure we report in the bottom part the evolution $p_\theta(\mathbf{x}_T), \dots, p_\theta(\mathbf{x}_{t:T}), \dots, p_\theta(\mathbf{x}_{0:T})$ of such process on the batch for the same values of t of the forward.

QCGDM: Here, we describe in details the implementation of the quantum forward and classical backward used for the single qubit QCGDM model illustrated in Section 2.2. The quantum forward dynamics was classically simulated in Pennylane with a circuit initialized with a pure quantum state that was iteratively degraded at each time step $t \in [1, T \equiv 5]$. The noise was introduced by a depolarizing channel applied T times. Formally, the action of the channel on the quantum state is described by:

$$\rho_t = (1 - p_t)\rho_{t-1} + \frac{p_t}{3}(X\rho X + Y\rho Y + Z\rho Z) \tag{9}$$

where $p_t \in [0, 1]$ represents the probability that the qubit depolarizes at time t. The values of p_t were linearly scheduled in time with uniformly spaced values in $[1, T]$. All the values of the states $\rho_0, \dots, \rho_T$ were collected and used for the training of the backward phase.

The denoising process was implemented with neural networks that were trained to simulate the reverse noisy dynamic in order to obtain an approximation $\hat{\rho}_0$ of the initial pure state ρ_0 starting from the maximally mixed state $\frac{\mathbf{I}}{2} \equiv \rho_T$. The full process was implemented in PyTorch with T different neural networks specialized to reconstruct the mixed state $\hat{\rho}_t$ from another mixed state at time $t+1$. Each one of them had a single hidden layer of *five* neurons with ReLU activation function, $f(x) = max(x, 0)$, for a total of 32 parameters, and it was trained to minimize the loss between the ρ_t from the forward pass and the $\hat{\rho}_t$ predicted when the input is ρ_{t+1}. In detail, the loss $\mathcal{L}$ was computed and minimized for batches of $n \equiv 16$ reconstructed states $\hat{\rho}_t$ with random values of t. Formally, the loss was defined by:

$$\mathcal{L} = \frac{1}{n} \sum_{t \sim [0,T]^n} (1 - F(\hat{\rho}_t, \rho_t)) \tag{10}$$

where $F(\rho_1, \rho_2) = \left(\mathrm{tr}\sqrt{\sqrt{\rho_1}\rho_2\sqrt{\rho_1}}\right)^2$ is the quantum fidelity between two states. The optimization was performed with Adam with learning rate 10^{-2}. The loss function was classically calculated in our simulations. However, the swap test could be used in a possible implementation on NISQ devices.

As the problem was formalized, the inputs and outputs of the neural networks were mixed quantum states. A quantum state for one qubit is represented by a 2×2 positive semi-definite Hermitian matrix with trace one, for this reason to represent it are necessary only three real values. The three elements could be expressed as spherical coordinates with radius $r \in [0, 1]$, polar angle $\theta \in [0, \pi]$ and azimuth $\phi \in [0, 2\pi)$. Therefore, the input and output layers of the neural networks were both of dimension three and the final activation functions for the three neurons were respectively $\sigma(r)$, $\pi\sigma(\theta)$, and $2\pi\sigma(\phi)$ with $\sigma(\cdot)$ the sigmoid function.

QQGDM: In this section, we give the details for the implementation of the QQGDM in Section 2.3. The forward phase was implemented with the same depolarizing channel of QCGDM explained in Section QCGDM, while the backward phase was realized with a quantum dynamic. The latter, was implemented with a non-unitary dynamic for each time step $t \in [T-1, 0]$, with $T = 5$. In particular, such dynamic was formalized in the context of open quantum systems with one qubit initialized with $\hat{\rho}_{t+1}$ (in prediction) that interacts with an environment that was traced out to prepare the output state $\hat{\rho}_t$ on the same system qubit. The environment relied on another qubit and the interaction with the system was realized with a trainable PQC on both qubits. In detail, the used PQCs were implemented with the Pennylane StronglyEntanglingLayers with five layers and default other arguments for a total of 30 trainable parameters. The t-specific PQCs were trained analogously to the neural networks of Section QCGDM. In particular, using Adam with learning rate 10^{-2} to minimize the loss of Equation (10) calculated on batches composed by 16 pairs $(\hat{\rho}_t, \rho_t)$ with random values of t where $\hat{\rho}_t$ was obtained when the system was initialized with ρ_{t+1} from the forward.

Acknowledgements

M.P. and S.M. acknowledged financial support from PNRR MUR project PE0000023-NQSTI. F.C. also acknowledged financial support by the European Commission's HORIZON EUROPE Framework Programme under the Research and Innovation Action GA n. 101070546–MUQUABIS, by the European Union's Horizon 2020 research and innovation programme under FET-OPEN GA n. 828946–PATHOS, by the European Defence Agency under the project Q-LAMPS Contract No B PRJ-RT-989, and under the National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.1, Call for tender No. 104 published on 2.2.2022 by the Italian Ministry of University and Research (MUR), funded by the European Union NextGenerationEU Project Title "Trustworthy hybrid quantum-classical Artificial Intelligence for Medical Image Analysis (ThAI-MIA)" CUP B53D23005000006.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

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

Keywords

diffusion models, generative models, quantum computing, quantum machine learning, quantum noise

Received: November 11, 2023
Revised: June 12, 2024
Published online:

[1] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, S. Ganguli, in *Proceedings of the 32nd International Conference on Machine Learning*, Proceedings of Machine Learning Research, (Eds.: F. Bach, D. Blei), vol. *37*, PMLR, Lille, France **2015**, pp. 2256–2265.

[2] O. Ronneberger, P. Fischer, T. Brox, in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, (Eds.: N. Navab, J. Hornegger, W. M. Wells, A. F. Frangi), Springer International Publishing, Cham, Switzerland **2015**, pp. 234–241.

[3] J. Ho, A. Jain, P. Abbeel, in *Advances in Neural Information Processing Systems*, vol. *33*, (Eds.: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin), Curran Associates, Inc., New York, NY, USA **2020**, pp. 6840–6851.

[4] D. P. Kingma, M. Welling, *Found. Trends Mach. Learn.* **2019**, *12*, 307.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, *Commun. ACM* **2020**, *63*, 139.

[6] F.-A. Croitoru, V. Hondru, R. T. Ionescu, M. Shah, *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 10850.

[7] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, L. Van Gool, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Curran Associates, Inc., New York, NY, USA **2022**, pp. 11461–11471.

[8] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, M. Norouzi, *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 4713.

[9] C. Saharia, W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet, M. Norouzi, in *ACM SIGGRAPH 2022 Conference Proceedings*, SIGGRAPH '22. Association for Computing Machinery, New York, NY, USA **2022**.

[10] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Piscataway, NJ **2022**, pp. 10684–10695.

[11] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, M. Chen, (Preprint) arXiv:2204.06125, v1, submitted: Apr **2022**.

[12] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, J. Ho, D. J. Fleet, M. Norouzi, *Adv. Neural Inform. Process. Syst.* **2022**, *35*, 36479.

[13] Stable diffusion, https://ommer-lab.com/research/latent-diffusion-models (accessed: July 2024).

[14] Dall-e 2, https://openai.com/dall-e-2 (accessed: July 2024).

[15] Google imagen, https://imagen.research.google (accessed: July 2024).

[16] P. Dhariwal, A. Nichol, in *Advances in Neural Information Processing Systems*, (Eds.: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan), vol. *34*, Curran Associates, Inc., New York, NY, USA **2021**, pp. 8780–8794.

[17] N. Savinov, J. Chung, M. Binkowski, E. Elsen, A. van den Oord, *(Preprint) arXiv:2112.06749*, v3, submitted: Apr **2022**.

[18] P. Yu, S. Xie, X. Ma, B. Jia, B. Pang, R. Gao, Y. Zhu, S.-C. Zhu, Y. N. Wu, in *International Conference on Machine Learning*, PMLR, New York **2022**, pp. 25702–25720.

[19] L. Lin, Z. Li, R. Li, X. Li, J. Gao, *Front. Inform. Techn. Electron. Eng.* **2023**, *25*, 19.

[20] Y. Tashiro, J. Song, Y. Song, S. Ermon, in *Advances in Neural Information Processing Systems*, (Eds.: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan), vol. *34*, Curran Associates, Inc., New York, NY, USA **2021**, pp. 24804–24816.

[21] J. M. Lopez Alcaraz, N. Strodthoff, *Transactions on Machine Learning Research* **2023**, *163*, 107115.

[22] K. Rasul, C. Seward, I. Schuster, R. Vollgraf, in *Proceedings of the 38th International Conference on Machine Learning*, (Eds.: M. Meila, T. Zhang), vol. *139*, Proceedings of Machine Learning Research, New York **2021**, pp. 8857–8868.

[23] Y. Li, X. Lu, Y. Wang, D. Dou, in *Advances in Neural Information Processing Systems*, (Eds.: S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh), vol. *35*, Curran Associates, Inc., New York, NY, USA **2022**, pp. 23009–23022.

[24] H. Lim, M. Kim, S. Park, N. Park, *(Preprint) arXiv:2301.08518*, v1, submitted: jan **2023**.

[25] E. Adib, A. S. Fernandez, F. Afghah, J. J. Prevost, *IEEE Access* **2023**, *11*, 7581.

[26] J. Preskill, *Quantum* **2018**, *2*, 79.

[27] M. H. Devoret, A. Wallraff, J. M. Martinis, *(Preprint) arXiv:cond-mat/0411174*, v1, submitted: Nov **2004**.

[28] J. Clarke, F. K. Wilhelm, *Nature* **2008**, *453*, 1031.

[29] J. Koch, T. M. Yu, J. Gambetta, A. A. Houck, D. I. Schuster, J. Majer, A. Blais, M. H. Devoret, S. M. Girvin, R. J. Schoelkopf, *Phys. Rev. A* **2007**, *76*, 042319.

[30] J. A. Schreier, A. A. Houck, J. Koch, D. I. Schuster, B. R. Johnson, J. M. Chow, J. M. Gambetta, J. Majer, L. Frunzio, M. H. Devoret, S. M. Girvin, R. J. Schoelkopf, *Phys. Rev. B* **2008**, *77*, 180502.

[31] Ibm quantum experience, https://quantum-computing.ibm.com (accessed: July 2024).

[32] Rigetti computing, https://www.rigetti.com (accessed: July 2024).

[33] Google quantum ai, https://quantumai.google (accessed: July 2024).

[34] D-wave, https://www.dwavesys.com.

[35] M. W. Johnson, M. H. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, E. M. Chapple, C. Enderud, J. P. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M. C. Thom, E. Tolkacheva, C. J. S. Truncik, S. Uchaikin, J. Wang, B. Wilson, G. Rose, *Nature* **2011**, *473*, 194.

[36] Xanadu quantum technologies, https://xanadu.ai (accessed: July 2024).

[37] P. Kok, W. J. Munro, K. Nemoto, T. C. Ralph, J. P. Dowling, G. J. Milburn, *Rev. Mod. Phys.* **2007**, *79*, 135.

[38] Ionq, https://ionq.com (accessed: July 2024).

[39] S. Allen, J. Kim, D. L. Moehring, C. R. Monroe, in *2017 IEEE International Conference on Rebooting Computing (ICRC)*, IEEE, Piscataway **2017**, pp. 1–3.

[40] C. Monroe, J. Kim, *Science* **2013**, *339*, 1164.

[41] Pasqal, https://www.pasqal.com (accessed: July 2024).

[42] Quera, https://www.quera.com (accessed: July 2024).

[43] L. Henriet, L. Beguin, A. Signoles, T. Lahaye, A. Browaeys, G.-O. Reymond, C. Jurczak, *Quantum* **2020**, *4*, 327.

[44] M. A. Nielsen, I. L. Chuang, *Quantum computation and quantum information*, Cambridge university press, Cambridge **2010**.

[45] R. P. Feynman, *Int. J. Theor. Phys.* **1982**, *21*, 467.

[46] D. Deutsch, R. Jozsa, *Proc. R. Soc. London. Series A: Math. Phys. Sci.* **1992**, *439*, 553.

[47] P. W. Shor, *SIAM J. Comput.* **1997**, *26*, 1484.
[48] L. K. Grover, in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96. Association for Computing Machinery, New York, NY, USA **1996**, pp. 212–219.
[49] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, S. Lloyd, *Nature* **2017**, *549*, 195.
[50] P. Wittek, *Quantum machine learning: what quantum computing means to data mining*, Academic Press, Cambridge **2014**.
[51] M. Schuld, F. Petruccione, *Machine learning with quantum computers*, Springer, Berlin, Heidelberg **2021**.
[52] E. Canonici, S. Martina, R. Mengoni, D. Ottaviani, F. Caruso, *Adv. Quantum Technol.* **2024**, *7*, 2300192.
[53] S. Martina, S. Hernández-Gómez, S. Gherardini, F. Caruso, N. Fabbri, *Mach. Learn.: Sci. Technol.* **2023**, *4*, 02LT01.
[54] S. Martina, S. Gherardini, F. Caruso, *Phys. Scr.* **2023**, *98*, 035104.
[55] S. Martina, L. Buffoni, S. Gherardini, F. Caruso, *Quantum Mach. Intell.* **2022**, *4*, 8.
[56] N. Dalla Pozza, L. Buffoni, S. Martina, F. Caruso, *Quantum Mach. Intell.* **2022**, *4*, 11.
[57] S. Das, J. Zhang, S. Martina, D. Suter, F. Caruso, *Quantum Mach. Intell.* **2023**, *5*, 16.
[58] S. Lloyd, C. Weedbrook, *Phys. Rev. Lett.* **2018**, *121*, 040502.
[59] P.-L. Dallaire-Demers, N. Killoran, *Phys. Rev. A* **2018**, *98*, 012324.
[60] C. Zoufal, A. Lucchi, S. Woerner, *npj Quantum Inf.* **2019**, *5*, 103.
[61] P. Braccia, F. Caruso, L. Banchi, *New J. Phys.* **2021**, *23*, 053024.
[62] P. Braccia, L. Banchi, F. Caruso, *Phys. Rev. Appl.* **2022**, *17*, 024002.
[63] A. Karamlou, J. Wootton, M. Pfaffhauser, in *Proceedings of the 15th International Conference on Natural Language Generation*, (Eds.: S. Shaik, T. Ferreira, A. Stent), Association for computational Linguistic, Waterville, Maine, USA **2022**, pp. 267–277.
[64] D. Ristè, M. P. Da Silva, C. A. Ryan, A. W. Cross, A. D. Córcoles, J. A. Smolin, J. M. Gambetta, J. M. Chow, B. R. Johnson, *npj Quantum Inf.* **2017**, *3*, 16.
[65] H.-Y. Huang, R. Kueng, J. Preskill, *Phys. Rev. Lett.* **2021**, *126*, 190505.
[66] M. Hibat-Allah, M. Mauri, J. Carrasquilla, A. Perdomo-Ortiz, *Commun. Phys.* **2024**, *7*, 68.
[67] F. Caruso, V. Giovannetti, C. Lupo, S. Mancini, *Rev. Mod. Phys.* **2014**, *86*, 1203.
[68] D. Aharonov, X. Gao, Z. Landau, Y. Liu, U. Vazirani, in *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, Association for Computing Machinery, New York, NY, USA **2023**, pp. 945–957.
[69] F. Caruso, S. F. Huelga, M. B. Plenio, *Phys. Rev. Lett.* **2010**, *105*, 190501.
[70] F. Caruso, A. Crespi, A. G. Ciriolo, F. Sciarrino, R. Osellame, *Nat. Commun.* **2016**, *7*, 11682.
[71] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, B. Poole, in *International Conference on Learning Representations*, (Eds.: A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, S. Levine), vol. *36*, Curran Associate, Inc., New York, NY, USA **2021**, pp. 37799–37812.
[72] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, M.-H. Yang, *ACM Comput. Surv.* **2023**, *56*, 4.
[73] F. Perez-Cruz, in *2008 IEEE International Symposium on Information Theory*, IEEE, Piscataway, NJ **2008**, pp. 1666–1670.
[74] S. Lloyd, M. Mohseni, P. Rebentrost, *Nat. Phys.* **2014**, *10*, 631.
[75] N. Wiebe, D. Braun, S. Lloyd, *Phys. Rev. Lett.* **2012**, *109*, 050505.
[76] X.-W. Yao, H. Wang, Z. Liao, M.-C. Chen, J. Pan, J. Li, K. Zhang, X. Lin, Z. Wang, Z. Luo, W. Zheng, J. Li, M. Zhao, X. Peng, D. Suter, *Phys. Rev. X* **2017**, *7*, 031041.
[77] Y. Du, M.-H. Hsieh, T. Liu, D. Tao, *Phys. Rev. Res.* **2020**, *2*, 033125.
[78] Z. Yu, Q. Chen, Y. Jiao, Y. Li, X. Lu, X. Wang, J. Z. Yang, *(Preprint) arXiv:2310.07528*, v1, submitted: Oct **2023**.
[79] S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, N. Killoran, *(Preprint) arXiv:2001.03622*, v2, submitted: Feb **2020**.
[80] I. Gianani, I. Mastroserio, L. Buffoni, N. Bruno, L. Donati, V. Cimini, M. Barbieri, F. S. Cataliotti, F. Caruso, *Adv. Quantum Technol.* **2022**, *5*, 2100140.
[81] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, J. R. McClean, *Nat. Commun.* **2021**, *12*, 2631.
[82] S. Gudder, *J. Math. Phys.* **2008**, *49*, 072105.
[83] E. B. Davies, J. T. Lewis, *Commun. Math. Phys.* **1970**, *17*, 239.
[84] H. M. Wiseman, *Quantum Semiclassical Opt.: J. European Opt. Soc. Part B* **1996**, *8*, 205.
[85] H.-P. Breuer, F. Petruccione, *The theory of open quantum systems*, Oxford University Press, Oxford **2002**.
[86] A. Rivas, S. F. Huelga, *Open quantum systems*, vol. *10*, Springer, Berlin, Heidelberg **2012**.
[87] M. M. Müller, S. Gherardini, T. Calarco, S. Montangero, F. Caruso, *Sci. Rep.* **2022**, *12*, 21405.
[88] A. Bouland, B. Fefferman, C. Nirkhe, U. Vazirani, *Nat. Phys.* **2019**, *15*, 159.
[89] M. Parigi, S. Martina, F. Caruso, *(Preprint) arXiv:2308.12013, v2, submitted: Sept* **2023**.
[90] B. Zhang, P. Xu, X. Chen, Q. Zhuang, *Phys. Rev. Lett.* **2024**, *132*, 100602.
[91] A. Cacioppo, L. Colantonio, S. Bordoni, S. Giagu, *e-print arXiv:2311.15444* **2023**.
[92] C. Chen, Q. Zhao, *(Preprint) arXiv:2401.07039, v2, submitted: Jun* **2024**.
[93] M. S. Rudolph, S. Lerch, S. Thanasilp, O. Kiss, S. Vallecorsa, M. Grossi, Z. Holmes, *(Preprint) arXiv:2305.02881, v1, submitted: May* **2023**.
[94] G. Lindblad, *Commun. Math. Phys.* **1975**, *40*, 147.
[95] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, N. Killoran, *Phys. Rev. A* **2019**, *99*, 032331.
[96] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M. S. Alam, G. Alonso-Linaje, B. AkashNarayanan, A. Asadi, J. M. Arrazola, U. Azad, S. Banning, C. Blank, T. R. Bromley, B. A. Cordier, J. Ceroni, A. Delgado, O. D. Matteo, A. Dusko, T. Garg, D. Guala, A. Hayes, R. Hill, A. Ijaz, T. Isacsson, D. Ittah, S. Jahangiri, P. Jain, E. Jiang, et al., *(Preprint) arXiv:1811.04968, v4, submitted: Jul* **2022**.
[97] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, *Adv. Neural Inf. Proc. Syst.* **2019**, *32*, 8026.
[98] M. Schuld, A. Bocharov, K. M. Svore, N. Wiebe, *Phys. Rev. A* **2020**, *101*, 032308.
[99] D. P. Kingma, J. Ba, *(Preprint) arXiv:1412.6980, v9, submitted: Jan* **2017**.