

Quantum latent distributions in deep generative models

Omar Bacarreza, Thorin Farnsworth, Alexander Makarovskiy, Hugo Wallner, Tessa Hicks,
Santiago Sempere-Llagostera, John Price, Robert J. A. Francis-Jones, and William R. Clements*
ORCA Computing

Many successful families of generative models leverage a low-dimensional latent distribution that is mapped to a data distribution. Though simple latent distributions are commonly used, it has been shown that more sophisticated distributions can improve performance. For instance, recent work has explored using the distributions produced by quantum processors and found empirical improvements. However, when latent space distributions produced by quantum processors can be expected to improve performance, and whether these improvements are reproducible, are open questions that we investigate in this work. We prove that, under certain conditions, these "quantum latent distributions" enable generative models to produce data distributions that classical latent distributions cannot efficiently produce. We also provide actionable intuitions to identify when such quantum advantages may arise in real-world settings. We perform benchmarking experiments on both a synthetic quantum dataset and the QM9 molecular dataset, using both simulated and real photonic quantum processors. Our results demonstrate that quantum latent distributions can lead to improved generative performance in GANs compared to a range of classical baselines. We also explore diffusion and flow matching models, identifying architectures compatible with quantum latent distributions. This work confirms that near-term quantum processors can expand the capabilities of deep generative models.

I. INTRODUCTION

In recent years, several successful families of deep generative models have emerged that learn to map samples from a latent distribution, a typically low-dimensional probability distribution, to a higher-dimensional data distribution. For instance, models that use latent distributions such as generative adversarial networks (GANs) [1, 2], latent diffusion models [3, 4], flow models [5, 6] and variational autoencoders (VAEs) [7, 8] have all achieved remarkable performance on a wide range of datasets from images to proteins. During training, these models learn a mapping between the structure of the latent distribution and that of the data. As such, the structure of this latent distribution plays a critical role. For instance, in GANs, making the structure of the latent distribution used by the generator correspond to that of the data has been shown to significantly improve performance [2, 9, 10].

The design space in which these new algorithms have been developed has typically been limited to the range of functions that are efficient to implement on modern CPUs and GPUs. As such, relatively simple latent distributions are typically used, for example a multivariate Gaussian distribution transformed by a small neural network [2]. However, the use of simple latent distributions with finite-capacity neural network architectures can be a limiting factor for modeling complex datasets [10]. As an example, many quantum processes cannot be efficiently simulated using classical (i.e. non-quantum) computational methods [11]. Learning the data distribution arising from such a process is thus likely to be a challenging task for classical generative models using a simple latent distribution.

Quantum computers have been proposed as a way to broaden the range of latent distributions that can be efficiently produced [12]. This is motivated by recent progress in quantum computing technology, with several quantum computers now competitive against classical supercomputers [13, 14] for some distribution sampling tasks [15]. Using a quantum computer to produce samples from a non-classical latent distribution has yielded promising empirical results on a wide range of models and datasets [12, 16–20]. However, the lack of a theoretical understanding of how a quantum distribution can affect model performance and the limited benchmarking performed to date have made it difficult to evaluate this approach.

In this work, we develop a deeper understanding of the use of quantum distributions in the latent space, referred to as "quantum latent distributions", and perform extensive experiments to benchmark this approach. We theoretically investigate the relationship between the complexity class of the latent distribution and that of the generated distribution. We prove that with a quantum latent distribution some types of neural network architectures can produce a wider range of distributions in the data space than could be achieved with classical latent distributions. This can allow a model to achieve improved performance on some datasets. We develop an intuition for why and when this improved performance can be achieved in practice. Based on this theory, we select an experimental setting in which we benchmark a range

* wclements@orcacomputing.com

of latent distributions and GAN models. We find that a quantum latent distribution can indeed lead to improved generator performance compared to a relevant range of classical distributions. We then propose avenues for extending the use of a quantum latent distributions beyond GANs to diffusion and flow matching models.

Our main contributions are as follows:

- We show that, under some assumptions on the generative model, using a quantum distribution in the latent space can broaden the range of output distributions and lead to improved performance on some datasets.
- We benchmark this approach on both a toy quantum dataset and on the QM9 quantum chemistry dataset, and show that a quantum latent distribution in a GAN can improve model performance compared to a range of relevant classical distributions.
- We identify families of deep generative models beyond GANs that are compatible with the use of a quantum latent distribution.

While the benefits of using a quantum latent distribution are dataset and model dependent, our results validate the notion that these distributions can be a useful tool to improve the performance of deep generative models.

II. BACKGROUND AND RELATED WORK

The impact of the latent distribution on the performance of a generative model has been a significant topic of investigation. The choice of latent distribution has been empirically found to have a significant impact on model performance in different types of architectures, including GANs [2, 9, 21–25], VAEs [8, 26], and flow matching models [27]. Some general arguments have been put forward to explain why, for example, uncorrelated latent distributions are not well suited for capturing correlated features in complex data [2]. However, developing a more general theoretical understanding of the relationship between the latent distribution and the data distribution has been challenging. To address this issue, [10] introduced a novel distance between these two spaces, which is minimized when the latent most effectively capitalizes on the capacity of the generator neural network. They harness this insight to increase model performance using a two-stage decoupled autoencoder (DAE) architecture that shapes the latent distribution. In our work, we extend this formalism to develop an understanding of the impact of the computational complexity of the latent distribution on model performance.

There have been several previous investigations of quantum latent distributions in generative models with a focus on GANs, in which empirical performance improvements were observed. However, much of this work focused on demonstrating feasibility, with a smaller emphasis on benchmarking or on developing theoretical foundations for this approach. In [28] a quantum latent distribution is used within a generative algorithm to propose new molecules for cancer therapy, and in [19] this approach is used for image reconstruction. In [17, 18] and [12, 16], small-scale quantum latent distributions in GANs are found to outperform a classical latent distribution on the QM9 and MNIST datasets. However, in those works the use of a trained quantum distribution compared to an untrained classical baseline makes it difficult to draw generalizable conclusions. [20] empirically show an improvement in the BigGAN model [23] using a range of untrained quantum latent distributions, but do not benchmark against other classical latent distributions or provide theoretical foundations for this approach. We go beyond this prior work both by developing a theoretical understanding of the use of quantum distributions, and by benchmarking against a range of classical distributions in an apples-to-apples experimental setting.

A. Quantum circuit sampling

Some classes of discrete distribution sampling tasks can be performed faster on near-term quantum computers than on classical computers [15]. Two specific realizations of quantum computers that can perform such tasks are boson sampling [11, 29] and random circuit sampling [13, 30]. In boson sampling, identical single photons are sent into a random interference circuit, and measurements of the output locations of these photons provide samples from a classically-intractable probability distribution. Similarly, in random circuit sampling, a random sequence of gates is applied to qubits, and the output measurements of the qubit states are also a provably hard-to-sample distribution.

The experiments we perform in this work use photonic boson sampling distributions [11]. Photonic quantum systems have several advantages compared to other quantum computing modalities that motivate using them to generate latent distributions. Though they are not universal for quantum computation, they naturally produce highly non-uniform distributions [31] that differ significantly from commonly used classical latent distributions. Moreover, photonic systems are not subject to decoherence, which pushes the outputs of quantum circuits implemented with other modalities

closer to a classical uniform distribution [20, 32]. The advantages of photonics were leveraged in [33, 34] to perform some of the largest-scale demonstrations of quantum machine learning to date.

III. QUANTUM DISTRIBUTIONS IN GENERATIVE MODELS

In this section we introduce complexity classes for quantum and classical sampling tasks. We establish conditions under which a quantum latent distribution can lead to improved model performance compared to classical distributions. We present theorems without proof, with complete proofs and rigorous definitions found in the appendix.

A. Preliminaries

Sampling problems: We consider the problem of drawing samples from a given probability distribution. We define $\mathcal{C}$ as the class of probability distributions that can be approximately sampled in $\text{Poly}(n, 1/\epsilon)$ time using classical computers. Here, n represents the dimension of the samples to be generated, ϵ the error within which we sample from this distribution. Quantum computers can sample from some distributions that cannot be efficiently sampled from classically [15]. We thus define $\mathcal{Q}$ as the class of distributions that can be sampled in $\text{Poly}(n, 1/\epsilon)$ with a quantum computer, but not with a classical computer. We provide rigorous definitions of these classes in the appendix.

In this section, we consider a family of neural network architectures G mapping from R^{d_z} to R^d , where d_z is the latent space dimension and d is the data space dimension with $d_z \leq d$. We assume that these architectures have bounded complexity, where complexity is loosely defined in terms of width or depth of the network, such that the output can be efficiently produced from the input.

GAN-induced distances: Next, we consider a metric introduced in [10] to quantify the performance of a latent distribution for a generative task. For any distance between distributions $D(.,.)$ in the data space, we define a generalized distance associated with neural network architectures in G as

$$D^G(P_z, P_x) = \inf_{g \in G} D(P_{g(z)}, P_x) \quad (1)$$

where P_z, P_x denote the probability density of distributions defined over the latent space and the data space, respectively, and $P_{g(z)}$ is the pushforward distribution. This metric is particularly well suited for GAN architectures, since it can be viewed as a discriminator loss, and also introduces a notion of similarity between distributions in different dimensions. Informally, a latent distribution P_z is best suited for generating a data distribution P_x using generators within G if it minimizes this distance.

B. Computational complexity of the latent distribution

Our first contribution is making a connection between the complexity class of the latent distribution and the complexity class of the output distribution of a deep generative model. We start with a simple observation about the complexity of pushforward distributions.

Remark 1. Consider a latent distribution $P_z \in \mathcal{C}$ and a neural network $g \in G$ that is Lipschitz-continuous. Then the pushforward distribution $P_{g(z)} \in \mathcal{C}$.

Conversely, if a latent distribution $P_z \in \mathcal{Q}$, then the pushforward distribution is not necessarily in $\mathcal{Q}$. There are scenarios in which an initial quantum distribution is converted to a classical distribution, for example if the neural network is trained to produce constant outputs. A relevant question is whether this collapse from $\mathcal{Q}$ to $\mathcal{C}$ is expected to be a common occurrence or the exception.

Here, we derive sufficient conditions under which $P_{g(z)}$ cannot be efficiently sampled from classically:

Theorem 1. We consider a neural network $g \in G$ such that its inverse g^{-1} exists, is efficiently classically implementable and is also Lipschitz continuous. Let P_z be in $\mathcal{Q}$. Then the pushforward distribution $P_{g(z)}$ is not in $\mathcal{C}$.

Proof sketch: With these assumptions on the generator, we can show that if the pushforward distribution is in $\mathcal{C}$, then by inverting g the latent distribution can also be efficiently sampled and would therefore be in $\mathcal{C}$.

Though these assumptions may appear restrictive, they apply to some types of deep neural network architectures. As an example, consider a deep feedforward neural network mapping a small-dimensional latent space to a larger-dimensional data space through layers of increasing width with a LeakyReLU activation. Since by assumption no linear

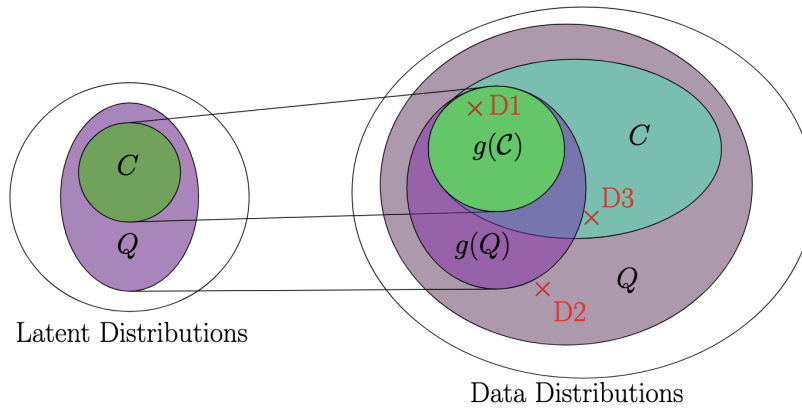


FIG. 1: An illustration of the relation between the complexity of latent distributions and that of the pushforward distributions. $\mathcal{C}$ represents classical distributions, $\mathcal{Q}$ represents non-classical distributions, and $g(\mathcal{Q})$ and $g(\mathcal{C})$ are the pushforward distributions achieved using a class of neural networks g with finite complexity. When the target dataset is in the pushforward distribution of $\mathcal{C}$ (D1), a quantum distribution is not expected to provide a benefit. However, when the target dataset is not in the pushforward distribution of $\mathcal{C}$ and can be either quantum (D2) or classical (D3) the pushforward distribution of a quantum latent distribution may be closer, leading to better performance.

layer decreases the size of the inputs, the input of each linear layer can in general be efficiently reconstructed from the outputs by inverting a set of linear equations. The use of LeakyReLU makes the entire network invertible. Moreover, feedforward neural networks are typically Lipschitz-continuous [35]. This architecture thus satisfies the assumptions. In our benchmarking on the QM9 dataset, we adopt a model architecture that closely matches this example.

Beyond this simple example, we note that other types of neural networks such as convolutional neural networks are also generally Lipschitz continuous [35]. A notable exception is dot product self-attention, though alternatives that are Lipschitz continuous have been developed [36]. Moreover, the Lipschitz constant of a neural network is sometimes used as a proxy for model complexity [37]. Since we consider models with bounded complexity, the Lipschitz continuity assumption is thus often satisfied in practice for a wide range of architectures.

Even when a neural network does not strictly satisfy the conditions above, the output distribution may still be non-classical for a range of possible input quantum distributions. Theorem 1 provides a theoretical basis for suggesting that many modern neural network architectures can in practice use latent distributions within $\mathcal{Q}$ to produce a broader set of distributions than would be possible only with distributions within $\mathcal{C}$. This is illustrated in figure 1.

C. Impact on model performance

In this section, we discuss the implications to model performance, defined in terms of GAN-induced distance, through the use of a quantum latent distribution. First, we show that at least some data distributions can be better approximated using a quantum latent distribution:

Corollary 1. *We consider a neural network g satisfying the conditions of Theorem 1, and a quantum latent distribution $P_{z_Q} \in \mathcal{Q}$. Using the Wasserstein distance as our metric, we have both $D^G(P_{z_Q}, P_{g(z_Q)}) = 0$ by definition, and for any classical latent distribution $P_{z_C} \in \mathcal{C}$ there exists $\epsilon > 0$ such that $D^G(P_{z_C}, P_{g(z_Q)}) > \epsilon$.*

This shows that the class of pushforward distributions achieved by a quantum latent distribution under certain conditions cannot be approximated with any choice of classical latent distribution. Intuitively, the corollary demonstrates that there exist classes of generative tasks for which the performance we can achieve with classical latent distributions will be strictly worse than that which can be achieved with quantum latent distributions. This may include datasets that result from a classical transformation of an underlying quantum process.

We can now reason in terms of performance on different types of datasets. For a target data distribution that is in $\mathcal{C}$, we can distinguish between two cases illustrated in figure 1. First, if this distribution is in the pushforward distribution of classical latents that are also in $\mathcal{C}$, then there is no theoretical advantage to using a quantum latent. In the second case, the target distribution is not in the pushforward distribution of classical latents. Since we consider generators with finite capacity and latent spaces that are smaller than the data space, this may be a common scenario. In that case, some achievable quantum pushforward distributions will be closer than any achievable classical pushforward distributions in terms of GAN-induced distance. As such, a quantum latent distribution may lead to an advantage even for a classical data distribution.

D. Achieving an advantage in practice

The results above establish a theoretical basis for the use of quantum distributions in generative models, but the conditions described above for achieving a performance advantage over classical distributions are still highly abstract. Here, we propose two mechanisms through which quantum distributions can be helpful, which can provide practical guidance as to when to use such a distribution.

First, the use of a quantum latent distribution can endow a model with a beneficial "inductive bias" [38] for certain types of dataset. A quantum latent distribution can be expected to provide such a bias for a dataset arising from a fundamental physical process that is more efficiently described by quantum mechanics than by classical mechanics. Moreover, as suggested by corollary 1 the performance of models with classical latent distributions is inherently limited for such datasets. This leaves a broader scope for a performance improvement with a quantum distribution.

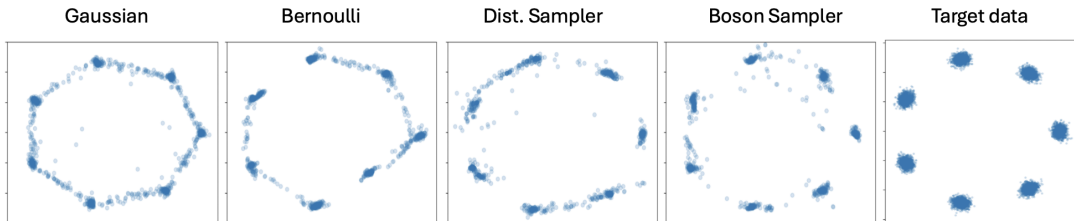


FIG. 2: Comparison between GANs trained on a 2D mixture of Gaussians dataset with different latent distributions.

The four latent distributions represent a progression from the commonly used Gaussian distribution (left) to a quantum distribution (boson sampler, right) with more information provided in section 4. The main failure mode is a tendency to interpolate between different modes, and the model with the quantum latent distribution is least affected. This could be explained by the fact that quantum distributions cannot be tractably factorized into independent factors of variation that could encourage smooth interpolations.

Second, a quantum latent distribution prevents a model from learning a fully factorized representation of the data, which we hypothesize can be helpful for multimodal datasets. A quantum distribution that is classically hard to simulate cannot be efficiently factorized into independent factors of variation. Though factorizable representations are often viewed positively since they are more interpretable, empirical evidence for other benefits are mixed [39, 40]. Factorized representations can perform poorly on multimodal datasets without additional modifications [41]. Though factorized representations have been shown to facilitate smooth interpolation in the latent space [42], when the data is multimodal a representation that encourages interpolation between the modes may be detrimental. To further investigate, we trained GANs with different latent distributions on a simple multimodal 2D mixture-of-Gaussians dataset. Our results are shown in figure 2 and support this hypothesis, where we observe that the quantum distribution produces the model that interpolates the least between the modes. Further empirical evidence for improved data coverage from quantum latent distributions was also observed in [20].

Taken together, these mechanisms provide guidelines for when quantum latent distributions may provide a practical advantage. Though we do not expect them to provide a benefit for all models and datasets (with negative examples provided in appendix H), quantum latent distributions may be well suited for datasets where the underlying data generating process is poorly described by classical mechanics, or datasets that are highly multimodal. To empirically verify this intuition, in the next section we perform experiments focusing on datasets arising from physics and chemistry, where we compare the performance of different latent distributions on several metrics including data diversity.

IV. BENCHMARKING GANS WITH QUANTUM LATENT DISTRIBUTIONS

In this section, we empirically investigate whether the use of a quantum latent distribution can lead to a performance improvement over relevant classical distributions. We consider generative adversarial networks (GANs), where the use of different latent distributions is well-studied. We perform two experiments: one on a small-scale toy dataset produced by a quantum process, and one on the QM9 quantum chemistry dataset. For both experiments, as illustrated in Figure 3 we compare different types of latent space distributions that produce samples z of length L ($z_{dim} = L$). We select three classical latent distributions and one quantum distribution:

- **Gaussian:** the commonly used Gaussian distribution (continuous), $z \in \mathbb{R}^L \sim \mathcal{N}(0, I)$
- **Bernoulli:** random uniform bit strings (discrete), $z \in \{0, 1\}^L$

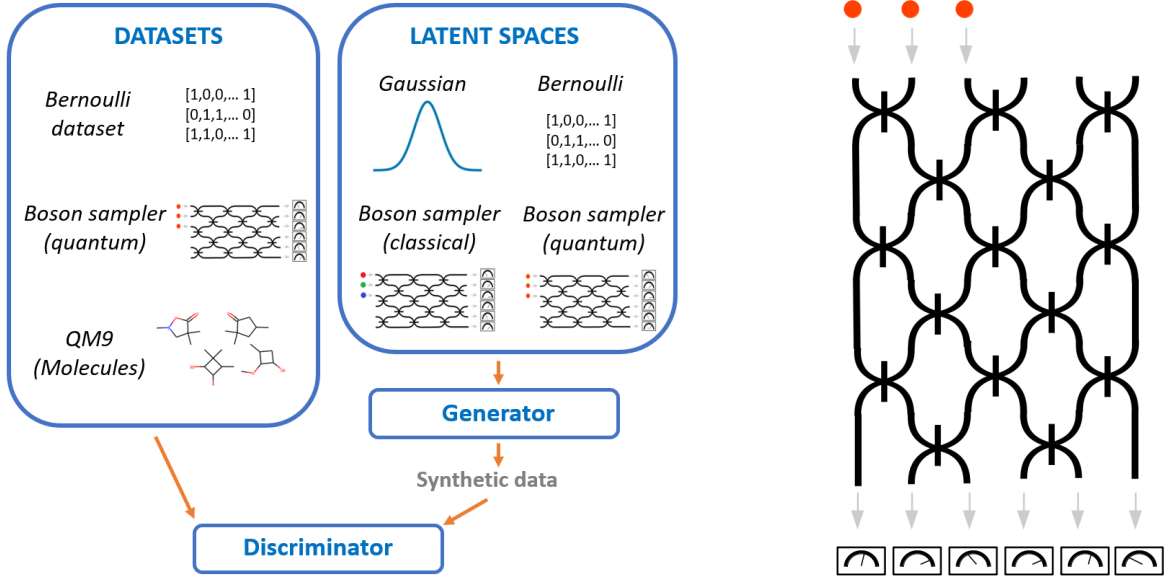


FIG. 3: Left: We compare GANs that use different latent space distributions on different datasets. We focus on quantum latent distributions generated by boson samplers, a type of quantum processor. Right: in a boson sampler, several identical photons enter an optical interference network and a measurement is performed to determine where the photons leave the network. Classical simulations of this probabilistic process are inefficient, and intractable beyond a few tens of photons.

- **Dist. sampler:** arising from distinguishable photons sent into a size- L interferometer
- **Boson sampler:** arising from indistinguishable photons sent into a size- L interferometer

We select the Gaussian distribution since this is the most commonly used classical latent distribution. The other two distributions represent two intermediate steps that bring us closer to the quantum distribution. The Bernoulli distribution is a discrete rather than continuous latent distribution, which yielded good performance in [23]. The distribution arising from sending distinguishable single photons, which do not interfere with each other (for example photons with different frequencies), into a size- L interferometer is a discrete distribution that can be produced efficiently classically. The difference between the distinguishable and indistinguishable photon distributions is due to quantum interference between photons when they are indistinguishable, such that a difference in performance between these two latent distributions can be attributed to the different statistics arising from this interference.

To ensure an apples-to-apples comparison, for all experiments our models differ only through the nature of the process that produced the samples that are sent to the generator. Moreover, both types of photonic distributions arise from randomly initialized interference circuits. To ensure that performance differences arise from general properties of these distributions and not from specific circuit realizations, these circuits are re-sampled for every seed (with an exception for the largest-sized latent spaces, see appendix for details). We note that these circuits could additionally be trained [43], although training methods for quantum circuits often scale poorly [44]. We do not train them in this work since this could lead to an unfair advantage in benchmarking against the static Gaussian and Bernoulli distributions.

A. Experiment on a toy dataset

For our small-scale experiment, we consider two toy datasets: a "Quantum dataset" produced by simulating [45] 8 identical photons interfering in a 16-channel random optical circuit, and a "Bernoulli dataset" produced by a 16-dimensional Bernoulli distribution. The optical circuits used for the latent and data distributions are independently sampled, preventing trivial identity mapping. Since the two target distributions are discrete, to quantify the performance of a trained model we use the average distance between the outputs of the generator and their nearest integers. A smaller distance indicates that the generator has been more successful in learning the discrete nature of the distribution.

Our results comparing the performance of the trained models are shown in Table I. A first observation is that the classical dataset is easier to learn, with smaller distances and smaller differences between the latent distributions. On the harder quantum dataset, we observe significant differences between the latent distributions, with the quantum

TABLE I: L1 distances between the numbers generated by the GANs and their closest integers. Each column corresponds to a different latent distribution, and each row corresponds to a different dataset. The error bounds represent the uncertainty of the mean, estimated over 12 runs.

	Gaussian	Bernoulli	Dist. sampler	Boson sampler
Quantum dataset	0.061 ± 0.001	0.065 ± 0.001	0.041 ± 0.002	0.036 ± 0.001
Bernoulli dataset	0.012 ± 0.002	0.020 ± 0.013	0.017 ± 0.002	0.015 ± 0.002

latent distribution achieving the best results. In particular, the quantum latent distribution outperforms the latent distribution that uses distinguishable photons. This implies that the quantum interference between photons, which is the only difference between these two distributions, is a useful resource that the model used to achieve better performance on the quantum dataset. These results validate the expectation that a quantum distribution can be a more suitable latent distribution when the data also arises from a quantum process.

B. Experiments on the QM9 dataset

Next, we consider the QM9 quantum chemistry dataset [46]. Generating chemical structures with specific properties is a challenging problem, partly because the underlying dynamics are described by quantum mechanical principles. As such, quantum chemistry is a promising field for realizing the potential advantages of a quantum latent distribution. The QM9 dataset benefits from being a reasonably small-size dataset that is amenable to extensive benchmarking, and which has been previously studied within the context of GANs [47–49]. We use a GAN architecture based on MolGAN [47], with some improvements described in the appendix. The generator is a feedforward neural network which is loosely based on the requirements of theorem 1: all layers except the final layer are non-decreasing, and we use a LeakyReLU activation function. The discriminator is a relational graph convolutional network.

We perform a wide range of experiments, with 20 random seeds per experiment, in which we test different combinations and sizes of latent distributions. These experiments allow us to determine how different latent distributions can affect performance. To quantify performance, we use the Frechet Chemical Distance (FCD) [50], the number of valid and unique molecules generated by the model from 10k generation attempts, and the number of molecules among this set which are novel and not in the training set. A successful model has low FCD and generates high numbers of valid and unique molecules that are not in the training set. We note that all these metrics correlate with the diversity of the generated data.

1. Type and size of the latent distribution

We first compare results over different latent distributions and different sizes: 16, 32 and 48. For the distinguishable and indistinguishable photon distributions, we consider unstructured optical circuits, with transfer matrices described by Haar-random unitary matrices, and with half as many input photons as optical channels. For example, the size-16 latent is produced by 8 photons in 16 channels. We use the algorithm in [45] to simulate these systems.

Our results are shown in table II, with additional results such as training curves in the appendix. We find that, for all 3 latent space sizes, the quantum distribution outperforms the classical distributions. The largest performance improvement occurs for the size-16 latent space, where the quantum distribution outperforms all other latent distributions on all metrics. Though the gap narrows for larger latent spaces, and overall performance decreases, the size-48 quantum latent distribution still leads to a higher number of valid and unique generated molecules than the classical baselines. Moreover, as shown by the improvement over distinguishable photons, the statistics arising from quantum interference is a factor that improves performance. These results show that a quantum latent distribution can outperform other relevant classical latent distributions on a chemistry data set.

These experiments were limited to size-48 latents due to the exponentially increasing complexity of simulating a quantum processor. Generating 1 million size-48 samples with the algorithm in [45] took about 3.75 hours on 100 AMD EPYC™ 7003 processors, which may be close to a regime where a near-term quantum processor would run faster. The persistence of a small advantage at size 48 on the number of valid, unique and valid molecules suggests that an advantage over the classical distributions may still be achievable for a quantum processor of a size that is not practically simulable. Though in this case a smaller latent space yields improved results, for other models and datasets larger-scale latents have been shown to work best [23], which would motivate further exploration of this approach for larger-scale quantum distributions.

TABLE II: Metrics ($\pm$ standard error over 20 seeds) for different latents with different dimensions

Latent Type	z_{dim}	FCD $\downarrow$	# Valid & unique $\uparrow$	# Novel $\uparrow$
Boson samp.	16	1.160 $\pm$ 0.06	2522 $\pm$ 65	1331 $\pm$ 37
Dist. samp.	16	1.333 $\pm$ 0.07	1954 $\pm$ 103	1067 $\pm$ 54
Bernoulli	16	1.822 $\pm$ 0.09	1244 $\pm$ 102	702 $\pm$ 56
Gaussian	16	1.529 $\pm$ 0.08	1814 $\pm$ 115	1017 $\pm$ 64
Boson samp.	32	1.536 $\pm$ 0.08	1791 $\pm$ 106	951 $\pm$ 37
Dist. samp.	32	1.533 $\pm$ 0.06	1633 $\pm$ 81	919 $\pm$ 45
Bernoulli	32	1.646 $\pm$ 0.09	1391 $\pm$ 77	790 $\pm$ 44
Gaussian	32	1.823 $\pm$ 0.07	1320 $\pm$ 53	768 $\pm$ 35
Boson samp.	48	1.696 $\pm$ 0.08	1528 $\pm$ 65	856 $\pm$ 40
Dist. samp.	48	1.713 $\pm$ 0.06	1307 $\pm$ 77	746 $\pm$ 43
Bernoulli	48	1.671 $\pm$ 0.08	1456 $\pm$ 82	816 $\pm$ 48
Gaussian	48	1.781 $\pm$ 0.07	1352 $\pm$ 58	766 $\pm$ 30

2. Type of quantum circuit

To understand whether the results presented in the previous section are specific to Haar-random optical circuits or are due to more general properties of the statistics of photon interference, we also ran experiments with different types of optical circuits. We simulate two realizations of a boson sampling circuit based on optical delay lines, which have been proposed and demonstrated as an experimentally feasible route to large-scale boson sampling [14, 51, 52]. These realizations respectively have two delay lines of the same length (a "1-1" configuration), and three delay lines in a "1-3-9" configuration where each line delays each photon by three times the previous line.

TABLE III: Metrics ($\pm$ standard error over 20 seeds) for different latents using different quantum circuits

Latent Type	z_{dim}	FCD $\downarrow$	# Valid & unique $\uparrow$	# Novel $\uparrow$
Boson samp. 1-1	16	1.245 $\pm$ 0.06	1375 $\pm$ 72	692 $\pm$ 34
Dist. samp. 1-1	16	1.291 $\pm$ 0.07	1182 $\pm$ 96	580 $\pm$ 47
Boson samp. 1-3-9	16	1.187 $\pm$ 0.06	1990 $\pm$ 106	1063 $\pm$ 58
Dist. samp. 1-3-9	16	1.376 $\pm$ 0.09	1459 $\pm$ 100	775 $\pm$ 53

Our results are shown in Table III. We find that for each type of optical circuit, the quantum distribution outperforms the classical distribution arising from distinguishable photons. This shows that for this model and dataset the statistics arising from quantum interference confer an advantage that does not depend on the specifics of the circuit architecture. We also find that the circuit architecture has an impact on the overall performance of the latent distribution, with the Haar-random circuit performing best overall for size-16 latents. This suggests that developing methods for circuit architecture search or joint training of the circuit with the neural network could lead to improved results.

3. Experiments with a real quantum processor

Realistic boson sampling systems are subject to a range of imperfections such as optical loss, limited photon number resolution, and imperfect indistinguishability. To assess the extent to which our previous results are transferrable to a real quantum processor, we performed experiments on a real boson sampling system, a commercially available ORCA Computing PT-2. This system can interfere 16 photons in 32 channels and its optical network consists of optical delay lines in the "1-1" configuration. More information about this system can be found in the appendix. We collected half a million samples from this system with a single circuit realization, and compared the results to comparable size-32 latent distributions.

Our results are shown in Table IV. We find that the results from the real quantum processor closely match the simulated quantum processor despite experimental imperfections. Both quantum distributions also outperform the classical distributions. These results indicate that a performance advantage achieved with a simulated quantum system can provide a good indication of the performance of a real quantum processor, which can be helpful for extrapolating the performance of small-scale simulable distributions to larger-scale, non-simulable distributions.

TABLE IV: Metrics ($\pm$ standard error over 20 seeds) for a real quantum device and simulated latents

Latent Type	z_{dim}	FCD $\downarrow$	# Valid & unique $\uparrow$	# Novel $\uparrow$
Boson samp. 1-1 (real)	32	1.410 $\pm$ 0.07	1962 $\pm$ 75	1084 $\pm$ 45
Boson samp. 1-1 (simulated)	32	1.414 $\pm$ 0.07	1903 $\pm$ 66	1073 $\pm$ 36
Dist. samp. 1-1	32	1.560 $\pm$ 0.09	1638 $\pm$ 102	900 $\pm$ 50
Bernoulli	32	1.646 $\pm$ 0.09	1391 $\pm$ 77	790 $\pm$ 44
Gaussian	32	1.823 $\pm$ 0.07	1320 $\pm$ 53	768 $\pm$ 35

V. QUANTUM DISTRIBUTIONS FOR DIFFUSION AND FLOW MATCHING

In contrast to GANs where the latent distribution can be straightforwardly swapped out for a different distribution, the specifics of how such a process can be carried out for other models are less clear. This has restricted most studies of quantum latent distributions in deep generative models only to GANs. Here, we broaden the set of models that can make use of this method. Though the experiments in this section do not exhibit a performance advantage with a quantum latent distribution, they demonstrate the feasibility of this approach and motivate further exploration.

A. Denoising Diffusion GANs

We first investigate diffusion models, which often achieve superior performance to GANs in fields such as images [53] and protein structure generation [54]. Latent diffusion models generally make Gaussianity assumptions on the latent distribution [3], such that the use of a quantum distribution is not straightforward. However, it has been observed that the reverse process, when considered over large time steps, involves modeling a complex non-Gaussian distribution, for which a GAN can be well suited [55]. We thus focus on Denoising Diffusion GANs (DDGANs) [55], which use the standard forward diffusion process but use a GAN to perform the reverse process. They achieve results comparable to standard diffusion models with faster inference speeds, with a model that is easier to train than other GANs. This work aligns with the hypothesis [10] that at least part of the observed improvement in diffusion models over GANs may stem from the additional computation at inference due to the recursive sampling process rather than a fundamental paradigm difference.

We investigated unconditional generation on the CIFAR10 dataset [56] for different latent distributions, using the model architecture in [55]. Where the original model used a size-256 latent space, our experiments used a size-48 latent space due to the classical complexity of simulating larger quantum circuits. We found results comparable to those in [55]. This experiment shows the feasibility of this approach and motivates future work to identify datasets and metrics where a quantum latent distribution can improve results.

B. Flow matching

We also identify flow matching models as being well suited to use a quantum distribution in the latent space. Flow matching has emerged as a flexible tool for producing high-quality data with potentially fewer inference steps than diffusion models. Though many such models require a source distribution that is Gaussian, recently flow matching models that can use an arbitrary source distribution have been proposed [5, 6]. To illustrate the feasibility of a quantum latent distribution as a source distribution, Figure 4 represents a small-scale model trained using the method from [27] to map 2D-projected quantum latent vectors to the "moons" dataset. Further work would be required to identify models and datasets on which a quantum distribution can provide a benefit.

VI. DISCUSSION

Here, we discuss two potential limitations of using a quantum latent distribution for a generative model. These are the extent to which other classical distributions could be used instead of the quantum distribution to achieve improved performance, and whether the quantum computer's output could be classically simulated to achieve the same result.

First, though we compared the quantum distribution to a range of progressively more similar classical distributions, we did not perform an exhaustive search of different classical latent distributions. More complex classical distributions could be used, for example trained distributions using an autoencoder at the expense of additional training [10]. Some

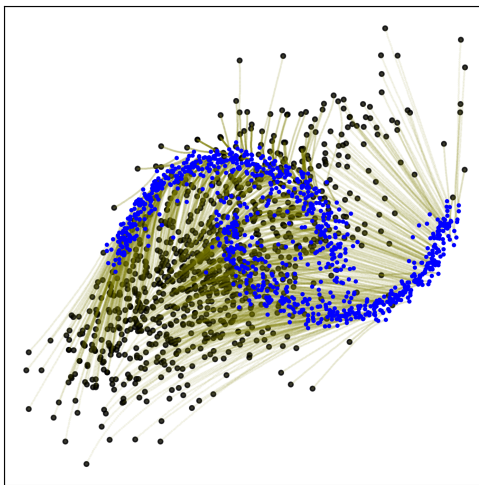


FIG. 4: A flow matching model trained to map a set of quantum latent vectors projected onto a 2D plane (black) to the "moons" dataset (blue)

of these methods may yield improved performance on this problem. However, investigating a wide range of alternative classical distributions is impractical and computationally expensive, while the increasing availability of quantum computers [57] means that they are progressively becoming a practical alternative. Furthermore, this work provides a principled approach for identifying the settings that are most likely to benefit from a quantum distribution, namely datasets originating from a physical process that is poorly described by classical mechanics and highly multimodal datasets. As such, in these settings the use of a quantum computer may be a more effective use of resources than investigating a wider range of classical distributions.

Additionally, though our comparison between distinguishable and indistinguishable photons allows us to show that the statistics arising from quantum interference can improve performance, other classical distributions may be able to reproduce similar statistics without a quantum processor. In particular, classical methods for simulating photonic quantum processors in the presence of noise have been developed [58, 59]. These methods may be able to achieve a similar performance improvement, however for producing latent distributions they are likely to be impractical. The approach of [58] reproduces the low-order statistics of a boson sampling distribution with a quadratic overhead compared to a linear overhead at most with a real physical system. The alternative approach of [59] required 9 minutes on 288 GPUs to set up the simulation initially. Neither approach is thus likely to run faster than a real quantum processor. Though the speed of generating samples is less of a concern for the apples-to-apples benchmarking setting that we selected in this work where the latent distribution is not trained, this can be a major limitation for more advanced variational settings. Therefore, if the latent distribution were to be trained jointly with the neural network, for example using the method in [43], these existing classical approximation methods would not be suitable.

VII. CONCLUSION

In this work, we examined the use of quantum distributions in the latent space of deep generative models, providing both theoretical and empirical arguments for a potential benefit. We showed that under some assumptions on the generator architecture, a quantum latent distribution can allow a model to produce a broader class of data distributions. Through benchmarks on several datasets, we empirically found that quantum latent distributions can indeed outperform relevant classical distributions.

Though these results establish that near-term quantum processors can help expand the capabilities of generative models, several questions remain to be addressed and could be the focus of future work. First, though we propose two mechanisms through which a quantum latent distribution could be used to achieve a practical performance improvement, a deeper theoretical understanding of the inductive bias that these distributions provide and how the non-factorizability of the latent distribution affects the outputs would be beneficial. Second, we anticipate that, in the same way that trained classical latent distributions can lead to improved results [10], developing methods to train the quantum processor jointly with the generator would also yield better performance. Finally, extending our experiments on diffusion and flow matching to larger-scale models and different data modalities would help evaluate the potential of this approach to improve the current state of the art on challenging real-world problems.

VIII. AUTHOR CONTRIBUTIONS AND ACKNOWLEDGEMENTS

OB, TF and HW performed the experiments reported in this paper. AM developed the theory. SSL, JP and RJAFJ operated the ORCA Computing PT-2 and retrieved the samples used for our experiments. WRC conceived the project, and both TH and WRC supervised it. OB, TF, AM, HW, TH and WRC contributed to writing the manuscript.

We thank the team at ORCA Computing for their comments and support. This work was supported by the Quantum Data Centre of the Future, project number 10004793, funded by the Innovate UK Industrial Strategy Challenge Fund (ISCF).

-
- [1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in neural information processing systems*, vol. 27, 2014.
 - [2] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.
 - [3] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.
 - [4] A. Blattmann, R. Rombach, H. Ling, T. Dockhorn, S. W. Kim, S. Fidler, and K. Kreis, “Align your latents: High-resolution video synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22563–22575, 2023.
 - [5] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le, “Flow matching for generative modeling,” *arXiv preprint arXiv:2210.02747*, 2022.
 - [6] M. S. Albergo and E. Vanden-Eijnden, “Building normalizing flows with stochastic interpolants,” *arXiv preprint arXiv:2209.15571*, 2022.
 - [7] D. P. Kingma and M. Welling, “Auto-encoding variational Bayes,” 2022.
 - [8] A. Van Den Oord, O. Vinyals, *et al.*, “Neural discrete representation learning,” *Advances in neural information processing systems*, vol. 30, 2017.
 - [9] T. Arici and A. Celikyilmaz, “Associative adversarial networks,” *NeurIPS 2016 Workshop on Adversarial Training*, 2016.
 - [10] T. Hu, F. Chen, H. Wang, J. Li, W. Wang, J. Sun, and Z. Li, “Complexity matters: Rethinking the latent space for generative modeling,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 29558–29579, 2023.
 - [11] S. Aaronson and A. Arkhipov, “The computational complexity of linear optics,” in *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pp. 333–342, 2011.
 - [12] M. S. Rudolph, N. B. Toussaint, A. Katarbwa, S. Johri, B. Peropadre, and A. Perdomo-Ortiz, “Generation of high-resolution handwritten digits with an ion-trap quantum computer,” *Physical Review X*, vol. 12, no. 3, p. 031010, 2022.
 - [13] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, *et al.*, “Quantum supremacy using a programmable superconducting processor,” *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.
 - [14] L. S. Madsen, F. Laudenbach, M. F. Askarani, F. Rortais, T. Vincent, J. F. Bulmer, F. M. Miatto, L. Neuhaus, L. G. Helt, M. J. Collins, *et al.*, “Quantum computational advantage with a programmable photonic processor,” *Nature*, vol. 606, no. 7912, pp. 75–81, 2022.
 - [15] D. Hangleiter and J. Eisert, “Computational advantage of quantum random sampling,” *arXiv preprint arXiv:2206.04079*, 2022.
 - [16] M. Wilson, T. Vandal, T. Hogg, and E. G. Rieffel, “Quantum-assisted associative adversarial network: Applying quantum annealing in deep learning,” *Quantum Machine Intelligence*, vol. 3, no. 1, p. 19, 2021.
 - [17] J. Li, R. O. Topaloglu, and S. Ghosh, “Quantum generative models for small molecule drug discovery,” *IEEE transactions on quantum engineering*, vol. 2, pp. 1–8, 2021.
 - [18] P.-Y. Kao, Y.-C. Yang, W.-Y. Chiang, J.-Y. Hsiao, Y. Cao, A. Aliper, F. Ren, A. Aspuru-Guzik, A. Zhavoronkov, M.-H. Hsieh, *et al.*, “Exploring the advantages of quantum generative adversarial networks in generative chemistry,” *Journal of Chemical Information and Modeling*, vol. 63, no. 11, pp. 3307–3318, 2023.
 - [19] T. Xiao, X. Zhai, J. Huang, J. Fan, and G. Zeng, “Quantum deep generative prior with programmable quantum circuits,” *Communications Physics*, vol. 7, no. 1, p. 276, 2024.
 - [20] H. Jin and K. M. Merz Jr, “Improving gans by leveraging the quantum noise from real hardware,” *arXiv preprint arXiv:2507.01886*, 2025.
 - [21] M. Ben-Yosef and D. Weinshall, “Gaussian mixture generative adversarial networks for diverse datasets, and the unsupervised clustering of images,” *arXiv preprint arXiv:1808.10356*, 2018.
 - [22] S. Mukherjee, H. Asnani, E. Lin, and S. Kannan, “Clustergan: Latent space clustering in generative adversarial networks,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 4610–4617, 2019.
 - [23] A. Brock, J. Donahue, and K. Simonyan, “Large scale GAN training for high fidelity natural image synthesis,” in *International Conference on Learning Representations*, 2019.
 - [24] C. Xiao, P. Zhong, and C. Zheng, “BourGAN: Generative networks with metric embeddings,” *Advances in neural information processing systems*, vol. 31, 2018.
 - [25] G. Luise, M. Pontil, and C. Ciliberto, “Generalization properties of optimal transport GANs with latent distribution

- learning,” *arXiv preprint arXiv:2007.14641*, 2020.
- [26] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, “Adversarial autoencoders,” *arXiv preprint arXiv:1511.05644*, 2015.
 - [27] A. Tong, K. Fatras, N. Malkin, G. Huguet, Y. Zhang, J. Rector-Brooks, G. Wolf, and Y. Bengio, “Improving and generalizing flow-based generative models with minibatch optimal transport,” *arXiv preprint arXiv:2302.00482*, 2023.
 - [28] M. G. Vakili, C. Gorgulla, J. Snider, A. Nigam, D. Bezrukov, D. Varoli, A. Aliper, D. Polykovsky, K. M. P. Das, H. C. Iii, *et al.*, “Quantum-computing-enhanced algorithm unveils potential KRAS inhibitors,” *Nature biotechnology*, 2025.
 - [29] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu, *et al.*, “Quantum computational advantage using photons,” *Science*, vol. 370, no. 6523, pp. 1460–1463, 2020.
 - [30] R. Movassagh, “The hardness of random quantum circuits,” *Nature Physics*, vol. 19, no. 11, pp. 1719–1724, 2023.
 - [31] S. Aaronson and A. Arkhipov, “BosonSampling is far from uniform,” *arXiv preprint arXiv:1309.7460*, 2013.
 - [32] S. Wang, E. Fontana, M. Cerezo, K. Sharma, A. Sone, L. Cincio, and P. J. Coles, “Noise-induced barren plateaus in variational quantum algorithms,” *Nature communications*, vol. 12, no. 1, p. 6961, 2021.
 - [33] V. Cimini, M. M. Sohoni, F. Presutti, B. K. Malia, S.-Y. Ma, R. Yanagimoto, T. Wang, T. Onodera, L. G. Wright, and P. L. McMahon, “Large-scale quantum reservoir computing using a gaussian boson sampler,” *arXiv preprint arXiv:2505.13695*, 2025.
 - [34] Z. Yin, I. Agresti, G. de Felice, D. Brown, A. Toumi, C. Pentangelo, S. Piacentini, A. Crespi, F. Ceccarelli, R. Osellame, *et al.*, “Experimental quantum-enhanced kernels on a photonic processor,” *arXiv preprint arXiv:2407.20364*, 2024.
 - [35] A. Virmaux and K. Scaman, “Lipschitz regularity of deep neural networks: analysis and efficient estimation,” *Advances in Neural Information Processing Systems*, vol. 31, 2018.
 - [36] H. Kim, G. Papamakarios, and A. Mnih, “The lipschitz constant of self-attention,” in *International Conference on Machine Learning*, pp. 5562–5571, PMLR, 2021.
 - [37] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky, “Spectrally-normalized margin bounds for neural networks,” *Advances in neural information processing systems*, vol. 30, 2017.
 - [38] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, *et al.*, “Relational inductive biases, deep learning, and graph networks,” *arXiv preprint arXiv:1806.01261*, 2018.
 - [39] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem, “Challenging common assumptions in the unsupervised learning of disentangled representations,” in *international conference on machine learning*, pp. 4114–4124, PMLR, 2019.
 - [40] M. Montero, J. Bowers, R. Ponte Costa, C. Ludwig, and G. Malhotra, “Lost in latent space: Examining failures of disentangled models at combinatorial generalisation,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 10136–10149, 2022.
 - [41] Y.-H. H. Tsai, P. P. Liang, A. Zadeh, L.-P. Morency, and R. Salakhutdinov, “Learning factorized multimodal representations,” in *International Conference on Learning Representations*, 2019.
 - [42] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, “InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets,” *Advances in neural information processing systems*, vol. 29, 2016.
 - [43] G. Facelli, D. D. Roberts, H. Wallner, A. Makarovskiy, Z. Holmes, and W. R. Clements, “Exact gradients for linear optics with single photons,” *arXiv preprint arXiv:2409.16369*, 2024.
 - [44] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, “Barren plateaus in quantum neural network training landscapes,” *Nature communications*, vol. 9, no. 1, p. 4812, 2018.
 - [45] P. Clifford and R. Clifford, “The classical complexity of boson sampling,” in *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 146–155, SIAM, 2018.
 - [46] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. Von Lilienfeld, “Quantum chemistry structures and properties of 134 kilo molecules,” *Scientific data*, vol. 1, no. 1, pp. 1–7, 2014.
 - [47] N. De Cao and T. Kipf, “MolGAN: An implicit generative model for small molecular graphs,” *arXiv preprint arXiv:1805.11973*, 2018.
 - [48] G. L. Guimaraes, B. Sanchez-Lengeling, C. Outeiral, P. L. C. Farias, and A. Aspuru-Guzik, “Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models,” *arXiv preprint arXiv:1705.10843*, 2017.
 - [49] C. Li and Y. Yamanishi, “TenGAN: Pure transformer encoders make an efficient discrete gan for de novo molecular generation,” in *International Conference on Artificial Intelligence and Statistics*, pp. 361–369, PMLR, 2024.
 - [50] K. Preuer, P. Renz, T. Unterthiner, S. Hochreiter, and G. Klambauer, “Fréchet ChemNet distance: a metric for generative models for molecules in drug discovery,” *Journal of chemical information and modeling*, vol. 58, no. 9, pp. 1736–1741, 2018.
 - [51] A. Deshpande, A. Mehta, T. Vincent, N. Quesada, M. Hinsche, M. Ioannou, L. Madsen, J. Lavoie, H. Qi, J. Eisert, *et al.*, “Quantum computational advantage via high-dimensional Gaussian boson sampling,” *Science advances*, vol. 8, no. 1, p. eabi7894, 2022.
 - [52] S. Novák, D. D. Roberts, A. Makarovskiy, R. García-Patrón, and W. R. Clements, “Boundaries for quantum advantage with single photons and loop-based time-bin interferometers,” *arXiv preprint arXiv:2411.16873*, 2024.
 - [53] P. Dhariwal and A. Nichol, “Diffusion models beat GANs on image synthesis,” *Advances in neural information processing systems*, vol. 34, pp. 8780–8794, 2021.
 - [54] M. Baek, F. DiMaio, I. Anishchenko, J. Dauparas, S. Ovchinnikov, G. R. Lee, J. Wang, Q. Cong, L. N. Kinch, R. D. Schaeffer, *et al.*, “Accurate prediction of protein structures and interactions using a three-track neural network,” *Science*, vol. 373, no. 6557, pp. 871–876, 2021.
 - [55] Z. Xiao, K. Kreis, and A. Vahdat, “Tackling the generative learning trilemma with denoising diffusion GANs,” *arXiv preprint*

arXiv:2112.07804, 2021.

- [56] A. Krizhevsky, G. Hinton, *et al.*, “Learning multiple layers of features from tiny images,” 2009.
- [57] M. Slys, P. Rydlichowski, K. Kurowski, O. Bacarezza, E. C. Gomez, Z. Chandani, B. Heim, P. Khalate, W. R. Clements, and J. Fletcher, “Hybrid classical-quantum supercomputing: A demonstration of a multi-user, multi-qpu and multi-gpu environment,” 2025.
- [58] B. Villalonga, M. Y. Niu, L. Li, H. Neven, J. C. Platt, V. N. Smelyanskiy, and S. Boixo, “Efficient approximation of experimental Gaussian boson sampling,” *arXiv preprint arXiv:2109.11525*, 2021.
- [59] C. Oh, M. Liu, Y. Alexeev, B. Fefferman, and L. Jiang, “Classical algorithm for simulating experimental gaussian boson sampling,” *Nature Physics*, vol. 20, no. 9, pp. 1461–1468, 2024.
- [60] W denotes the Wasserstein metric.
- [61] The TVD error we achieve is a constant multiple of ϵ . As such we reach the same conclusions on complexity since $\text{Poly}(n, 1/c\epsilon) \equiv \text{Poly}(n, 1/\epsilon)$ for some constant c .
- [62] J. B. Spring, B. J. Metcalf, P. C. Humphreys, W. S. Kolthammer, X.-M. Jin, M. Barbieri, A. Datta, N. Thomas-Peter, N. K. Langford, D. Kundys, *et al.*, “Boson sampling on a photonic chip,” *Science*, vol. 339, no. 6121, pp. 798–801, 2013.
- [63] H.-L. Liu, H. Su, S.-Q. Gong, Y.-C. Gu, H.-Y. Tang, M.-H. Jia, Q. Wei, Y. Song, D. Wang, M. Zheng, *et al.*, “Robust quantum computational advantage with programmable 3050-photon gaussian boson sampling,” *arXiv preprint arXiv:2508.09092*, 2025.
- [64] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of Wasserstein GANs,” *Advances in neural information processing systems*, vol. 30, 2017.
- [65] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, “Analyzing and improving the image quality of stylegan,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8110–8119, 2020.
- [66] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [67] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training GANs,” *Advances in neural information processing systems*, vol. 29, 2016.

Appendix A: Complexity Proofs

Here we give precise definitions and complete proofs for the results stated in Section 3. We begin by rigorously defining the complexity classes for drawing samples from a given probability distribution.

In [11], SampP is defined as the class of sampling problems that can be approximately solved by a probabilistic polynomial-time classical algorithm, and SampBQP is the class of sampling problems that can be approximately solved in polynomial-time by a quantum algorithm. Both of these complexity classes are defined in terms of binary variables. Though this does account for binary storage methods of finite-precision continuous values, this is not a natural setting when considering latent distributions for generative models.

Within this paper, we consider distributions over continuous spaces, which motivates extending these complexity classes to include continuous notions of approximate sampling using metrics such as the Wasserstein metric. For the classes of quantum distributions we consider, we similarly wish to step away from the language of binary variables in order to facilitate less constrained descriptions of sampling distributions such as Boson Sampling in which measurement results are discrete (the number of photons in an optical channel) but not binary. As such, we introduce two new complexity classes we will use throughout this section.

Definition: *Efficient Approximate Continuous Classical Sampling*

We define, with $\mathcal{C}$, the space of sampling problems of the form $\mathcal{S} = (P_{z_n})_{n \in \mathbb{N}}$ (where each P_{z_n} is a probability distribution over $\mathbb{R}^n$) such that we can sample from distributions $(\hat{P}_{z_n})_{n \in \mathbb{N}}$ with $W(P_{z_n}, \hat{P}_{z_n}) < \epsilon[60]$ using a classical algorithm in $\text{Poly}(n, 1/\epsilon)$ time.

Definition: *Efficient Approximate Discrete Quantum Sampling*

We define, with $\mathcal{Q}$, the space of sampling problems of the form $\mathcal{S} = (P_{z_n})_{n \in \mathbb{N}}$ (where each P_{z_n} is a probability distribution over $\mathbb{N}^n$) such that we can sample from a distribution $\hat{P}_{g_n(z_n)}$ such that $\|P_{z_n} - \hat{P}_{g_n(z_n)}\| < \epsilon$ using a quantum circuit in $\text{Poly}(n, 1/\epsilon)$ time, and there exists no classical algorithm that can do the same.

We note that, in the scenario where each entry in $\mathbb{N}$ is restricted by a given upper bound, this class is exactly equivalent to SampBQP/SampP. Boson Sampling from n -mode circuits is a task thought to be in $\mathcal{Q}$ [31].

We use different metrics for the proximity of probability distributions for each class in order to capture two distinct notions of ‘closeness’ when discussing discrete and continuous sampling tasks. We note that the SampP class is not a strict subset of the class $\mathcal{C}$ as the latter uses Wasserstein distance to define the closeness of distributions rather than total variation distance. For our purposes of considering continuous classical latent distributions and data distributions, we will use the Wasserstein metric.

Next, we describe the setting that we use for the generative models we study throughout this section.

Generative Model Setting: We define our collection of latent distributions as a collection $(P_{z_n})_{n \in \mathbb{N}}$, where P_{z_n} is a distribution over $\mathbb{R}^n$. We will consider collections of neural network architectures $(g_n)_{n \in \mathbb{N}}$ where g_n takes as input a vector of length n . In this section, we will assume these architectures are all efficiently classically implementable in the sense that there exists a $\text{Poly}(n)$ classical algorithm that applies the collection of functions. We will generally use $(P_{g_n(z_n)})_{n \in \mathbb{N}}$ to denote the pushforward distributions, where:

$$g_n(Z) \sim P_{g_n(z_n)} \quad \text{where} \quad Z \sim P_{z_n}$$

Remark 1: If the collection of latent distributions $(P_{z_n})_{n \in \mathbb{N}} \in \mathcal{C}$, and the neural networks $(g_n)_{n \in \mathbb{N}}$ are Lipschitz with constant c , then the pushforward distributions $(P_{g_n(z_n)})_{n \in \mathbb{N}} \in \mathcal{C}$.

Proof: Let $(\hat{P}_{z_n})_{n \in \mathbb{N}}$ be the collection of efficiently classically sample-able approximations guaranteed to exist by $(P_{z_n})_{n \in \mathbb{N}} \in \mathcal{C}$. Let $\hat{P}_{g_n(z_n)}$ denote the distribution sampled from by applying g_n to values sampled according to $\hat{P}_{z_n}$. We note:

$$\begin{aligned} W(P_{g_n(z_n)}, \hat{P}_{g_n(z_n)}) &= \inf_{\gamma \in \Gamma(P_{g_n(z_n)}, \hat{P}_{g_n(z_n)})} \mathbb{E}_{(X,Y) \sim \gamma} |X - Y| \\ &= \inf_{\gamma \in \Gamma(P_{z_n}, \hat{P}_{z_n})} \mathbb{E}_{(X,Y) \sim \gamma} |g_n(X) - g_n(Y)| \\ &\leq c \cdot \inf_{\gamma \in \Gamma(P_{z_n}, \hat{P}_{z_n})} \mathbb{E}_{(X,Y) \sim \gamma} |X - Y| \\ &= cW(P_{z_n}, \hat{P}_{z_n}) \\ &< c\epsilon \end{aligned}$$

As we can apply (g_n) in $\text{Poly}(n)$ time, we can sample from $(\hat{P}_{g_n(z_n)})_{n \in \mathbb{N}}$ in $\text{Poly}(n, 1/\epsilon)$ time. Thus, $(P_{g_n(z_n)})_{n \in \mathbb{N}} \in \mathcal{C}$. $\square$

Theorem 1: Let the neural networks $(g_n)_{n \in \mathbb{N}}$ be invertible such that there is a classical algorithm that applies g_n^{-1} in $\text{Poly}(n)$ time and further, g_n^{-1} is Lipschitz continuous with constant c . Let $(P_{z_n})_{n \in \mathbb{N}}$ be in $\mathcal{Q}$. Then, sampling from the collection of pushforward distributions $(P_{g_n(z_n)})_{n \in \mathbb{N}}$ is not in $\mathcal{C}$.

Proof: For a contradiction, we assume that the sampling task defined by $(P_{g_n(z_n)})_{n \in \mathbb{N}}$ is in $\mathcal{C}$. Let $(\hat{P}_{g_n(z_n)})_{n \in \mathbb{N}}$ denote the classically efficiently sample-able approximation guaranteed to exist by the definition of $\mathcal{C}$. Let $\hat{P}_{z_n}$ denote the distribution sampled by applying g_n^{-1} to values sampled according to $\hat{P}_{g_n(z_n)}$. We claim that $W(P_{z_n}, \hat{P}_{z_n}) < c\epsilon$. Indeed,

$$\begin{aligned} W(P_{z_n}, \hat{P}_{z_n}) &= \inf_{\gamma \in \Gamma(P_{z_n}, \hat{P}_{z_n})} \mathbb{E}_{(X,Y) \sim \gamma} |X - Y| \\ &\leq \inf_{\gamma \in \Gamma(P_{z_n}, \hat{P}_{z_n})} \mathbb{E}_{(X,Y) \sim \gamma} c \cdot |g_n(X) - g_n(Y)| \\ &= c \cdot \inf_{\gamma \in \Gamma(P_{g_n(z_n)}, \hat{P}_{g_n(z_n)})} \mathbb{E}_{(X',Y') \sim \gamma} |X' - Y'| \\ &= cW(P_{g_n(z_n)}, \hat{P}_{g_n(z_n)}) \\ &< c\epsilon \end{aligned}$$

Thus, $\exists$ a collection of couplings $\gamma_n^* \in \Gamma(P_{z_n}, \hat{P}_{g_n(z_n)})$ such that, $\forall n$:

$$\mathbb{E}_{(X,Y) \sim \gamma_n^*} |X - Y| \leq c\epsilon \quad \Rightarrow \quad \mathbb{P}_{(X,Y) \sim \gamma_n^*} (|X - Y| \geq 0.5) \leq 2c\epsilon$$

We note that, if $X \in \mathbb{R}^n, Y \in \mathbb{N}^n$ and $|X - Y| < 1/2$, then rounding every entry of X to the nearest integer returns Y . Thus, the above result shows that sampling from $\hat{P}_{g_n(z_n)}$ and then rounding each entry to the nearest integer, samples from a distribution within $2c\epsilon$ total variation distance of P_{z_n} . Sampling from $(P_{g_n(z_n)})_{n \in \mathbb{N}}$ is in $\mathcal{C}$, applying g_n^{-1} is done in $\text{Poly}(n)$ time and rounding decimal values to integers can be done in constant time so we can sample from this distribution classically in $\text{Poly}(n, 1/\epsilon)$ time. As these distributions are within $2c\epsilon$ total variation distance of P_{z_n} , the task of sampling from $(P_{z_n})_{n \in \mathbb{N}}$ within ϵ TVD can be done classically in $\text{Poly}(n, 1/\epsilon)$ time.[61] This contradicts $(P_{z_n})_{n \in \mathbb{N}} \in \mathcal{Q}$. We conclude the claimed result. $\square$

Recall that, in Section 3, we define GAN-induced distances. We will prove the following result with this language. First, we define $(G_{n,\epsilon})_{n \in \mathbb{N}, \epsilon \in \mathbb{R}_+}$ to be collections of neural network architectures, with the condition that there exists a $\text{Poly}(n, 1/\epsilon)$ time classical algorithm capable of implementing all functions in $G_{n,\epsilon}$. We fix D to be the Wasserstein metric.

Corollary 1: Let us be in the same setting as described in Theorem 1. Let $(\hat{P}_{z_n})_{n \in \mathbb{N}} \in \mathcal{C}$ be a collection of latent distributions we can sample from exactly in $\text{Poly}(n)$ time. Let $(\hat{P}_{g_n,\epsilon(z_n)})_{n \in \mathbb{N}, \epsilon \in \mathbb{R}}$ be the resulting collection of pushforward distributions.

There exists $\epsilon \in \mathbb{R}, n \in \mathbb{N}$ such that:

$$D^{G_{n,\epsilon}}(\hat{P}_{z_n}, P_{g_n(z_n)}) \geq \epsilon$$

whereas, by definition,

$$D^{g_n}(P_{z_n}, P_{g_n(z_n)}) = 0 \quad \forall n$$

Proof: By assumption, we can sample from $(\hat{P}_{g_n,\epsilon(z_n)})$ in $\text{Poly}(n, 1/\epsilon)$ time. If $\forall n, \epsilon$, we have $W(\hat{P}_{g_n,\epsilon(z_n)}, P_{g_n(z_n)}) < \epsilon$, then $(P_{g_n(z_n)})_{n \in \mathbb{N}} \in \mathcal{C}$. This contradicts Theorem 1. Thus $\exists n, \epsilon$ such that :

$$W(\hat{P}_{g_n(z_n)}, P_{g_n(z_n)}) \geq \epsilon$$

. $\square$

Precisely, the result demonstrated is that, no matter the choice of classical latent distribution and efficiently implementable neural network architecture, it is not possible to simulate the performance of a specific set of GAN architectures utilising latent distributions generated by quantum circuits.

Appendix B: Boson sampling systems

1. Boson sampling theory

Boson sampling is a non-universal model of quantum computation proposed by [11], in which identical photons are sent into an interference circuit, and a measurement is performed to determine where the photons left the circuit. A

lossless interference circuit with N channels is described by an $N \times N$ unitary matrix U , which can be physically implemented by a wide range of physical devices such as integrated chips [62] or optical fiber loops [14]. If k identical photons are sent into the first k channels of an interference device, then the probability of measuring an output configuration where t_i photons were found in output channel i is

$$p(t_1, \dots, t_N) = \frac{|\text{Perm}(U_{k,T})|^2}{t_1! \dots t_N!}$$

where Perm indicates the matrix permanent and $U_{k,T}$ is the submatrix of U obtained by taking the first k columns of U and by repeating each row i t_i times (if $t_i = 0$ then row i is not in $U_{k,T}$). The permanent of an n by n matrix M with elements $x_{i,j}$ is defined as

$$\text{Perm}(M) = \sum_{\sigma \in S_n} \prod_{i=1}^n x_{i,\sigma_i}$$

where S_n is the set of all permutations of the numbers 1 to n . Although this expression is similar to that of a matrix determinant, unlike a determinant calculating the permanent of a matrix is in general a computationally hard problem. Indeed, if M is an arbitrary real or complex matrix then this problem is $\#P$ -hard [11]. Since calculating a permanent is hard, [11] show that the problem of sampling from the output distribution of a boson sampler, even approximately, is also hard. The classical complexity of simulating a boson sampler increases exponentially with the number of photons, such that current classical computers cannot simulate more than a few tens of photons [45].

2. Current state of the art

Over the last few years, improvements in photon source, circuit and detector technology have led to several boson sampling experiments at a scale that is a challenge for classical computers. For instance, [29] performed a boson sampling experiment in which up to 76 photons were measured in 100 channels using a fixed interference circuit. Later, [14] performed a boson sampling experiment in an optical fiber loop system involving up to 219 photons in 216 channels. However, classical algorithms were later discovered that can leverage the high loss levels in these systems to simulate them [59]. More recently, a demonstration of boson sampling with thousands of photons was performed, with the authors claiming that this cannot be simulated with any existing classical algorithm [63].

3. Boson samplers with delay lines

Boson samplers based on optical delay lines have been proposed and demonstrated as an experimentally feasible route to large-scale boson sampling [14, 51, 52]. In these systems, illustrated in figure 5, a single photon source is used to produce a sequence of identical single photons, which then interfere with other in a sequence of optical delay lines (or "loops"). The resulting superposition state is measured by a single photon number resolving detector. The main advantage of this approach is that only a single photon source and detector are required, even though many photons and time-bins may be involved. Moreover, only three loops of different lengths are required to reach the classically hard to simulate regime [51, 52]. Furthermore, in [14] and in the ORCA Computing PT-2, the coupling parameter between each loop and each input/output is re-programmable between each time step using fast electro-optic modulators. This opens the door to the possibility of jointly training these parameters with the parameters of a neural network.

4. ORCA Computing PT-2 processor

The boson sampling system used in our experiments is an ORCA Computing PT-2, which is a commercially available loop-based boson sampling system. It consists of a photon source based on parametric downconversion, two sequential optical delay lines of the same length, and photon-number resolving superconducting nanowire detectors. It can support up to 16 photons interfering in 32 time-bins. The coupling parameters between each optical delay line and the input/output channels are fully programmable.

Due to optical losses in the system, typically significantly fewer photons are detected than are sent in. As such, we also used post-selection where we populated all 32 input time bins and discarded all results in which fewer than 16 photons were measured. Since lost photons in a linear optical network act as though they had never existed in the first

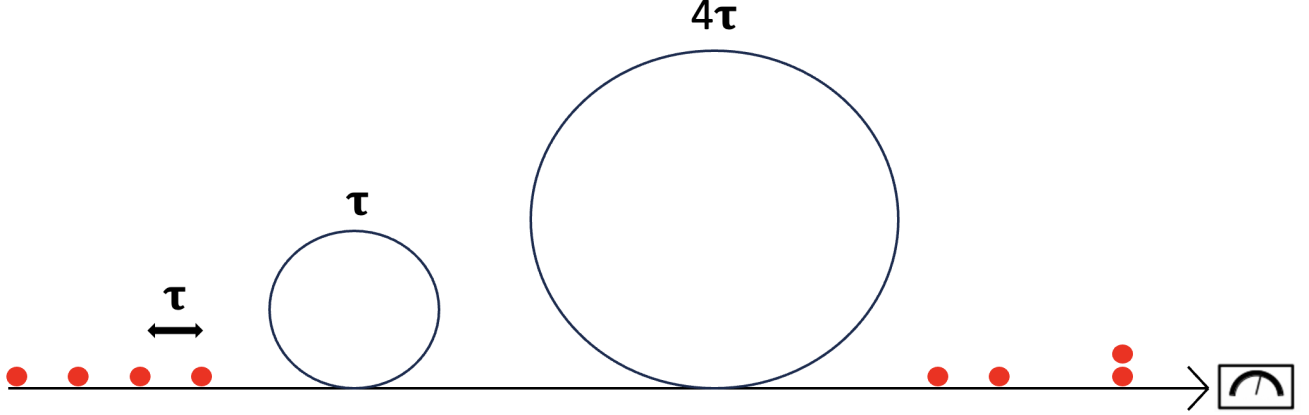


FIG. 5: Illustration of a loop-based boson sampling system. Here, four sequential single photons, separated by time τ , are sent into a system consisting of two sequential loops with lengths such that each photon that enters the loop can interfere with a subsequent photon. Sequences of loops of different lengths ensures that a photon can interfere with a photon several time-steps behind.

place, this regime effectively mimics roughly 16 photons in 32 channels with randomized input locations. Despite this input randomization, quantum effects such as photon bunching still occur and affect the output statistics.

To run our experiments, we collected a dataset of 500k samples with fixed set of randomly initialized coupling parameters, which took 40 minutes on this system. The system we used is hosted at the UK National Quantum Computing Centre.

Appendix C: Experiments on 2D mixtures of Gaussians

These experiments used a WGAN-GP framework, where both the generator and discriminator are feedforward neural networks using two hidden layers with 256 neurons and LeakyReLU activations. All latent distributions were size-16. Both the distinguishable and indistinguishable samples were produced by randomly initialized simulated photonic systems with a 3-loop architecture in a 1-3-9 configuration. The models were trained for 5000 training steps, and all distributions were centered to have a mean value of 0 before being injected into the generator. Though figure 2 presents results for only a single seed, the experiments were repeated for 5 seeds with repeatable behavior. When training for 10,000 iterations, we observed that the latent distribution with distinguishable photons was often able to close the gap with the indistinguishable photons, which implies that the generator is able to overcome the difference between those distributions on this dataset with further training. However, further training did not improve results with Gaussian and Bernoulli latent distributions.

Appendix D: Experiments on a toy dataset

For our small-scale experiments, we use fully connected networks with 2 hidden layers of 512 neurons for both the generator and the critic. The generator output is of dimension 16. Both models use a Relu activation function in all their hidden layers. All the latent distributions that we consider also have a dimension of 16. The training is done using batches of 500 samples over 40k iterations. As in [64], we update the critic 5 times for each generator update. We use a RMSProp optimizer with a learning rate of 5×10^{-4} . The quantum latent distribution is obtained by sampling from a simulated boson sampler with 8 photons in an interference network with 16 channels. The interference network is mathematically described by a 16×16 unitary matrix, which we draw randomly from the Haar measure for each experiment. For each of the 12 experimental runs, we independently drew 3 random unitary matrices: one for the quantum latent distribution, one for the non-interfering photons, and one for the data the GAN is trained on.

Appendix E: QM9 experiments

Our implementation started from the repository at <https://github.com/kfzyqin/Implementation-MolGAN-PyTorch>. We made a few changes to lead to improved performance. First, the MolGAN generator was modified by feeding affine transforms of the latent code $\mathbf{z}$ to all layers of the feedforward network, similar to the approach taken in [65]. The activation functions used in the generator were also changed to LeakyReLU. Moreover, instead of using 3 hidden layers, we used 5 hidden layers with sizes 64, 176, 288, 400 and 512.

Data augmentation was used to extend the training dataset and make the model more amenable to translation and rotation invariance of molecules, which helped improve permutation invariance in the graph representation. A "permutation probability" parameter set to 0.3 was used to determine the strength of the data augmentation.

In all experiments, we used a batch size of 256 and the Adam [66] optimizer with a learning rate of 10^{-4} and trained the models for 20000 iterations. Initially, the higher learning rate of 10^{-3} from the original repository was used, but we found that the lower learning rate of 10^{-4} led to significantly improved results.

The metrics were calculated with the code provided in <https://github.com/MorganCThomas/MolScore>. Example training curves for the size-16 latent distributions, with the FCD regularly evaluated during training, are shown in figure 6. Some examples of generated molecules can be found in figure 7

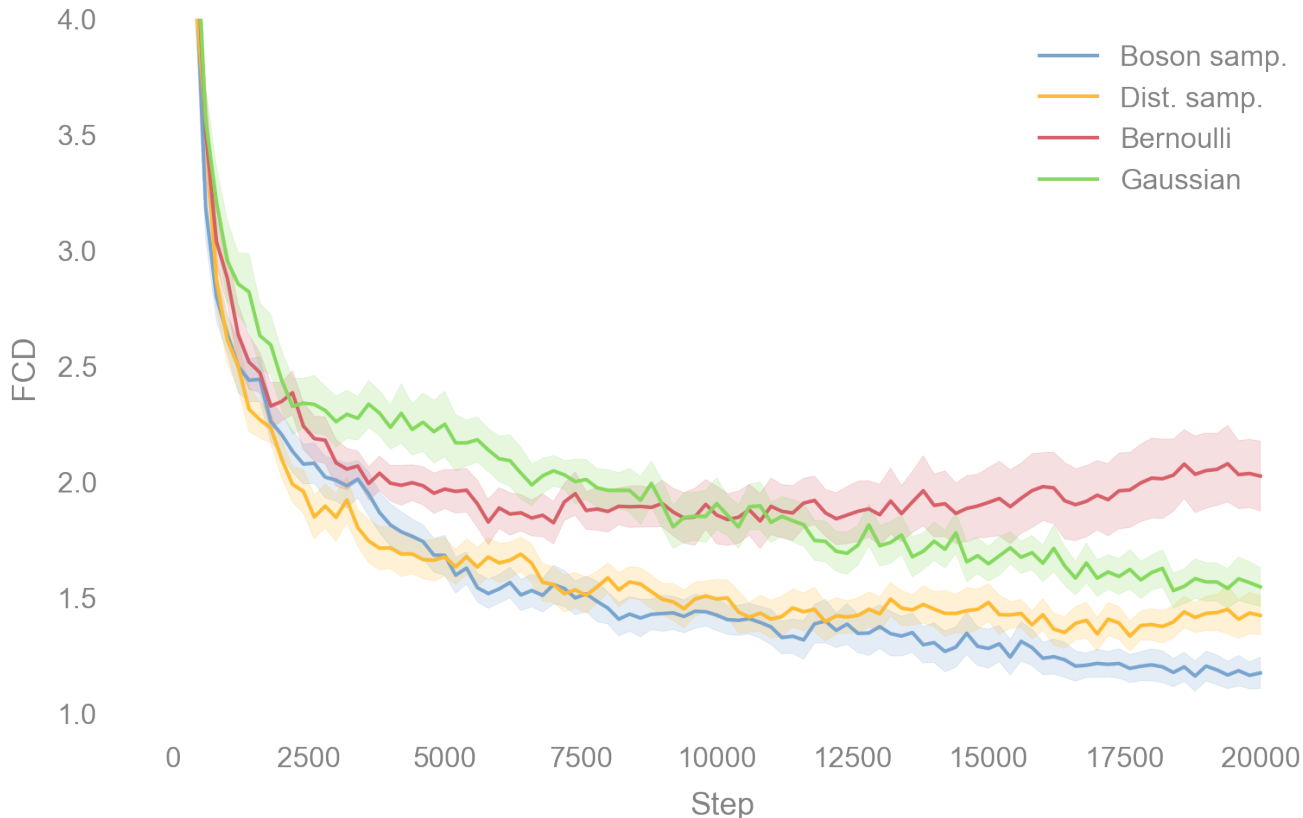


FIG. 6: FCD scores during training for different latents of $z_{dim}=16$. The mean and standard error of the mean are calculated from 20 random seeds.

1. Note on circuit randomization

In general, to ensure that performance differences arise from general properties of the latent space distributions and not from specific circuit realizations, the optical circuits for both distinguishable and indistinguishable photons are re-sampled for every seed. This is the case for all experiments with a toy dataset, and also for all size-16 and most size-32 latents on the QM9 dataset.

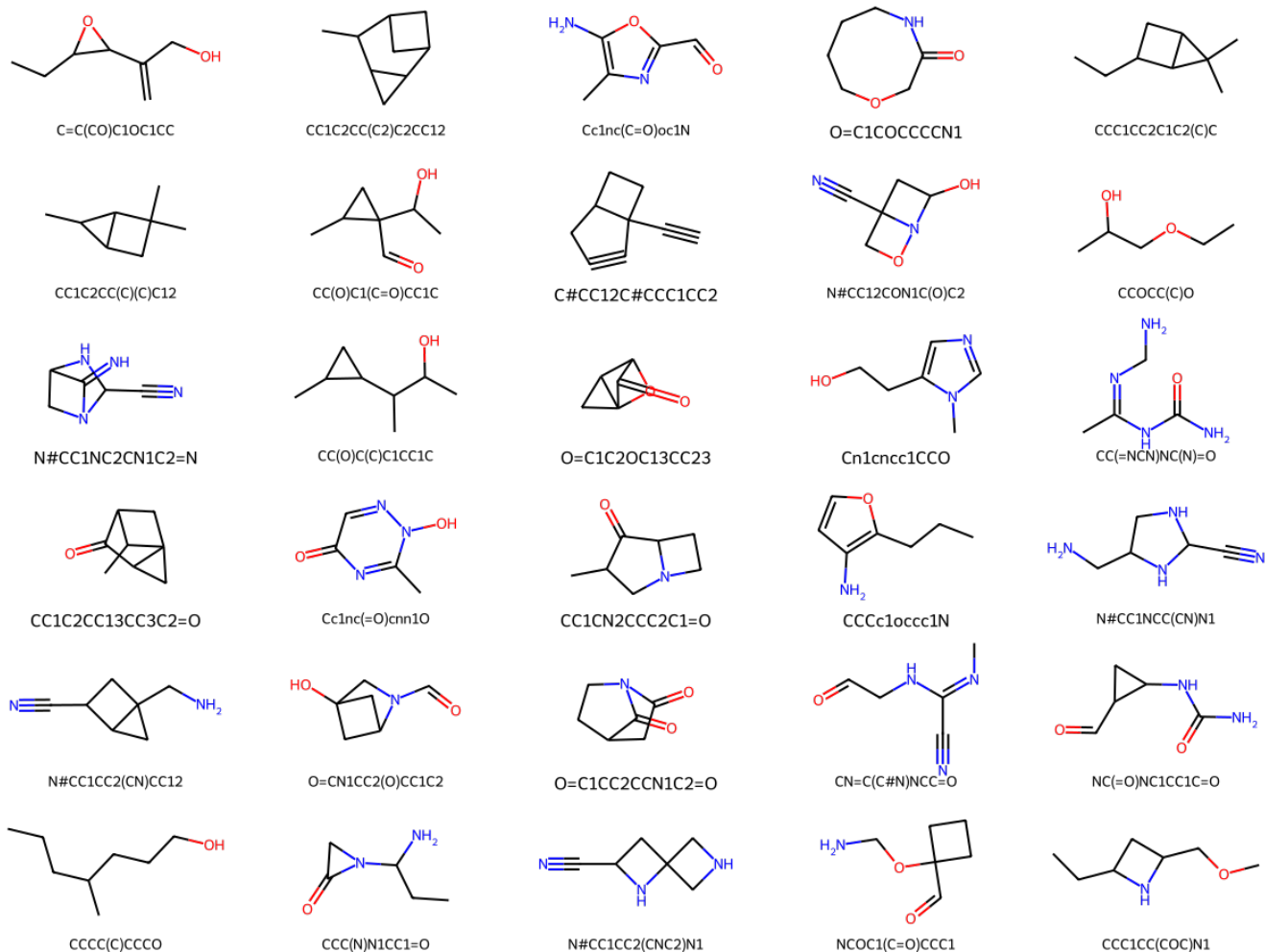


FIG. 7: Examples of generated molecules using a trained model with a quantum latent distribution.

However, there are two instances in which the circuit was not re-randomized for every training seed, due to the time and computational requirements involved in collecting a sufficiently large number of samples to use as a latent distribution:

- Size-48 Haar-random circuits, for which collecting a single set of 1 million samples for indistinguishable photons took 3.75 hours on 100 CPUs. However, we observed from our experiments on size-32 Haar-random circuits that there is not much variability between the results obtained for different seeds for this type of circuit.
- Experiments using the ORCA PT-2 system, where collecting a single set of 500k samples took 40 minutes and experimental time was limited.

2. Computational resources

The computational resources used for our QM9 experiments can be divided into the time taken to collect the latent space samples, and the training time. It took 40 minutes to collect 500k samples with an ORCA PT-2. For the simulated boson sampler size-32 models, 20 different sets of optical circuit parameters were used to precompute 1 million samples (a total of 20 million) and 1 million samples for the size-48 models using a single set of optical circuit parameters. The time taken to obtain these samples using a simulated boson sampler can be estimated using Table V. The time taken to collect samples from simulated distinguishable photons was negligible.

Each training experiment ran on a single NVIDIA HGX™ A100 GPU 80GB. The size-16 models were trained using 20 different sets of optical circuit parameters where the samples were generated on the spot. For each of these types of

models it took around 90 minutes to train when using the distinguishable sampler, 115 minutes for the boson sampler, and 48 minutes for both the Gaussian and Bernoulli distributions. Training the models for size-32 and size-48 and for both distinguishable and boson samplers (where we used a custom dataloader to load pre-computed samples) took around 55 minutes each. The time taken for other classical latents is the same as for size-16.

The tables below show the time and rates needed to generate simulated samples of a boson sampler with n photons in m optical channels.

TABLE V: Sampling time for 500 samples on a single AMD EPYC™ 7003 processor

n	m	Time
8	16	19.6 ms $\pm$ 34.3 μ s
16	32	4.15 s $\pm$ 49 ms
24	48	670 s $\pm$ 16.5 s

Appendix F: DDGANs

Diffusion models consider a process that gradually corrupts examples of a data distribution with noise. First, a forward diffusion process adds noise step by step, where this noise is sampled from a Gaussian distribution. The reverse process is where the generative process takes place. Given a fully corrupted image $\mathbf{x}_T$ and the current time step t , a neural network predicts the mean of a Gaussian distribution $\mu_\theta(x_t, t)$, such that sampling from that distribution removes the noise applied in one step.

The idea behind Denoising Diffusion GANS ([55]) is that to have a faster reverse process we need to go back multiple steps at a time, in which case the assumption that the denoising distributions are Gaussian fails. A solution is to perform the reverse process using more complex multimodal distributions. To achieve this, a full multimodal conditional GAN is used at each step of the denoising diffusion process. The generative process is performed using 4 denoising steps ($T = 4$). For a given (fixed) Gaussian noise input $\mathbf{x}_T$, different images $\mathbf{x}_0$ are obtained using different samples from the latent distribution z . The combination of the initial Gaussian noise and samples from the latent determine the final result.

The boson and distinguishable sampler latents for these experiments were produced by two unstructured optical circuits, with transfer matrices described by Haar-random unitary matrices, with 24 photons in 48 channels.

We use the code in <https://github.com/NVlabs/denoising-diffusion-gan> to run these experiments. The code was minimally modified to use the boson and distinguishable samplers as latents for the generator during training and inference. We used the commands in the repo to train and evaluate the models with the only difference being that the latent space dimension in the original model was $z_{dim} = 256$. Some example outputs are shown in figure 8.

TABLE VI: Frechet Inception Distance (FID) for different latent distributions with a denoising diffusion GAN, with means and standard errors reported over 5 training seeds

Latent Type	FID $\downarrow$
Indistinguishable photons	4.025 $\pm$ 0.075
Distinguishable photons	4.071 $\pm$ 0.064
Gaussian	4.040 $\pm$ 0.040

1. Computational resources

Each training run took around 23.2 hours each, running on four NVIDIA HGX™ A100 GPU 80GB.

Appendix G: Flow matching

We build on the Optimal-Transport variant of Conditional Flow Matching from [27], as implemented in <https://github.com/atong01/conditional-flow-matching>. To reconcile the dimensionality mismatch between the quantum

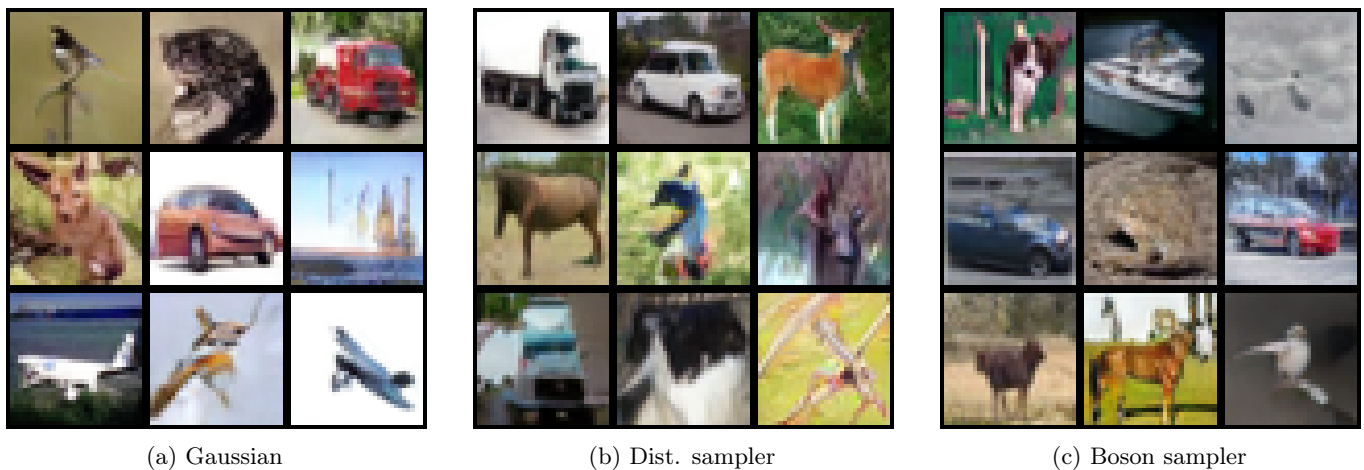


FIG. 8: Some example images produced by our trained DDGAN models with three different latent distributions.

latent distribution and our two-dimensional data, we introduce an untrained decoder that maps quantum latents to 2D source samples. Specifically, we first apply an affine transformation, scaling and shifting each latent vector by its global mean and variance, and then project the result through a Kaiming-normal-initialized kernel. This ensures that the resultant source distribution remains comparable in scale to the standard Gaussian source distribution. The decoder’s parameters remain fixed throughout training, and all other training procedures follow the original codebase. During sampling, we draw from the quantum latent distribution, decode each sample, and then perform the usual flow-matching routine.

Appendix H: Negative results

In this section, we present additional results for experiments in which we found that the performance of a model was not improved by using a quantum latent distribution.

1. StyleGAN with CIFAR-10

We investigated the StyleGAN model [2] with the CIFAR-10 dataset and different types of latent distributions. We used the inception score [67] as the metric for model performance. Table VII shows the inception scores (IS) [67] achieved by the trained models. We find that the performance of all models is similar, suggesting that the choice of latent distribution matters less for this type of model and dataset.

TABLE VII: Inception scores ($\pm$ standard error over 5 seeds) obtained on CIFAR 10 using different latent distributions.

	Gaussian	Bernoulli	Quantum
Inception score $\uparrow$	7.57 ± 0.02	7.66 ± 0.04	7.54 ± 0.01

2. QM9 experiments

We also found that small changes to the model and training regime used in our QM9 experiments could make a significant difference and erase a large part of the observed performance differences. Using the same model but with a higher learning rate of 10^{-3} (instead of 10^{-4}), a permutation probability of 0.2, and 10,000 training steps (instead of 20,000) led to the results shown in table VIII for size-16 latents. We see that the results are worse overall, and that there is also a much smaller difference between the different latents. This indicates that performance differences achieved using a different latent distributions are sensitive to model and training parameters dataset.

TABLE VIII: Metrics ($\pm$ standard error over 20 seeds) for different latents using training runs with a higher learning rate and a lower permutation probability

Latent Type	z_{dim}	FCD $\downarrow$	# Valid & unique $\uparrow$	# Novel $\uparrow$
Boson samp. 1-3-9	32	1.623 ± 0.09	1971 ± 64	1161 ± 45
Dist. samp. 1-3-9	32	1.717 ± 0.08	1842 ± 56	1071 ± 38
Gaussian	32	1.554 ± 0.06	1756 ± 58	1048 ± 42