Supplementary Information: Quantum Generative Adversarial Networks for Learning and Loading Random Distributions

Christa Zoufal,[1,2,*] Aurélien Lucchi,[2] and Stefan Woerner[1]

[1] *IBM Research – Zurich, Rueschlikon 8803, Switzerland*
[2] *ETH Zurich, Zurich 8092, Switzerland*

I. SUPPLEMENTARY NOTES

A. Isometric Quantum Generator

Closed quantum systems follow a unitary evolution. The evolution of an open quantum system, i.e. a quantum system that interacts with an environment, evolves according to an isometry instead of a unitary [1].

In general, every isometry can be described by a unitary that acts on a larger system. In other words, an isometry is given by a partial trace of a unitary quantum state evolution. The dynamics of an open quantum systems, and thus also a quantum generator acting as an isometry, can be implemented with additional ancilla qubits. Depending on the setting, the use of an isometric quantum generator can be advantageous to learn random distributions.

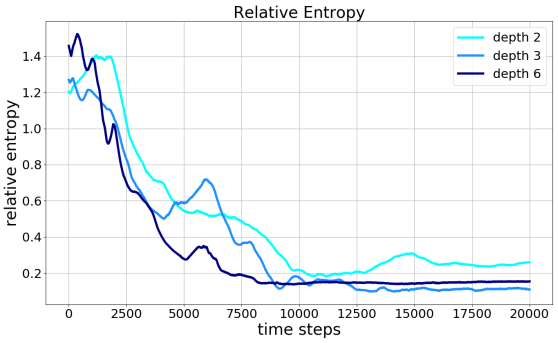
B. Multivariate Historical Data for Portfolio Optimization

The qGAN scheme can also be used to learn and load multivariate random distributions. Here,we present the learning and loading of a distribution underlying the first two principle components of multivariate, constant maturity treasury rates of US government bonds. Note that the trained quantum channel can be used within the discussed QAE algorithm to evaluate, for instance, the fair price of a portfolio of government bonds, see [2].

The following results are computed with a quantum simulation. The training data set X consists of more than $5,000$ samples, whereby data samples smaller than the $5\%-$percentile and bigger than the $95\%-$percentile have been discarded to reduce the number of required qubits for a reasonable representation of the distribution. The optimization scheme uses data batches of size $1,200$ and is run for $20,000$ training epochs.

Furthermore, we use depth $k \in \{2,3,6\}$, unitary quantum generators that act on $n = 6$ qubits, i.e. 3 qubits per dimension (principle component). The input state $|\psi_{\text{in}}\rangle$ is prepared as a multivariate uniform distribution and the generator parameters $\boldsymbol{\theta}$ are initialized with random draws from a uniform distribution on the interval $[-\delta, +\delta]$ with $\delta = 10^{-1}$.

Here, the classical discriminator is composed of a $512-$ node input layer, a $256-$ node hidden-layer, and a single-node output layer. Equivalently to the discriminator described in the main text, the hidden layers apply linear transformations followed by Leaky ReLU functions [3] and the output layer employs a linear transformation followed by a sigmoid function. The evolution of the relative entropy between the generated and the real probability distribution is shown in Fig. 1.



Supplementary Figure 1 Relative entropy progress for a multivariate random distribution. The progress of the relative entropy between the quantum generator and the multivariate random distribution underlying the training data is shown for quantum generators with depth $k \in \{2,3,6\}$.

II. SUPPLEMENTARY METHODS

A. Practical Initialization of a Normal Distribution

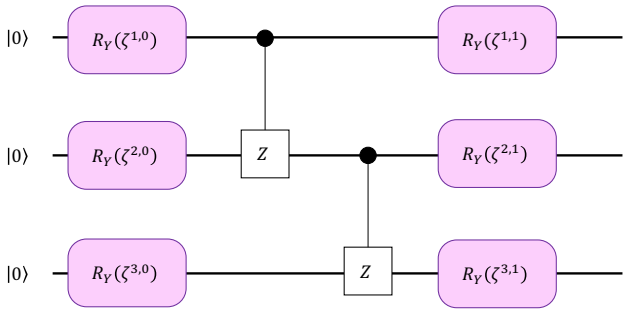
As proven in [4], a normal distribution can be efficiently loaded into a quantum state. However, the suggested loading method requires the use of involved quantum arithmetic techniques. Considering the illustrative examples from the main text, it is sufficient to load an approximate normal distribution as the initialization state. This can be achieved by fitting the parameters of a 3-qubit variational quantum circuit with depth 1 with a least squares loss function. More specifically, we minimize the distance between the measurement probabilities p_ζ^i of the circuit output and the probability density function of a discretized normal

* ouf@zurich.ibm.com

distribution q^i

$$\min_{\zeta} \sum_i \left\| p_\zeta^i - q^i \right\|^2 . \tag{1}$$

The circuit used for training is depicted in Fig. 2. Note that this approach does not scale, particularly not for higher-dimensional distributions. The sole purpose of this approach is to generate shallow testing circuits.



Supplementary Figure 2 Variational quantum circuit for approximate loading of a normal distribution. To reduce the number of gates required for initialization with a normal state, we use the illustrated circuit to load an approximate discretized normal distribution.

We trained the circuit parameters $\boldsymbol{\zeta}_{\mathrm{ln}}$ to approximate a normal distribution with the mean and the standard deviation of the data samples drawn from each a log-normal distribution with $\mu = 1$ and $\sigma = 1$,

$$\boldsymbol{\zeta}_{\mathrm{ln}} = [0.3580, 1.0903, 1.5255, 1.3651, 1.4932, -0.9092] ,$$

$\boldsymbol{\zeta}_{\mathrm{tr}}$ to approximate a triangular distribution with lower limit $l = 0$, upper limit $u = 7$ and $\mu = 2$,

$$\boldsymbol{\zeta}_{\mathrm{tr}} = [1.5343, 1.6183, 0.8559, -0.4041, 0.4953, 1.2238]$$

and $\boldsymbol{\zeta}_{\mathrm{bm}}$ to approximate a bimodal distribution consisting of two superimposed Gaussian distributions with $\mu_1 = 0.5$, $\sigma_1 = 1$ respectively $\mu_2 = 3.5$, $\sigma_2 = 0.5$,

$$\boldsymbol{\zeta}_{\mathrm{bm}} = [0.4683, 0.8200, 1.4512, 1.1875, 1.3883, -0.8418] ,$$

whereby the least square errors are of the order 10^{-4}.

B. Analytic Gradients of a Variational Quantum Circuit

Compared to gradient-free optimization, gradient-based optimization methods have the potential to improve convergence rates, e.g. in a convex vicinity of local optima [5]. We now discuss a method to calculate analytic gradients [6–10] for the variational circuit layout illustrated in the main text.

Applying our $n-$qubit generator to the input state gives

$$\begin{aligned} |g_{\boldsymbol{\theta}}\rangle &= G_{\boldsymbol{\theta}} |\psi_{\mathrm{in}}\rangle \\ &= \prod_{p=1}^{k} \left(\bigotimes_{q=1}^{n} \left(R_Y \left(\theta^{q,p} \right) \right) U_{\mathrm{ent}} \right) \bigotimes_{q=1}^{n} \left(R_Y \left(\theta^{q,0} \right) \right) |\psi_{\mathrm{in}}\rangle \\ &= \sum_{j=0}^{2^n-1} \sqrt{p_{\boldsymbol{\theta}}^j} \, |j\rangle . \end{aligned} \tag{2}$$

We measure $|g_{\boldsymbol{\theta}}\rangle$ m times to obtain data samples $\boldsymbol{g}^l$, $l \in \{1, \ldots, m\}$ which can take 2^n different values. The generator loss function for a data batch of size m reads

$$L_G \left(\phi, \boldsymbol{\theta} \right) = -\frac{1}{m} \sum_{l=1}^{m} \log \left(D_\phi \left(\boldsymbol{g}^l \right) \right) , \tag{3}$$

or equivalently,

$$L_G \left(\phi, \boldsymbol{\theta} \right) = - \sum_{j=0}^{2^n-1} p_{\boldsymbol{\theta}}^j \log \left(D_\phi \left(\boldsymbol{g}^j \right) \right) , \tag{4}$$

with

$$p_{\boldsymbol{\theta}}^j = |\, \langle j | g_{\boldsymbol{\theta}} \rangle \,|^2 . \tag{5}$$

Updating the parameters $\boldsymbol{\theta}$ with gradient based methods requires the evaluation of

$$\frac{\partial L_G \left(\phi, \boldsymbol{\theta} \right)}{\partial \boldsymbol{\theta}^{i,l}} = - \sum_{j=1}^{m} \frac{\partial p_{\boldsymbol{\theta}}^j}{\partial \boldsymbol{\theta}^{i,l}} \log \left(D_\phi \left(\boldsymbol{g}^j \right) \right) . \tag{6}$$

According to [8], Eq. (6) can be evaluated by

$$\frac{\partial p_{\boldsymbol{\theta}}^j}{\partial \boldsymbol{\theta}^{i,l}} = \frac{1}{2} \left(p^j_{\boldsymbol{\theta}^{i,l}_+} - p^j_{\boldsymbol{\theta}^{i,l}_-} \right) , \tag{7}$$

with $\boldsymbol{\theta}^{i,l}_{\pm} = \boldsymbol{\theta}^{i,l} \pm \frac{\pi}{2} e_{i,l}$ and $e_{i,l}$ denoting the (i,l)-unit vector of the respective parameter space.

C. Statistical Measures

Two different statistical measures are utilized to evaluate the performance of the qGAN. Both measures are defined as a distance of two (empirical) probability distributions P and Q.

The Kolmogorov-Smirnov statistic [11, 12] is based on the (empirical) cumulative distribution functions $P\left(X \leq x\right)$ and $Q\left(X \leq x\right)$ and is given by

$$D_{\mathrm{KS}} \left(P || Q \right) = \sup_{x \in X} |P \left(X \leq x \right) - Q \left(X \leq x \right)| . \tag{8}$$

The statistic can be used as a goodness-of-fit test. Given the null-hypothesis $P\left(x\right) = Q\left(x\right)$, we draw

$s = 500$ samples from both distributions and choose a confidence level $(1-\alpha)$ with $\alpha = 0.05$. The null-hypothesis is accepted if

$$D_{\mathrm{KS}}(P\|Q) \leq \sqrt{\frac{\ln\frac{2}{\alpha}}{s}} = 0.0859. \tag{9}$$

Another measure that can be used to characterize the closeness of (empirical) discrete probability distributions $P(x)$ and $Q(x)$ is the relative entropy, also called Kullback-Leibler divergence [1, 13]. This entropy-related measure is given by

$$D_{\mathrm{RE}}(P||Q) = \sum_{x\in X} P(x)\log\left(\frac{P(x)}{Q(x)}\right). \tag{10}$$

The relative entropy represents a non-negative quantity, i.e. $D_{\mathrm{RE}}(P||Q) \geq 0$, where $D_{\mathrm{RE}}(P||Q) = 0$ holds if $P(x) = Q(x)$, for all values of x.

D. Quantum Amplitude Estimation

Given a quantum channel

$$\mathcal{A}|0\rangle^{\otimes n+1} = \sqrt{1-a}\,|\psi_0\rangle|0\rangle + \sqrt{a}\,|\psi_1\rangle|1\rangle, \tag{11}$$

where $|\psi_0\rangle$, $|\psi_1\rangle$ denote n-qubit states, the QAE algorithm [14], illustrated in Fig. 3, enables the efficient evaluation of the amplitude a. The algorithm requires m additional evaluation qubits that control the applications of an operator $\mathcal{Q} = -\mathcal{A}\mathcal{S}_0\mathcal{A}^\dagger\mathcal{S}_{\psi_0}$ where $\mathcal{S}_0 = \mathbb{I}^{\otimes n+1} - 2|0\rangle\langle 0|^{\otimes n+1}$ and $\mathcal{S}_{\psi_0} = \mathbb{I}^{\otimes n+1} - 2|\psi_0\rangle\langle\psi_0|\otimes|0\rangle\langle 0|$.

The error in the outcome - ignoring higher terms - can be bounded by $\frac{\pi}{2^m}$. Considering that 2^m is the number of quantum samples used for the estimate evaluation, this error scaling is quadratically better than the classical Monte Carlo simulation.

To use QAE for the pricing of European options, we need to construct and implement a suitable oracle $\mathcal{A}$. First, we load the uncertainty distribution that represents the spot price S_T of the underlying asset at the option's maturity T into a quantum state $\sum_{i=0}^{2^n-1}\sqrt{p_i}\,|i\rangle$. It should be noted that small errors in this state preparation only lead to small errors in the final result. Then, we add an ancilla qubit $|0\rangle$ and use a comparator circuit which applies an X gate to the ancilla if $i > K$, i.e.

$$|i\rangle|0\rangle \mapsto \begin{cases} |i\rangle|0\rangle & \text{, if } i \leq K \\ |i\rangle|1\rangle & \text{, if } i > K, \end{cases} \tag{12}$$

where K denotes the strike price. Now, the state reads

$$\sum_{i=0}^{K}\sqrt{p_i}\,|i\rangle|0\rangle + \sum_{i=K+1}^{2^n-1}\sqrt{p_i}\,|i\rangle|1\rangle. \tag{13}$$

Finally, we control the mapping of the payoff function to the amplitude of another ancilla qubit $|0\rangle$ with the comparison ancilla. This construction implements channel $\mathcal{A}$ and approximates the quantum state

$$\mathcal{A}|0\rangle^{\otimes n+1} = \sum_{i=0}^{K}\sqrt{p_i}\,|i\rangle|0\rangle|0\rangle + \sum_{i=K+1}^{2^n-1}\sqrt{p_i}\,|i\rangle|1\rangle\left(\sqrt{1-f(i)}\,|0\rangle + \sqrt{f(i)}\,|1\rangle\right), \tag{14}$$

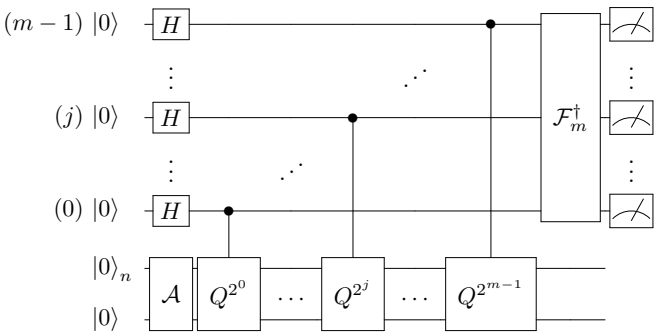
with $f(i) = \frac{i-K}{2^n-K-1}$. For practical reasons, we avoid the involved implementation of the exact linear objective rotation given in Eq. (14) by applying the approximation scheme introduced in [2].

Eventually, the probability of measuring $|1\rangle$ in the last ancilla is equal to

$$\mathbb{P}[|1\rangle] = \frac{1}{2^n-K-1}\sum_{i=K+1}^{2^n-1} p_i(i-K) \tag{15}$$

$$= \frac{1}{2^n-K-1}\mathbb{E}[\max\{0, S_T - K\}]. \tag{16}$$

We can see from comparing Eq. (11) and Eq. (14) that $\mathbb{P}[|1\rangle] = a$. It follows that we can use QAE to efficiently evaluate $\mathbb{E}[\max\{0, S_T - K\}] = \mathbb{P}[|1\rangle](2^n - K - 1)$.

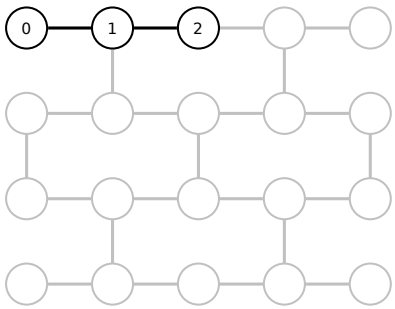


Supplementary Figure 3 Quantum Amplitude Estimation. The illustrated quantum circuit corresponds to the Quantum Amplitude Estimation algorithm with the inverse Quantum Fourier Transform [1] being denoted by $\mathcal{F}_m^\dagger$.

E. Hardware Efficient Circuit Implementation

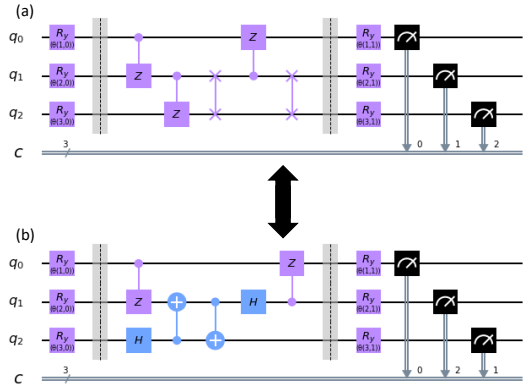
Due to the connectivity layout of the IBM Q Boeblingen chip, shown in Fig. 4, any subset of three qubits - we use qubits 0, 1, 2 - has linear connectivity only. Thus, the implementation of the entanglement block presented in the main text requires the use of SWAP gates, as shown in Fig. 5(a).

The implementation of $CZ \circ SWAP$ with the gate set currently available for IBM Q backends requires the use of 4 CX gates, i.e. 3 for the $SWAP$ and 1 for the



Supplementary Figure 4 IBM Q Boeblingen. The figure illustrates the connectivity of the IBM Q Boeblingen 20 superconducting qubit chip, as well as the qubits used for the qGAN training.

CZ. However, we can reduce the number of required CX gates, see Fig. 5(b). As shown in [15], the action of circuit (a) is equivalent to the action of circuit (b), which only utilizes 2 CX gates. During the training, circuit (b) maps the measurement of q_1 (q_2) on bit c_2 (c_1) to compensate for the second $SWAP$ in circuit (a). However, when using the generator circuit for data loading in another algorithm, such as QAE, an actual $SWAP$ gate must be implemented.



Supplementary Figure 5 Hardware efficient quantum generator implementation. The action of the illustrated circuits is equivalent. Since the quantum circuit at the bottom requires fewer CX gates, it is the favorable implementation choice for training a qGAN with actual quantum hardware. Notably, the lower circuit projects the measurement of qubit q_1 (q_2) on bit c_2 (c_1).

[1] Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information* (Cambridge University Press, 2010).

[2] Woerner, S. & J. Egger, D. Quantum risk analysis. *npj Quantum Information* **5** (2019).

[3] Pedamonti, D. Comparison of non-linear activation functions for deep neural networks on mnist classification task (2018). Preprint at `https://arxiv.org/pdf/1804.02763.pdf`.

[4] Grover, L. & Rudolph, T. Creating superpositions that correspond to efficiently integrable probability distributions (2002). Preprint at `https://arxiv.org/pdf/quant-ph/0208112.pdf`.

[5] Harrow, A. & Napp, J. Low-depth gradient measurements can improve convergence in variational hybrid quantum-classical algorithms (2019). Preprint at `https://arxiv.org/pdf/1901.05374.pdf`.

[6] Farhi, E. & Neven, H. Classification with quantum neural networks on near term processors (2018).

[7] Schuld, M., Bocharov, A., Svore, K. M. & Wiebe, N. Circuit-centric quantum classifiers (2018). Preprint at `https://arxiv.org/pdf/1804.00633.pdf`.

[8] Mitarai, K., Negoro, M., Kitagawa, M. & Fujii, K. Quantum circuit learning. *Phys. Rev. A* **98**, 032309 (2018).

[9] Zeng, J., Wu, Y., Liu, J.-G., Wang, L. & Hu, J. Learning and inference on generative adversarial quantum circuits. *Phys. Rev. A* **99** (2019).

[10] Liu, J.-G. & Wang, L. Differentiable learning of quantum circuit born machines. *Phys. Rev. A* **98**, 062324 (2018).

[11] Chakravarti, I., Laha, R. & Roy, J. *Handbook of methods of applied statistics* (Wiley, 1967).

[12] Justel, A., Peña, D. & Zamar, R. A multivariate kolmogorov-smirnov test of goodness of fit. *Statistics and Probability Letters* **35**, 251 – 259 (1997).

[13] Kullback, S. & Leibler, R. A. On information and sufficiency. *Ann. Math. Statist.* **22**, 79–86 (1951).

[14] Brassard, G., Hoyer, P., Mosca, M. & Tapp, A. Quantum Amplitude Amplification and Estimation. *Contemporary Mathematics* **305** (2002).

[15] Vatan, F. & Williams, C. Optimal quantum circuits for general two-qubit gates. *Physical Review A - Atomic, Molecular, and Optical Physics* **69** (2004). 0308006.