Quantum Neural Architecture Search with Quantum Circuits Metric and Bayesian Optimization

Trong Duong [1] **Sang Truong** [*2] **Minh Tam** [*3] **Bao Bach** [*4] **Ju-Young Ryu** [5] **June-Koo Kevin Rhee** [5]

Abstract

Quantum neural networks are promising for a wide range of applications in the Noisy Intermediate-Scale Quantum era. As such, there is an increasing demand for automatic quantum neural architecture search. We tackle this challenge by designing a quantum circuits metric for Bayesian optimization with Gaussian process. To this goal, we propose a new quantum gates distance that characterizes the gates' action over every quantum state and provide a theoretical perspective on its geometrical properties. Our approach significantly outperforms the benchmark on three empirical quantum machine learning problems including training a quantum generative adversarial network, solving combinatorial optimization in the MaxCut problem, and simulating quantum Fourier transform. Our method can be extended to characterize behaviors of various quantum machine learning models.

1. Introduction

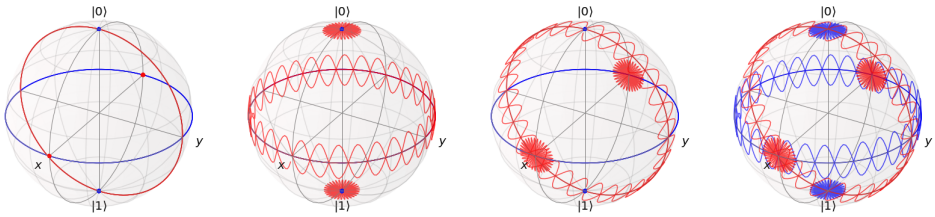
There has been hope that complex problems will be solved efficiently on universal fault-tolerant quantum computers. However, a useful fault-tolerant quantum computer would require a sustainably large number of high-fidelity logical qubits and gate operations. Recent advances in the field are making quantum devices with a few hundred physical qubits and limited error correction into reality. Even though it is still far from an ideal quantum computer, those near-term devices might have some uses in certain optimization problems. Generally one has to find the minimal value of an objective function that usually involves a Hamiltonian, which could be of quantum origin or be mapped onto from a classical objective value. For example, Maximum Cut (MaxCut) is an NP-complete combinatorial optimization problem (Garey et al., 1976) involves finding a partition $\{A, B\}$ of a given graph such that it maximizes sum of edge weights for edges connecting A and B. We assign a binary value x_i to every node corresponding to its class. The classical objective function is $C(x) = \sum_{i,j=1}^{n} w_{ij} x_i x_j$, where w_{ij} is the edge weight. By mapping $x_i \mapsto \frac{1}{2}(1 - Z_i)$, where Z_i is the Pauli Z operator with matrix representation $\mathrm{diag}(1, -1)$, the objective can be translated into a Hamiltonian

$$H = \sum_{x \in \{0,1\}^n} C(x) \left|x\right\rangle \left\langle x\right| \tag{1}$$

in a quantum approximate optimization algorithm. The problem becomes finding the extreme value $\max_{|\psi\rangle \in \mathcal{H}} \langle\psi| H |\psi\rangle$, which is a general form of optimization to be solved with variational quantum models. Since going through every element of the Hilbert space $\mathcal{H}$ is impossible, we parametrize the search space by a Parametrized Quantum Circuit (PQC) or a Quantum Neural Network (QNN), expressed by unitary matrix $U(\theta)$ that rotates an initial state $|0\rangle$. Whether the codomain $\{U(\theta)|0\rangle\}$ is a good search space depends entirely on the QNN architecture. Prior work estimates how well the search space induced by a QNN approximates the Hilbert space (Sim et al., 2019) and how good its learning capacity is from a statistical perspective (Berezniuk et al., 2020). The availability of Noisy Intermediate-Scale Quantum (NISQ) devices allows executing quantum circuits of low depth where QNN-based algorithms constitute a huge class. The question regarding how to specify a proper quantum neural architecture is addressed by utilizing certain underlying structure and symmetry of specific problems such as Quantum Approximate Optimization Algorithms (Farhi et al., 2014) or Variational Quantum Eigensolver (Peruzzo et al., 2014). However, necessary domain knowledge is not always available in practice. This highlights the need for a method to automate the QNN design process. Some approaches resort to quantum characteristics to do cell-based search (Ostaszewski et al., 2021), adaptive wavefunction ansatzs (Grimsley et al., 2019; Tang et al., 2021), variable-structure circuit (Bilkis et al., 2021), and hardware-efficient models (**?**Du et al., 2022). Another line of work, on the other hand, borrows ideas from classical neural architecture search (NAS). Representatives for

[*]Equal contribution [1]Department of Mathematical Sciences, KAIST [2]Department of Computer Science, Stanford University [3]Department of Applied Physics, Aalto University [4]Department of Computer Science and Engineering, Ho Chi Minh City University of Technology [5]School of Electrical Engineering, KAIST. Correspondence to: Trong Duong <trongduong@kaist.ac.kr>, Sang Truong <sttruong@cs.stanford.edu>.

2nd AI4Science Workshop at the 39th International Conference on Machine Learning (ICML), 2022. Copyright 2022 by the author(s).



(a) R_Z and R_X (b) R_Z and R_Z^Y (c) R_Z and R_X^Y (d) R_Z^X and R_X^Y

Figure 1: Orbits of single-qubit rotations (rotation direction in the subscript) and oscillatory rotations (oscillation direction in the superscript) when acting on 6 anchor states $|\pm x\rangle, |\pm y\rangle, |\pm z\rangle$ of the one-qubit system. The blue/red line and dots indicate orbit of the first/second gate. (a) R_Z and R_X trace out analogous circles and two clusters at the poles, $d_{\text{core}}(R_Z, R_X) > 0, d_{\text{shape}}(R_Z, R_X) = 0$. (b) Both gates apply rotations around the z-axis, $d_{\text{core}}(R_Z, R_Z^Y) \approx 0$. (c) R_X^Y only has a small deviation from a circle $d_{\text{core}}(R_Z, R_X^Y) \approx d_{\text{core}}(R_Z, R_X), d_{\text{shape}}(R_Z, R_X^Y) \approx 0$. (d) Two oscillatory rotations R_Z^X and R_X^Y trace out similar curves, $d_{\text{core}}(R_Z^X, R_X^Y) \approx d_{\text{core}}(R_Z, R_X), d_{\text{shape}}(R_Z^X, R_X^Y) = 0$.

this class include genetic algorithms (Rattew et al., 2019), meta neural predictor (Zhang et al., 2021), differentiable search (Zhang et al., 2020), and reinforcement learning (Kuo et al., 2021). Many approaches from this class deploy a good final architecture that can even outperform expert's designs. However, those methods behave in a black-box manner and often provide little insight for the quantum view.

In this work we aim to apply Bayesian Optimization (BO), a well-known method for NAS (Kandasamy et al., 2018; Ru et al., 2020), to Quantum NAS (QNAS). General information and setting of QNN and BO will be introduced in Section 2. Applying BO to QNAS requires an appropriate covariance function in space of QNN. The covariance function, or *kernel*, is a key component to the success of the BO problem. Our main contribution, which is presented in Section 3, is an operationally meaningful distance measure between two (possibly parametrized) quantum operators or *quantum gates*, followed by a distance metric in the space of QNN constructed using optimal transport. Experiments of QNAS for three specific problems with the kernel based on that metric are presented in Section 4. The paper finalizes with Section 5 and 6 where we present the result along with some discussions and future work.

2. Preliminaries

Quantum Neural Networks A QNN can be realized by a quantum circuit that contains a sequence of fixed and parametrized unitary operators, so-called *quantum gates* acting on its qubits $q_0, q_1, ..., q_n$, where n is the number of processing qubit. Usually it is restricted to a sequence of fixed gates W_i and one-parameter gates $U_i(\theta_i)$

$$U(\theta) = U_L(\theta_L)W_L \dots U_2(\theta_i)W_2U_1(\theta_1)W_1. \quad (2)$$

A fixed gate can be the *identity* operator, hence two types of gates are not necessarily alternating. For this work we consider the most common logical gates in quantum computing (Nielsen & Chuang, 2010; Gadi, 2019):

$$\begin{aligned} &W_i \in \{H, X, Y, Z; CX, CY, CZ\} \\ &U_i \in \{R_\sigma(\cdot); CR_\sigma(\cdot), R_{\sigma\sigma}(\cdot)\}, \quad \sigma \in \{X, Y, Z\}^{1} \end{aligned} \quad (3)$$

The semicolons separate single-qubit gates in the first half and two-qubit gates in the second half. Natural objective functions for training QNN is usually the expectation value of a Hermitian measurement operator $\mathcal{M}$ evaluated at the final state $U(\theta)|\psi\rangle$ obtained by applying the QNN on an initial state $|\psi\rangle$, denoted by $E(|\psi\rangle, \theta) = \langle\psi| U(\theta)^\dagger \mathcal{M} U(\theta) |\psi\rangle$. Optimizing the objective value can be done with a classical optimization algorithm.

Bayesian Optimization The goal of Bayesian Optimization (BO) is to find the global optimum of a fixed but unknown function $f : \mathcal{X} \to \mathbb{R}$, which is potentially expensive and noisy to evaluate. We assume that f is drawn from a prior distribution $p(f)$. Starting with an arbitrary dataset $\mathcal{D}_0$, BO sequentially queries the function to update our belief about it with guidance from an acquisition function, which measures the increase in utility of having an additional data point $(x, y = f(x) + \epsilon)$ (Brochu et al., 2010). A common choice to model $p(f)$ is through Gaussian Process (GP). A GP describes a collection of random variables with a joint normal distribution identified by a mean function $m : \mathcal{X} \to \mathbb{R}$ and a positive definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (Rasmussen, 2003). The GP can be used as a surrogate model for a black-box function $f(x) \sim \mathcal{GP}(m(x), k(x, X))$, where $X = [x_1, \cdots, x_{|\mathcal{D}|}]^T, Y = [y_1, \cdots, y_{|\mathcal{D}|}]^T$ are the observations we have made so far. With Gaussian prior, the poste-

[1] Fixed gates include Pauli operators X, Y, Z, Hadamard gate $H \cong XZX$, and controlled gates $C\sigma = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes \sigma$. Parametrized gates include rotations $R_\sigma(\theta) = \exp(-i\frac{\theta}{2}\sigma)$, controlled rotations $CR_\sigma(\theta) = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes R_\sigma(\theta)$, and double rotations $R_{\sigma,\sigma}(\theta) = \exp(-i\frac{\theta}{2}\sigma \otimes \sigma)$.

rior predictive $p(y|x, \mathcal{D})$ is a Gaussian with mean function $m(x, X)$ and covariance function $k(x, X)$:

$$\begin{aligned} m(x, X) &= K(x, X)K_X^{-1}(Y - m(X)) \\ k(x, X) &= K(x, x) - K(x, X)K_X^{-1}K(X, x) \\ y &\sim \mathcal{N}(m(x, X), k(x, X)) \end{aligned} \tag{4}$$

where $K(A, B)$ is a matrix whose the (i, j)-th element is computed as $k(x_i, x_j)$ with $x_i \in A$ and $x_j \in B$, and $K_X = K(X, X) + \sigma^2 I$ is the covariance between all noisy observations. The hyperparameters of the model are parameters of the mean and covariance functions and can be learned by maximizing the marginal likelihood of the input data. In this paper we use the Expected Improvement (EI) acquisition function, which measures the expectation of improvement over the current maximum value

$$\mathrm{EI}_t(x) = \mathbb{E}[(f(x) - f_{t-1}^*)^+ | (x_i, y_i)_{i=1}^{t-1}], \tag{5}$$

where $f_{t-1}^* = \max_{i \le t-1} f(x_i)$ is the best value up to time t, and $g(x)^+ = \max(0, g(x))$.

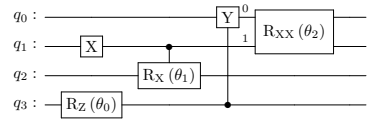
3. Methods

Three representations of a QNN we use are gate-based circuit, directed acyclic network (DAG), and vector representation as shown in Figure 2. The circuit representation is used for optimizing the objective function $\max_\theta E(|\psi\rangle, \theta)$. The DAG representation reveals the topological structure of the QNN. The vector representation only serves as the placeholder of the QNN and will not be used for anything except for sampling steps in BO. So, it should not raise a concern for being numerically arbitrarily chosen.

3.1. Distance Between Two Quantum Gates

We propose a similarity measure between two (possibly parametrized) quantum gates. Although the two unitary operations induced by the gates can be different, their effect might follow similar patterns throughout the Hilbert space. The difference between the exact effects will be captured by *core distance* d_{core}, while the difference between their patterns by shape distance d_{shape}. For example, one-qubit rotation gates are the most common gates in use; applying any rotation gate on a state with many angles always generates a circular orbit, so the shape distance between two rotations is always zero. Yet they revolve around different axes, which results in a positive core distance. Intuition about those two types of distance is illustrated in Fig. 1. We define the distance between two quantum gates to be the average between those two types of distance:

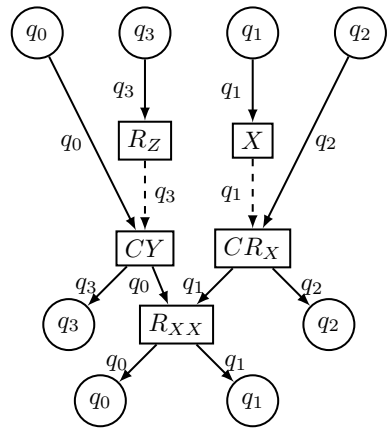
$$d_{\text{gate}}(U^1, U^2) = \frac{d_{\text{core}}(U^1, U^2) + d_{\text{shape}}(U^1, U^2)}{2} \tag{6}$$



(a) Gate-based representation

$$\begin{bmatrix} 0.00 & 0.00 & 0.00 & 0.25 & 0.75 \\ 1.00 & 0.00 & 0.75 & 0.00 & 0.25 \\ 0.00 & 0.00 & 0.25 & 0.00 & 0.00 \\ 0.00 & 1.00 & 0.00 & 0.75 & 0.00 \\ 0.02 & 0.35 & 0.45 & 0.08 & 0.75 \end{bmatrix}$$

(b) Numerical representation



(c) Directed acyclic graph representation

Figure 2: Three representations of a QNN. In general, the numerical representation has size of $(n + 1) \times N$, where n is the number of qubits and N the number of gates. The last row encodes the gate type, where every gate is given a *representative number*, an evenly spaced value in the interval $(0, 0.1)$ for each fixed gate and the interval $(0.1, 1)$ for each parametrized gate. For every gate in the circuit, the row(s) of highest value/two highest values among the first n rows in the corresponding column specifies the qubit the gate acts on. In the mapping from the matrix to the circuit, at each column, we choose the gate such that its representative number is closest to the value in the last row. Depending on its type, the row(s) with highest value(s) specifies qubit argument(s) of the gate. The mapping is not invertible; however it does not matter since the vector representations only stand as continuous placeholders for QNNs during sampling steps. The DAG $\mathcal{G} = (\mathcal{L}, \mathcal{E})$ with a set of gates $\mathcal{L}$ and directed connections between them $\mathcal{E}$. Note that each DAG node is sensitive to the order of incoming edges, which make the DAG and the gate-based circuit representations invertible.

3.1.1. Core Distance

Since every unitary operator is generated by some fixed Hermitian operator H such that $U = e^{iHt}$, we can com-

pare two unitary operators by their generating, assumably fixed, Hermitians. Since it is the solution to $\frac{dU}{dt} = iHU$ and U decides the evolution of quantum states, H governs the evolution over time. For example, $R_X(\cdot)$ and $R_Y(\cdot)$ are rotations about the x- and y-axis. Their corresponding Hermitian operators are Pauli X and Y. Hence it is natural to define a distance of R_X and R_Y using those Hermitian operators. Generally, we propose the difference between two generating Hermitian operators to be the *core distance* between two quantum gates. For a quantum gate U, fixed or parametrized, there exists a unique scalar $t > 0$ and a unique Hermitian operator H such that $\|H\|_* = 1$ and $U = e^{iHt}$. ($\|\cdot\|_*$ denote the nuclear norm; for Hermitian operators, the norm equals to the sum of absolute value of eigenvalues.) The core distance between the two unitary operators is given by

$$d_{\text{core}}(U^1, U^2) = \frac{\|H^1 - H^2\|_*}{2} \tag{7}$$

Since we only consider one-qubit and two-qubit gates, the unitary operator of the entire system of higher dimension is the tensor product of the gate's unitary operator with identity, e.g. $U = U_{\text{gate}} \otimes I$. The core distance is then affected by the qubits two gates apply on, i.e. in general, $d_{\text{core}}(U^1_{\text{gate}} \otimes I, U^2_{\text{gate}} \otimes I)$ is distinct to $d_{\text{core}}(U^1_{\text{gate}} \otimes I, I \otimes U^2_{\text{gate}})$.

3.1.2. Shape Distance

The shape distance represents the difference between orbital traces of two gates. For example, consider a one-qubit system which is represented by a Bloch sphere, the gate $R_X(\theta)$ traces out a circle around the x-axis. The orbit also has a circular shape for the gate $R_Y(\theta)$, except that the circle is around the y-axis. The shape distance should recognize their congruent shapes, i.e. $d_{\text{shape}}(R_X, R_Y) = 0$. This shape comparison approach only makes sense when both gates in the pair are parametrized; in fact, every gate in the set of parametrized gates we choose for our work is a type of rotation parametrized by a single angle θ. At the beginning, ignore the possible parameters, then two unitary operators U^1 and U^2 are the same if $|\langle\psi| U^{2\dagger}U^1 |\psi\rangle|^2 = 1$ for every quantum state $|\psi\rangle$ in the Hilbert space $\mathcal{H}$, i.e.

$$\int_{\mathcal{H}} \left(1 - |\langle\psi| U^{2\dagger}U^1 |\psi\rangle|^2\right) d\mu(\psi) = 0, \tag{8}$$

where the integral is taken with the Haar measure (Meckes, 2019). Simply speaking, it is the uniform distribution over unitary operators U, hence also the uniform distribution over quantum states $|\psi\rangle \equiv U|0\rangle$. To generalize the notion to parametrized quantum gates, think of $U^1(\cdot)|\psi\rangle$ and $U^2(\cdot)|\psi\rangle$ as two functions mapping θ to quantum states. The two gates are said to have the same shape when their codomains obtained as θ varies have the same shape up to a unitary transformation. This allows us to define the shape distance to be the minimal disparity induced by any unitary mapping. That is, two quantum gates are said to trace out the same orbit when for every quantum state $|\psi\rangle \in \mathcal{H}$, there exists a quantum state $|\phi\rangle$ and a unitary operator V such that

$$\left|\langle\phi| U^{2\dagger}(\theta)VU^1(\theta) |\psi\rangle\right|^2 = 1 \quad \forall\theta \tag{9}$$

The shape distance should consider the described fidelity over the Hilbert space, e.g.

$$\int_{\mathcal{H}}\int_{\Theta} \left|\langle\phi| U^{2\dagger}(\theta)VU^1(\theta) |\psi\rangle\right|^2 d\theta d\mu(\psi) \tag{10}$$

We use a collection of quantum states $\{|\psi_k\rangle\}_{k=1}^K$ to approximate the integral. Members of the collection must represent the Hilbert space in some sense. We decide to use a collection called *Mutually Unbiased Bases* (MUB) that contains $K = d(d+1)$ members for a qubit system of dimension $d = 2^n$. A key property of MUB is that its members spread out evenly in the Hilbert space, where two distinct members are either orthonormal or have a fidelity of $|\langle\psi_k|\psi_{k'}\rangle|^2 = 1/d$ (Durt et al., 2010). Moreover, the MUB also forms a complex projective 2-design that reduces an integral involving any unitary operator U over every quantum state with the Haar measure to a finite sum (Klappenecker & Rotteler, 2005)

$$\int_{\mathcal{H}} |\langle\psi| U |\psi\rangle|^2 d\mu(\psi) = \frac{1}{K}\sum_{k=1}^{K} |\langle\psi_k| U |\psi_k\rangle|^2 \tag{11}$$

See Appendix A.2 for further details about how to find that projective 2-design and its use in reduction of integral. We believe the symmetry of a MUB suffices to capture most of the information about the orbit traced out by acting a usual parametrized quantum gate on quantum states across the Hilbert space. Therefore we attempt to define the shape distance motivated by the following optimization problem

$$\begin{aligned}&\min_{V,\phi_k} \frac{1}{KT}\sum_{k,t}\left(1 - \left|\langle\phi_k| U^{2\dagger}(\theta_t)VU^1(\theta_t) |\psi_k\rangle\right|^2\right)\\ =&\min_{V,M} \frac{1}{KT}\sum_{k,t}\left(1 - \left|\langle e_k| M^\dagger U^{2\dagger}(\theta_t)VU^1(\theta_t) |\psi_k\rangle\right|^2\right),\end{aligned} \tag{12}$$

where $M \in \mathbb{C}^{d\times K}$ contains the (normalized) vector representation of every $|\phi_k\rangle$ in its columns, and $\{|e_k\rangle\}$ forms the standard basis of $\mathbb{C}^K$. However, we could not find a good way to optimize the $\|\cdot\|^2$ in (12). Instead we define the shape distance with the root of fidelity

$$\begin{aligned}&d_{\text{shape}}(U^1, U^2)\\ &\equiv \min_{V,M} \frac{1}{KT}\sum_{k,t}\left(1 - \left|\langle\phi_k| U^{2\dagger}(\theta_t)VU^1(\theta_t) |\psi_k\rangle\right|\right)\\ &= \min_{V,M} \frac{1}{KT}\sum_{k,t}\left(1 - \left|\langle e_k| M^\dagger U^{2\dagger}(\theta_t)VU^1(\theta_t) |\psi_k\rangle\right|\right)\end{aligned} \tag{13}$$

Using the fact that $\min_\alpha \|e^{i\alpha}|\psi\rangle - |\phi\rangle\|^2 = 2 - 2|\langle\phi|\psi\rangle|$, where the minimum is achieved at $\alpha = -\measuredangle\langle\psi|\phi\rangle$ and $\measuredangle$ denotes the angle θ in the Euler's representation of a complex number $re^{i\theta}$, we can rewrite the problem with Euclidean distance. The distance, which is called MINT-OREO (Maximum Integral of Orbital Embedding Overlaps), is implemented in the following form

$$\frac{1}{2KT}\min_{V,M,\alpha}\sum_{k,t}\|e^{i\alpha_{k,t}}VU^1(\theta_t)|\psi_k\rangle - U^2(\theta_t)M|e_k\rangle\|^2 \tag{14}$$

It is easy to see that $d_{\text{shape}}(U^1, U^2) = 0$ when both gates are fixed. We assign $d_{\text{shape}}(U^1, U^2) = \infty$ when a gate is fixed and the other is parametrized. If both gates are of parametrized type, we solve the optimization (14) by coordinate descent for $\alpha, V, \alpha, M, \ldots$ until convergence, where each update is optimal with respect to one variable. Finding the optimal unitary V is known as *Orthogonal Procrustes* with a closed-form solution (Gower & Dijksterhuis, 2004). The optimal solution to M can also be evaluated in closed form. We present details of the optimization and a discussion about the difficulty of optimizing the original sum-of-fidelity form in (12) in Appendix B.

The MINT-OREO distance has some notable properties. First, it is invariant to the qubits U^1 and U^2 apply on, i.e. each of the two gates can apply on any qubit(s) of the system and the distance is unchanged. This is because M can absorb the relative difference in the qubits being acted on by the two gates. Second, the distance is invariant to the class of parametrized gates. Let $A = \{R_X, R_Y, R_Z, R_{XX}, R_{YY}, R_{ZZ}\}, B = \{CR_X, CR_Y, CR_Z\}$ are the sets of rotations and controlled rotations. The intra-class distance is always 0 and the inter-class distance is the same for every pair, that is $d(a, a') = 0, d(b, b') = 0$, and $d(a, b) = d_{AB}$ for every $a, a' \in A, b, b' \in B$. We found out by experiments that the property holds for rotation in arbitrary directions. Third, the optimization converges in a few iterations and gives highly consistent results regardless of the initialization of V and M. Also, it does not need to use many parameter samples; the result for $T = 12$ only differs to that of $T = 240$ by $\sim 10^{-3}$. Although we cannot prove it but the numerical result demonstrates that MINT-OREO distance forms a pseudo metric at least among quantum gates of rotation type (rotations and controlled rotations around arbitrary directions). This property still holds when we consider all of the fixed and parametrized gates.

In addition, the shape distance given by the finite sum is 0 when $U_1(\theta)$ and $U_2(\theta)$ generate exactly the same shape. The result is stated formally in the following theorem, whose proof is also presented in Appendix B.

Theorem 1. *The shape distance given by the integral vanishes if and only if the shape distance given by a finite sum over anchor states vanishes. That is, for $T \geq 2$ distinct parameters $\theta_1, \ldots, \theta_T$, if there exist a unitary operator V and quantum states $|\phi\rangle$ such that*

$$\sum_{t=1}^{T}\int_{\mathcal{H}}\left(1 - \left|\langle\phi|U^{2\dagger}(\theta_t)VU^1(\theta_t)|\psi\rangle\right|\right)d\mu(\psi) = 0 \tag{15}$$

if and only if for the same V,

$$\frac{1}{KT}\sum_{k,t}\left(1 - \left|\langle\phi_k|U^{2\dagger}(\theta_t)VU^1(\theta_t)|\psi_k\rangle\right|\right) = 0, \tag{16}$$

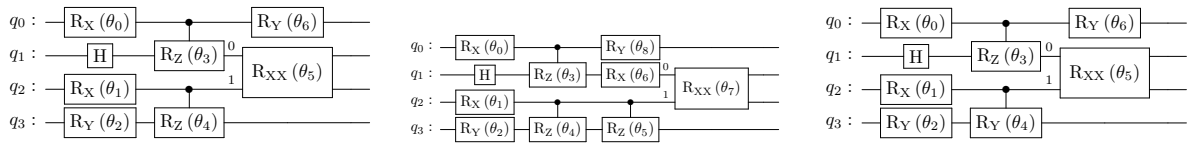
where $|\phi_k\rangle$ is counterpart to $|\psi_k\rangle$ in the same way as $|\phi\rangle$ is counterpart to $|\psi\rangle$.

3.2. Distance Between Two Quantum Neural Networks

The similarity measure between any pair of gates can be generalized into a distance metric between two QNN architectures using optimal transport, whose application in classical NN architecture called OTMANN (Optimal Transport Metrics for Architectures of Neural Networks) was introduced by (Kandasamy et al., 2018). A simple interpretation of optimal transport program is to find a best way to move earth (dirt) piles from sources to destinations (Rubner et al., 2000). With the same analogy, the metric is defined as the minimum of a matching scheme that attempts to match the computation at the gates of one network to the gates of another, where penalties are given when there is any mismatch in the type of gates and topological difference between architectures. The minimum of those penalties is the OTMANN distance. Throughout this section, we view two QNNs as two DAGs $\mathcal{G}_1 = (\mathcal{L}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{L}_2, \mathcal{E}_2)$ with $|\mathcal{L}_1| = n_1$ and $|\mathcal{L}_2| = n_2$ gates.

Gate Mass Gate mass is a number representing the amount of computation a gate can perform. Motivated from the fact that a unitary matrix of size $p \times p$ has $p^2 - 1$ real degrees of freedom, we define the mass parametrized gates as $lm(u) = \text{param-dim}(u) \times (\text{unitary-dim}(u)^2 - 1)$. For example, single-qubit rotation gates have the mass of 3, while controlled rotations and double rotations have the mass of 15. The mass is even shared among gates of the same type if they are consecutive on the wire(s), e.g. two consecutive R_Z gates acting on q_2 and q_3 in Figure **??** each has the mass of $15/2 = 7.5$.

Fixed gates, i.e. gates without parameters, on the other hand, have no value without the presence of parametrized gates. For instance, many quantum algorithms use fixed gates to initialize the quantum system into some favorable state such as a uniform superposition. The power of fixed gates in the circuit depends on the power of parametrized



(a) (b) (c)

Figure 3: An illustration of some similar QNN architectures. The following are unnormalized distance d and normalized distance $\bar{d}$ when $\nu = 0$: $d(a,b) = 3.300, \bar{d}(a,b) = 0.026; d(a,c) = 5.303, \bar{d}(a,c) = 0.042; d(b,c) = 8.603, \bar{d}(b,c) = 0.066$. Circuit (b) has a nonzero distance to (a) because of the extra R_X gate on the q_1 wire. As every other gate in (b) has a perfect match in (a), that completely unmatched R_X gate induces a distance of 3 (its mass) + 0.1×3 (extra mass gate H gets due to the extra R_X). Two CR_Z gates in q_2 and q_3 has the same effect and match to a single CR_Z gate. The circuits (a) and (c) differ by a replacement of a CR_Z gate by CR_Y and has the distance equal to the gate distance of 0.3535 multiplied by the common gate mass of 15. In this case $d(b,c) = d(a,b) + d(a,c)$ because the differing components are independent.

gates in the same circuit. Therefore, we assign them a fixed fraction of the layer mass of parametrized gates, i.e. for each fixed gate u, $lm(u) = \frac{\eta}{|\mathcal{DL}|}\sum_{s\in\mathcal{VL}} lm(s)$, where $\mathcal{DL}$ and $\mathcal{VL}$ are the sets of fixed layers and parametrized layers, respectively, and η is a fixed ratio. We choose $\eta = 0.1$ in our implementation. The total mass of an QNN is $tm(\mathcal{G}_i) = \sum_{u\in\mathcal{L}_i} lm(u)$.

Gate-type Mismatch Cost This is the cost incurred when one attempts to match a gate $u_i \in \mathcal{L}_1$ to another gate $v_j \in \mathcal{L}_2$ of different type. It is natural to choose the gate distance defined in 3.1 for the mismatch cost. Hence the gate-type mismatch cost matrix $C_{\text{gtm}} \in \mathbb{R}^{n_1\times n_2}$ is given by $(C_{\text{gtm}})_{ij} = d_{\text{gate}}(u_i, v_j)$. This matrix can be immediately obtained from a list of precomputed gate-pair distances.

Structural Dissimilarity Cost This term determines how different the relative position of two matched gates in their respective QNNs are. A small $(C_{\text{str}})_{i,j}$ value means the gate $u_i \in \mathcal{L}_1$ and the gate $v_j \in \mathcal{L}_2$ are at structurally similar position. The cost matrix is computed by

$$(C_{\text{str}})_{i,j} = \frac{1}{6n}\sum_{s\in\{\text{sp,lp,avg}\}}\sum_{t\in\{\text{ip,op}\}}\sum_{q=1}^{n}|\delta_t^{s,q}(i) - \delta_t^{s,q}(j)|, \tag{17}$$

where "sp", "lp", "avg", "ip", "op" abbreviate for shortest path, longest path, random walk, input, and output. The value $\delta_t^{s,q}(\cdot)$ measures the shortest/longest/average path length from/to the input/output node of the q-th qubit in the respective DAG circuit. When there is no path between two nodes, we assign to it the longest path length in the entire DAG.

For instance, in Figure 2(c), $\delta^{\text{ip},4}_{\text{sp/lp/avg}}(CY) = 2$ because there is only one path from the input node of q_3 to the CY node and it hops through two nodes including CY. On the other hand $\delta^{\text{op},3}_{\text{sp/lp/avg}}(CY) = 4$ because there is no path from the CY node to q_2, so the longest path length of 4 is assigned.

Optimal Transport Distance Let $\langle\cdot|\cdot\rangle$ denote the component-wise dot product and $\mathbf{1}_n$ an all-one vector. The optimal solution of the Kantorovich's formulation of Optimal Transport defines the distance between two QNNs. Let $\bar{n}_i = n_i + 1; Z, C \in \mathbb{R}^{\bar{n}_1\times\bar{n}_2}; y_1 \in \mathbb{R}^{\bar{n}_1}, y_2 \in \mathbb{R}^{\bar{n}_2}$ such that $\mathbf{1}_{\bar{n}_1}^T y_1 = \mathbf{1}_{\bar{n}_2}^T y_2$

$$\begin{aligned} d(\mathcal{G}_1, \mathcal{G}_2) = \ & \underset{Z}{\text{minimize}}\ \langle R, C\rangle \\ \text{s.t. } & Z \geq 0, Z\mathbf{1}_{\bar{n}_2} = y_1, Z^T\mathbf{1}_{\bar{n}_1} = y_2 \end{aligned} \tag{18}$$

In the formulation, C is the matrix containing pairwise dissimilarity between component gates of the two circuits, or the "ground distance" in the optimal transport literature, and $Z = [z_{ij}]$ is the amount of mass of $u_i \in \mathcal{L}_1$ matched to $v_j \in \mathcal{L}_2$. We set $C = \begin{bmatrix} C_{\text{gtm}} + \nu C_{\text{str}} & \mathbf{1}_{n_1} \\ \mathbf{1}_{n_2}^T & 0 \end{bmatrix}$, $y_1 = \left[\{lm(u)\}_{u\in\mathcal{L}_1}, tm(\mathcal{G}_2)\right]^T$, and $y_2 = \left[\{lm(u)\}_{u\in\mathcal{L}_2}, tm(\mathcal{G}_1)\right]^T$. The last row and column in the cost matrix C are dedicated to a made-up *null gate* in each QNN. The weight of structural cost ν is a hyperparameter for the distance, usually set to 0.1. Matching two gates is subject to gate-type mismatch and structural dissimilarity cost while matching a gate to the null gate induces to a cost of 1, an upper bound for a finite d_{gate} value. The null gate of a circuit is where a leftover mass of any gates in the other is collected when there is no possible match for it.

The linear program can be solved efficiently by many solvers, and the resulting distance is a metric for quantum circuit architectures. The proof of the following theorem can be found in Appendix C. Also, Figure 3 illustrates the distance concept through some similar-looking QNNs.

Theorem 2. *The optimal solution $d(\mathcal{G}_1, \mathcal{G}_2)$ of (18) forms a metric in the space of quantum circuit architectures when the weight of structural cost ν is positive. That is, for any quantum circuits $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$ with described component gates, it holds that $d(\mathcal{G}_1, \mathcal{G}_2) \geq 0$, $d(\mathcal{G}_1, \mathcal{G}_2) = d(\mathcal{G}_2, \mathcal{G}_1)$, $d(\mathcal{G}_1, \mathcal{G}_3) \leq d(\mathcal{G}_1, \mathcal{G}_2) + d(\mathcal{G}_2, \mathcal{G}_3)$, and $d(\mathcal{G}_1, \mathcal{G}_2) = 0 \iff \mathcal{G}_1 = \mathcal{G}_2$.*

3.3. Bayesian Optimization

The kernel we use for two QNNs x and x' in the Gaussian process is given by

$$k(x, x') = \alpha e^{-\sum_i \beta_i d_i(x,x')} + \bar{\alpha} e^{-\sum_i \bar{\beta}_i \bar{d}_i(x,x')}, \quad (19)$$

where d_i and $\bar{d}_i$ are unnormalized and normalized distances with some fixed ν_i. It is important to remark that the two terms are not Gaussian and Laplace kernels although they have the same looking. A counterexample was given in (Peyré et al., 2019) to show the exponential of a negative squared optimal-transport distance does not necessarily induce a positive definite kernel. In our experiments, we, however, never encounter a situation where the covariance matrix has a negative eigenvalue whose absolute value exceeds a small jitter. We specify $\nu_i \in \{0.1, 0.2, 0.4, 0.8\}$ as default hyperparameters. Parameters of the model, including $\alpha, \bar{\alpha}, \beta_i, \bar{\beta}_i$ and the noise variance, are estimated via maximum likelihood.

Since a function on the space of architectures is not differentiable, it is common to use Evolutionary Algorithm (EA) is used to optimize acquisition functions (Kandasamy et al., 2018; White et al., 2021; Ma et al., 2019; Ru et al., 2020). We create a set of N_{mut} samples at each generation by randomly mutating either the type or the acting wire(s) of a random gate in a circuit. Those with high acquisition function values will survive to the next generation with high probability. Although EA is not an efficient optimization algorithm for expensive functions, it works quite well for cheap evaluations of common acquisition functions. Further details are discussed in Appendix E.

4. Experiments

We demonstrate the QNN search for three problems: (i) simulate a Quantum Fourier Transform (QFT) (ii) solve MaxCut for graphs with random edge weights (iii) design a Quantum Generative Adversarial Net (QGAN), i.e. finding QNN architectures that optimize the corresponding black-box objective functions.

First, QFT is a common subroutine in many quantum algorithms. Its unitary operator is $U_{\text{QFT}} = \frac{1}{\sqrt{d}} \sum_{s,t=0}^{d-1} \omega_d^{st} |t\rangle \langle s|$ with system dimension $d = 2^n$ and phases $\omega_d^{st} = 2^{\pi i \frac{st}{d}}$. The QNN $U(\theta)$ learns to simulate the 2-qubit QFT by optimizing $\max_\theta \frac{1}{K} |\langle \psi_k| U_{\text{QFT}}^\dagger U(\theta) |\psi_k\rangle|^2$ given the anchor states $|\psi_k\rangle$ as inputs to the network.

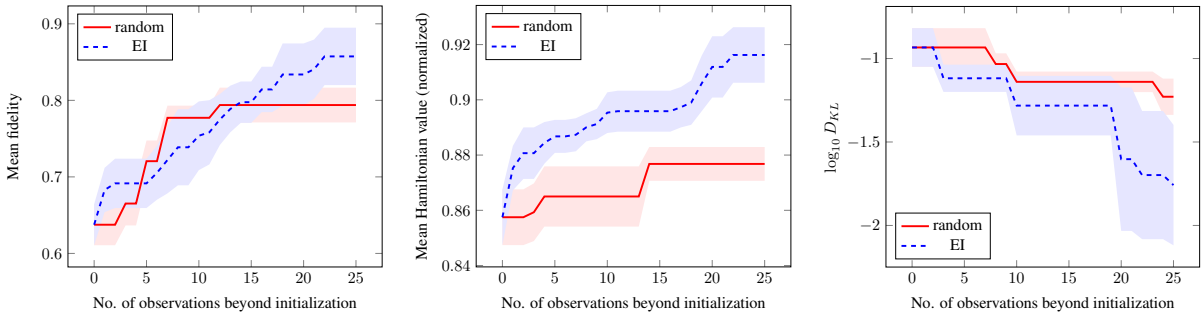
Second, in the MaxCut problem, the QNN learns to separately maximize $M = 10$ Hamiltonians corresponding to M graphs with every edge having weight chosen uniformly random from $\{0, 1, \ldots, 9\}$, i.e. $\frac{1}{M} \sum_{m=1}^{M} \max_\theta \frac{\langle 0|U^\dagger(\theta) H_m U(\theta)|0\rangle}{C_m}$, where H_m is the Hamiltonian as described in (1) and C_m the maximum cut value corresponding to the m-th graph. All the graphs in this experiment have 9 nodes, which also needs 9 qubits. Finding the shape distance in the space of dimension $d = 2^9$ is expensive, so we decide to use the shape distance for $d = 2^4$ for this case because the shape distance of two gate types does not change much as the number of qubits increases.

Quantum Generative Adversarial Network (QGAN) is an adapted version of Generative Adversarial Network (GAN) (Goodfellow et al., 2014) in which either or both the generator and the discriminator is realized by a QNN (Dallaire-Demers & Killoran, 2018; Lloyd & Weedbrook, 2018). While the discriminator is trained to distinguish between true and synthesized data samples, the generator attempts to mimic the true data distribution by fooling the discriminator. In this experiment we train a QGAN that involves a quantum generator against a classical discriminator for learning and loading a probability distribution. In particular we use a 3-qubit QNN generator to generate samples following an even mixture of two normal distributions $\mathcal{N}(0.5, 1^2)$ and $\mathcal{N}(3.5, 0.5^2)$ truncated and discretized into bins $0, \ldots, 7$. The classical discriminator is a simple neural network with three layers as described in (Zoufal et al., 2019). For the QGAN problem, we want to minimize the relative entropy, which is the KL divergence between the generated distribution P and the true distribution Q given by $D_{KL}(P||Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$.

The initialization of all experiments starts by randomly sampling and training 5 architectures. The implementation are made using Qiskit (Gadi, 2019) for quantum circuit design and BoTorch (Balandat et al., 2020) for Bayesian Optimization framework. The setting for training those objectives is presented in Appendix E.

5. Results and Discussion

Figure 4 plots the best objective results for three optimization problems over time that compare the performance of our QNAS algorithm using Expected Improvement ("EI") against random architecture sampling ("random"). We find that the QNAS algorithm performs consistently and significantly better than random sampling, common ansatzs, and problem-specific ansatzs, as presented in Table 1. In the QFT experiment, the algorithm found some circuits with 5 gates that simulate the QFT operator perfectly. They outperform common ansatzs with many more gates and parameters, and requires less gates (6 gates) than the most



(a) QFT (b) MaxCut (c) QGAN

Figure 4: Performance results. QFT: 2 qubits, 6 gates, 6 trials. MaxCut: 9 qubits, 5 gates, 6 trials. QGAN: 3 qubits, 12 gates, 3 trials. The y axes show the optimal objective values for learning problems. Confident band represents $\pm 1\sigma$.

Methods	QFT	MaxCut	QGAN
Bayes. Opt.			
EI	**1**	**0.94**	**0.002**
random	0.88	0.90	0.065
19 az. (Sim et al., 2019)			
depth 1	0.98		0.007
depth 2	1		
MaxCut az. (Farhi et al., 2014)			
depth 1		0.746	
depth 2		0.751	
depth 3		0.762	
QGAN az. (Zoufal et al., 2019)			
depth 1	0.7		0.476
depth 2	0.7		0.277
depth 3	0.7		0.092

Table 1: Performance of QNN obtained by four ways on three experiments. Regarding methods, we report the best objective value over all searched architectures for BO and over 19 ansatz (az.) templates. As MaxCut and QGAN ansatzs are fixed, we report the average value after a number of training trials. A field is omitted when it is inconvenient or highly time-consuming to use the combination.

optimized QFT circuit (7 gates) in Qiskit using the same pool of component gates. In the MaxCut experiment, architectures found by BO perform better than the MaxCut ansatz and random architectures by a large margin. It might be because the ansatz is a trotterization of the Hamiltonian which, for a high-dimension system, might require a large depth to converge; otherwise it lacks the flexibility for learning (Zhou et al., 2020). This suggests we could use simpler circuits to find approximate solution to hard optimization problems. Finally, we found circuits with 12 gates that has $D_{KL} \approx 10^{-3}$ in the QGAN experiment. The original paper use a circuit template with 24 gates and longer training time to achieve the same precision (Zoufal et al., 2019).

6. Conclusion and Future Work

We presented a BO algorithm for quantum neural architecture search, which could find good architectures when the the desired output states are not very restrictive. The main contribution is a geometrically meaningful distance between quantum gates and the extension to distance between quantum circuit architectures using optimal transport. The gate distance and circuit distance could have independent usages beyond BO algorithms. A possible direction is to examine the relation between the gate/circuit distance and learning behaviors of QNNs. Another direction is to design the generalization of the core distance when the Hamiltonian is not fixed and the shape distance when the parameter spaces for U^1 and U^2 are distinct , which could provide a way to analyze data generated by quantum processes.

References

Balandat, M., Karrer, B., Jiang, D. R., Daulton, S., Letham, B., Wilson, A. G., and Bakshy, E. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Advances in Neural Information Processing Systems 33*, 2020. doi: 10.48550/arXiv.1910.06403.

Berezniuk, O., Figalli, A., Ghigliazza, R., and Musaelian, K. A scale-dependent notion of effective dimension. *arXiv preprint arXiv:2001.10872*, 2020. doi: 10.48550/ARXIV.2001.10872.

Bilkis, M., Cerezo, M., Verdon, G., Coles, P. J., and Cincio, L. A semi-agnostic ansatz with variable structure for quantum machine learning. *arXiv preprint arXiv:2103.06712*, 2021. doi: 10.48550/arXiv.2103.06712.

Brochu, E., Cora, V. M., and De Freitas, N. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010. doi: 10.48550/arXiv.1012.2599.

Dallaire-Demers, P.-L. and Killoran, N. Quantum generative adversarial networks. *Phys. Rev. A*, 98, 2018. doi: 10.1103/PhysRevA.98.012324.

Du, Y., Huang, T., You, S., Hsieh, M.-H., and Tao, D. Quantum circuit architecture search for variational quantum algorithms. *npj Quantum Information*, 8, 2022.

Durt, T., Englert, B.-G., Bengtsson, I., and Życzkowski, K. On mutually unbiased bases. *International journal of quantum information*, 8(04), 2010. doi: 10.1142/S0219749910006502.

Farhi, E., Goldstone, J., and Gutmann, S. A quantum approximate optimization algorithm, 2014.

Gadi, A. e. a. Qiskit: An open-source framework for quantum computing, January 2019.

Garey, M., Johnson, D., and Stockmeyer, L. Some simplified np-complete problems. *Theoretical Computer Science*, 1 (3), 1976. doi: 10.1016/0304-3975(76)90059-1.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. doi: 10.1145/3422622.

Gower, J. C. and Dijksterhuis, G. B. *Procrustes problems*, volume 30. OUP Oxford, 2004. doi: 10.1093/acprof:oso/9780198510581.001.0001.

Grimsley, H. R., Economou, S. E., Barnes, E., and Mayhall, N. J. An adaptive variational algorithm for exact molecular simulations on a quantum computer. *Nature communications*, 10(1), 2019. doi: 10.1038/s41467-019-10988-2.

Kandasamy, K., Neiswanger, W., Schneider, J., Póczos, B., and Xing, E. P. Neural architecture search with bayesian optimisation and optimal transport. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2018. doi: 10.5555/3326943.3327130.

Klappenecker, A. and Rötteler, M. Constructions of mutually unbiased bases. Springer Berlin Heidelberg, 2004. doi: 10.1007/978-3-540-24633-6_10.

Klappenecker, A. and Rotteler, M. Mutually unbiased bases are complex projective 2-designs. In *Proceedings. International Symposium on Information Theory, 2005. ISIT 2005*. IEEE, 2005. doi: 10.1109/ISIT.2005.1523643.

Kuo, E.-J., Fang, Y.-L. L., and Chen, S. Y.-C. Quantum architecture search via deep reinforcement learning. *arXiv preprint arXiv:2104.07715*, 2021. doi: 10.48550/arXiv.2104.07715.

Lloyd, S. and Weedbrook, C. Quantum generative adversarial learning. *Phys. Rev. Lett.*, 121, 2018. doi: 10.1103/PhysRevLett.121.040502.

Ma, L., Cui, J., and Yang, B. Deep neural architecture search with deep graph bayesian optimization. In *2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. IEEE, 2019. doi: 10.1145/3350546.3360740.

Mead, A. Review of the development of multidimensional scaling methods. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 41(1), 1992. doi: 10.2307/2348634.

Meckes, E. S. *The random matrix theory of the classical compact groups*, volume 218. Cambridge University Press, 2019. doi: 10.1017/9781108303453.

Nielsen, M. A. and Chuang, I. L. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. doi: 10.1017/CBO9780511976667.

Ostaszewski, M., Grant, E., and Benedetti, M. Structure optimization for parameterized quantum circuits. *Quantum*, 5, 2021. doi: 10.22331/q-2021-01-28-391.

Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.-H., Zhou, X.-Q., Love, P. J., Aspuru-Guzik, A., and O'Brien, J. L. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5(1), 2014. doi: 10.1038/ncomms5213.

Peyré, G., Cuturi, M., et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6), 2019. doi: 10.1561/2200000073.

Rasmussen, C. E. Gaussian processes in machine learning. In *Summer school on machine learning*. Springer, 2003. doi: 10.1007/978-3-540-28650-9_4.

Rattew, A. G., Hu, S., Pistoia, M., Chen, R., and Wood, S. A domain-agnostic, noise-resistant, hardware-efficient evolutionary variational quantum eigensolver. *arXiv preprint arXiv:1910.09694*, 2019. doi: 10.48550/arXiv.1910.09694.

Ru, B., Wan, X., Dong, X., and Osborne, M. Interpretable neural architecture search via bayesian optimisation with weisfeiler-lehman kernels. In *8th International Conference on Learning Representations*, volume abs/2006.07556. arXiv, 2020. doi: 10.48550/arXiv.2006.07556.

Rubner, Y., Tomasi, C., and Guibas, L. J. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2), 2000. doi: 10.1023/A:1026543900054.

Sim, S., Johnson, P. D., and Aspuru-Guzik, A. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies*, 2(12), 2019. doi: 10.1002/qute.201900070.

Tang, H. L., Shkolnikov, V., Barron, G. S., Grimsley, H. R., Mayhall, N. J., Barnes, E., and Economou, S. E. qubit-adapt-vqe: An adaptive algorithm for constructing hardware-efficient ansätze on a quantum processor. *PRX Quantum*, 2(2), 2021. doi: 10.1103/PRXQuantum.2.020310.

White, C., Neiswanger, W., and Savani, Y. Bananas: Bayesian optimization with neural architectures for neural architecture search. In *AAAI*, 2021. doi: 10.48550/arXiv.1910.11858.

Zhang, S.-X., Hsieh, C.-Y., Zhang, S., and Yao, H. Differentiable quantum architecture search. *arXiv preprint arXiv:2010.08561*, 2020. doi: 10.48550/arXiv.2010.08561.

Zhang, S.-X., Hsieh, C.-Y., Zhang, S., and Yao, H. Neural predictor based quantum architecture search. *Machine Learning: Science and Technology*, 2(4), 2021. doi: 10.1088/2632-2153/ac28dd.

Zhou, L., Wang, S.-T., Choi, S., Pichler, H., and Lukin, M. D. Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. *Physical Review X*, 2020.

Zoufal, C., Lucchi, A., and Woerner, S. Quantum generative adversarial networks for learning and loading random distributions. *npj Quantum Information*, 5(1), 2019. doi: 10.1038/s41534-019-0223-2.

A. Mutually Unbiased Bases

A.1. Complex Projective 2-design

Reducing the integral over the Haar measure to a simple sum of terms of the same form, as in (11) is possible given the right representatives of the space. The collection of such representatives of the unitary space is called a quantum unitary design.

Definition 1. *Let $P_{t,t} : \mathcal{U}(\mathcal{H}) \to \mathbb{R}$ be a polynomial with homogeneous degree at most t in the entries of $W \in \mathcal{U}(\mathcal{H})$ and their complex conjugate. A finite set of unitary operators $\{W_k\}_1^K$ forms a unitary t-design when*

$$\frac{1}{K}\sum_{k=1}^{K} P_{t,t}(W_k) = \int_{\mathcal{U}(\mathcal{H})} P_{t,t}(W) d\mu(W), \tag{20}$$

where μ is the Haar measure, holds for all such polynomials $P_{t,t}$.

For any unitary t-design $\{W_k\}$ and any fixed pure state $|\psi\rangle$, the set $\{W_k |\psi\rangle\}$ is a complex projective t-design.

Since the expression is exact, and the integrand $|\langle\psi| U |\psi\rangle|^2 \equiv |\langle 0| W^\dagger U W |0\rangle|^2$ for some W in (11) is a polynomial of degree 2 in entries of W and $W^\dagger$, it can be computed given a unitary 2-design or a complex projective design. It is known from quantum information that the Clifford group forms a unitary 3-design that also makes it a 2-design by definition. However, the size of the group grows exponentially with the dimension of the Hilbert space. Another option is Mutually Unbiased Bases (MUB), which is a complex projective 2-design and only has $d(d+1)$ element vectors when $\mathcal{H} \subset \mathbb{C}^d$. An efficient construction of the MUB when the dimension is a power of a prime based on Galois field and Galois ring introduced by (Klappenecker & Rötteler, 2004) is briefly introduced.

A.2. Construction of Mutually Unbiased Bases for Prime Power Dimension Space

Mutually unbiased bases in the Hilbert space $\mathbb{C}^d$ are orthonormal bases in which every two bases $\{|e_1\rangle, \ldots, |e_d\rangle\}$ and $\{|f_1\rangle, \ldots, |f_d\rangle\}$ satisfy

$$|\langle e_j | f_k\rangle|^2 = \frac{1}{d}, \quad \forall j, k \tag{21}$$

When d is a power of a prime number, a set of MUB contain exactly $d+1$ bases, and $d(d+1)$ basis vectors form a complex projective 2-design (Klappenecker & Rotteler, 2005).

A.2.1. Odd prime power $d = p^n$

Denote $\mathbb{F}_d$ a finite field with d elements of odd characteristic p. The field trace map $\mathrm{tr} : \mathbb{F}_d \to \mathbb{F}_p$ is defined by $\mathrm{tr}(x) = x + x^p + x^{p^2} \cdots + x^{p^{n-1}}$.

For every $a \in \mathbb{F}_d$, the set

$$\begin{aligned} B_a &= \{|v_{a,b}\rangle : b \in \mathbb{F}_d\}, \\ |v_{a,b}\rangle &= d^{-1/2}(\omega_p^{\mathrm{tr}(ax^2+bx)})_{x\in\mathbb{F}_d} \in \mathbb{C}^d, \end{aligned} \tag{22}$$

where $\omega_p = \exp(2\pi i/p)$, is an orthonomal basis. Then the bases $B_a, a \in \mathbb{F}_q$ together with the standard basis form a set of $d+1$ mutually unbiased bases in $\mathbb{C}^d$.

A.2.2. Even Prime Power $d = 2^n$

The method for odd prime power does not work in the case $p = 2$, which is our interest for qubit systems. For a monic basic primitive polynomial $h(x)$ of degree n, the quotient ring $\mathrm{GR}(4, n) = \mathbb{Z}_4[x]/\langle h(x)\rangle$ is the Galois ring of degree n over $\mathbb{Z}_4$ (since all Galois ring constructed by different $h(x)$ are indeed isomorphic). There exists a nonzero element ξ in $\mathrm{GR}(4, n)$ with order $2^n - 1$. Any element $r \in \mathtt{GR}(4, n)$ has a unique expression $r = a + 2b$, where $a, b \in \mathcal{T}_n = \{0, 1, \xi, \ldots, \xi^{2^n-2}\}$. The `Teichmüller set` $\mathcal{T}_n$ allows us to characterize elements of the ring in terms of a, b. The trace map $\mathrm{tr} : \mathrm{GR}(4, n) \to \mathbb{Z}_4$ is defined by $\mathrm{tr}(x) = \sum_{k=0}^{n-1} \sigma^k(x)$, where $\sigma : \mathrm{GR}(4, n) \to \mathrm{GR}(4, n)$ defined by $\sigma(a + 2b) = a^2 + 2b^2$ is called the Frobenius automorphism. Then the orthonormal bases $B_a, a \in \mathcal{T}_n$ together with the standard basis form a collection of MUB, where

$$\begin{aligned} B_a &= \{|v_{a,b}\rangle : b \in \mathcal{T}_n\}, \\ |v_{a,b}\rangle &= d^{-1/2}\left(\exp\left(\frac{2\pi i}{4}\mathrm{tr}((a+2b)x)\right)\right)_{x\in\mathcal{T}_n} \end{aligned} \tag{23}$$

B. Maximum Integral of Orbital Embedding Overlaps (MINT-OREO) Distance

B.1. Optimizing the shape distance

The optimization problem (14) can be written in two forms

$$\min_{V,M} \sum_{k,t} 1 - \left|\langle e_k| M^\dagger U^{2\dagger}(\theta_t) V U^1(\theta_t) |\psi_k\rangle\right| \tag{24}$$

$$= \frac{1}{2} \min_{V,M,\alpha} \sum_{k,t} \| e^{i\alpha_{k,t}} V U^1(\theta_t) |\psi_k\rangle - U^2(\theta_t) M |e_k\rangle \|^2 \tag{25}$$

$$= \frac{1}{2} \min_{V,M,\alpha} \sum_{k,t} \| e^{i\alpha_{k,t}} U^{2\dagger}(\theta_t) V U^1(\theta_t) |\psi_k\rangle - M |e_k\rangle \|^2 \tag{26}$$

It can be solved by optimizing $\alpha, V, \alpha, M, \ldots$ iteratively, where each iteration constitutes those four updates. We use the expression (25) to optimize for α, V. Denote

$$\begin{aligned} L_1 &= [e^{i\alpha_{1,1}} U^1(\theta_1) |\psi_1\rangle \ldots e^{i\alpha_{K,T}} U^1(\theta_T) |\psi_K\rangle], \\ L_2 &= [U^2(\theta_1) M |e_1\rangle \ldots U^2(\theta_T) M |e_K\rangle] \end{aligned} \tag{27}$$

the matrices in $\mathbb{C}^{d\times KT}$ that contains all outputs resulted from applying U^1, U^2. Note that L_1, L_2 contains K groups of consecutive columns with $t = 1, \cdots, T$ for each k. Let $L_1 L_2^\dagger = R\Sigma T^\dagger$ be a singular value decomposition (after update α). The variables α and V are updated by

$$\begin{aligned} \alpha_{k,t} &\longleftarrow -\measuredangle \langle e_k| M^\dagger U^{2,\dagger}(\theta_t) V U^1(\theta_t) |\psi_k\rangle \\ V &\longleftarrow R^\dagger T \end{aligned} \tag{28}$$

The first rule results from the fact $\min_\alpha \| e^{i\alpha} |\phi\rangle - |\psi\rangle \|^2 = 2 - |\langle\phi|\psi\rangle|$ at $\alpha = -\measuredangle \langle\psi|\phi\rangle$, where $\measuredangle$ denotes the angle in the Euler's representation of a complex number. We can rewrite (25) as $\frac{1}{2}\|VL_1 - L_2\|_F^2$, whose minimization is equivalent to maximizing $\mathrm{Re}\,\mathrm{Tr}\left[L_2^\dagger V L_1\right] = \mathrm{Re}\,\mathrm{Tr}\left[\Sigma(T^\dagger V R)\right]$. It reaches the maximum of $\mathrm{Tr}(\Sigma)$ if $T^\dagger V R = I$ or $V = R^\dagger T$, which is the second rule.

We use the expression (26) to optimize for α, M

$$\begin{aligned} \alpha_{k,t} &\longleftarrow -\measuredangle \langle e_k| M^\dagger U^{2,\dagger}(\theta_t) V U^1(\theta_t) |\psi_k\rangle \\ M &\longleftarrow [\hat{M}_1 \ldots \hat{M}_k \ldots \hat{M}_K] \end{aligned} \tag{29}$$

Denote $S = [\ldots e^{i\alpha_{k,t}} U^{2\dagger}(\theta_t) V U^1(\theta_t) |\psi_k\rangle \ldots]$ and $E = [|e_1\rangle\ (T \text{ times}), \ldots, |e_K\rangle\ (T \text{ times})]$ with standard basis vectors $\{|e_k\rangle\}$ of $\mathbb{C}^K$. After α is updated, the problem in (26) is equivalent to minimizing $\|S - ME\|_F^2$. We can optimize every column of M independently. Let M_k be its k-th column and $S_{k,1}, \cdots S_{k,T}$ be consecutive columns of S. The optimization for a column of M can be formulated as

$$\min_{M_k} \sum_{t=1}^{T} \|M_k - S_{k,t}\|^2 \text{ s.t. } \|M_k\|^2 = 1 \tag{30}$$

It is straightforward that the problem is equivalent to $\max_{M_k} \mathrm{Re}\,(M_k^\dagger S_k)$ for $S_k = \sum_{t=1}^T S_{k,t}$. The optimal value for the column is $\hat{M}_k = \frac{S_k}{\|S_k\|}$ obtained from the fact that

$$\mathrm{Re}\,(x^\dagger y) = \begin{bmatrix} \mathrm{Re}\, x^T & \mathrm{Im}\, x^T \end{bmatrix} \begin{bmatrix} \mathrm{Re}\, y \\ \mathrm{Im}\, y \end{bmatrix}, \tag{31}$$

which is maximized when $x = \lambda y, \lambda \in \mathbb{R}$.

B.2. Finite sum characterizes integral over the Hilbert space

Now we prove Theorem. 1, which could be simplified to

$$\frac{1}{T}\sum_{t=1}^{T}\int_{\mathcal{H}}\left|\langle\phi|\,U^{2\dagger}(\theta_t)VU^1(\theta_t)\,|\psi\rangle\right|d\mu(\psi)=1 \tag{32}$$

if and only if

$$\left|\langle\phi_k|\,U^{2\dagger}(\theta_t)VU^1(\theta_t)\,|\psi_k\rangle\right|=1\quad\forall k,t \tag{33}$$

The forward direction is straightforward due to the continuity of $|\phi\rangle$. For the reverse statement, choose $t=1$ as real phase references, i.e. $|\phi_k\rangle=U^{2\dagger}(\theta_1)VU^1(\theta_1)\,|\psi_k\rangle$ for every k. For every $t>1$, $|\phi_k\rangle=e^{-i\alpha_{kt}}U^{2\dagger}(\theta_t)VU^1(\theta_t)\,|\psi_k\rangle$ for some real α_{kt}. The identity $\langle\phi_k|\phi_k\rangle=1$ implies that

$$\langle\psi_k|\left(U^{2\dagger}(\theta_1)VU^1(\theta_1)\right)^\dagger\left(U^{2\dagger}(\theta_t)VU^1(\theta_t)\right)|\psi_k\rangle=e^{i\alpha_{kt}} \tag{34}$$

From the construction of MUB, the set of anchor states contains the standard basis $\{|e_m\rangle\}_{i=m}^d$. Without loss of generality, assume $|\psi_m\rangle=|e_m\rangle\,,m=1,\ldots,d$. Since the action of a linear operator is completely defined with a basis, we can rewrite

$$\left(U^{2\dagger}(\theta_1)VU^1(\theta_1)\right)^\dagger\left(U^{2\dagger}(\theta_t)VU^1(\theta_t)\right)|e_m\rangle=e^{i\alpha_{mt}}\,|e_m\rangle \tag{35}$$

For ease of writing, denote $W_t=U^{2\dagger}(\theta_t)VU^1(\theta_t)$. Let $|\phi\rangle=W(\theta_1)\,|\psi\rangle$. The goal is to show $|\langle\phi|\,W(\theta)\,|\psi\rangle|=1$ for every $t=1,\ldots,T$ and $|\psi\rangle\in\mathcal{H}$.

Let $|\psi\rangle=\sum_{m=1}^d c_m\,|e_m\rangle\,,|\psi_k\rangle=\sum_{m=1}^d c_{k,m}\,|e_m\rangle$ be normalized expressions in the standard basis. Then

$$\langle\phi|\,W_t\,|\psi\rangle=\langle\psi|\,W_1^\dagger W_t\,|\psi\rangle=\left|\sum_{m=1}^d|c_m|^2e^{i\alpha_{mt}}\right| \tag{36}$$

For a fixed t, the phases $\{e^{i\alpha_{mt}}\}$ are not independent. From the first d anchors we can already determine $e^{i\alpha_{mt}}$ for all m. However, the other d^2 anchors constrain $e^{i\alpha_{mt}}$ as

$$\langle\phi_k|\,W_t\,|\psi_k\rangle|=1\longrightarrow\left|\sum_{m=1}^d|c_{k,m}|^2e^{i\alpha_{mt}}\right|=1 \tag{37}$$

The above constraints are equivalent to d^2 linear equations in d^2 variables $e^{-i\alpha_{mt}}e^{i\alpha_{nt}},m,n=1,\ldots,d$. It has at most one solution, which occur when $e^{i\alpha_{mt}}=e^{i\alpha_{1t}}$. This completes the proof as $\left|\sum_{m=1}^d|c_m|^2e^{i\alpha_{mt}}\right|=\left|\sum_{m=1}^d|c_m|^2e^{i\alpha_{1t}}\right|=1$.

B.3. Remark on L_2 formulation of shape distance

Back to the formulation for the shape distance, although we have not found an efficient way to optimize the sum-of-fidelity formulation

$$\max_{V,M}\sum_{k=1}^K\sum_{t=1}^T|\langle e_k|\,M^\dagger U^{2\dagger}(\theta_t)VU^1(\theta_t)\,|\psi_k\rangle\,|^2, \tag{38}$$

it could have a meaning in quantum information beyond our initial motivation. To optimize V, we extend $|\psi_k\rangle\to|\psi_k\rangle\otimes|k\rangle\otimes 1,M\,|e_k\rangle\to(M\,|e_k\rangle)\otimes|k\rangle\otimes 1,U^i(\theta_t)\to U^i(\theta_t)\otimes 1\otimes|t\rangle$ and $V\to V\otimes I\otimes I$. Define $L_1=[U^1(\theta_1)\,|\psi_1\rangle\ldots U^1(\theta_T)\,|\psi_K\rangle]$ and $L_2=[U^2(\theta_1)M\,|e_1\rangle\ldots U^2(\theta_T)M\,|e_K\rangle]$. We can rewrite (38) using Frobenius norm

$$\begin{aligned}&\sum_{k=1}^K\sum_{t=1}^T|\langle e_k|\,M^\dagger U^{2\dagger}(\theta_t)VU^1(\theta_t)\,|\psi_k\rangle\,|^2\\=&\|L_2^\dagger(V\otimes I\otimes I)L_1\|_F^2\\=&\mathrm{Tr}\left(W_2^\dagger\Sigma_2^\dagger R_2^\dagger(V\otimes I\otimes I)R_1\Sigma_1\Sigma_1^\dagger R_1^\dagger(V^\dagger\otimes I\otimes I)R_2\Sigma_2W_2\right)\\=&\mathrm{Tr}\left(\rho_2(V\otimes I\otimes I)\rho_1(V^\dagger\otimes I\otimes I)\right)\end{aligned} \tag{39}$$

One can view $\rho_i = R_i D_i R_i^\dagger$ as a density matrix consisting of a probe system and the environment, where diagonal matrices D_i that contain nonnegative eigenvalues of $L_i L_i^\dagger$. The optimization problem $\max_V \text{Tr}\left(\rho_2(V \otimes I \otimes I)\rho_1(V^\dagger \otimes I \otimes I)\right)$ could be interpreted as finding the best unitary mapping V of the system to map ρ_1 to ρ_2, both of which are quantum states of the system and the environment.

To optimize M, we follow the same idea as above by extending our system as followed $|e_k\rangle \rightarrow |e_k\rangle \otimes |k\rangle, U^2(\theta_t)VU^1(\theta_t)|\psi_k\rangle \rightarrow \left(U^2(\theta_t)VU^1(\theta_t)|\psi_k\rangle\right) \otimes |k\rangle$ and $M \rightarrow M \otimes I$. We define $L_2' = [|e_1\rangle \dots |e_K\rangle]$ and $L_1' = [U^2(\theta_1)VU^1(\theta_1)|\psi_1\rangle \dots U^2(\theta_T)VU^1(\theta_T)|\psi_K\rangle]$. Using Frobenius norm, the maximization problem over M can be casted as

$$\max_M \sum_{k=1}^{K}\sum_{t=1}^{T} |\langle e_k| M^\dagger U^{2\dagger}(\theta_t)VU^1(\theta_t)|\psi_k\rangle| = \max_M \text{Tr}\left(\rho_2'(M^\dagger \otimes I)\rho_1'(M \otimes I)\right), \tag{40}$$

This problem has a similar form as optimizing V except M is only constrained to have normalized columns.

C. Optimal-Transport Distance as a Metric

The optimal-transport distance is non-negative and symmetric straightforward from its definition and constraints. Now we alter the settings to prove the triangle inequality. Let $tm(\mathcal{G}_i), i = 1, 2, 3$ be the total mass of $\mathcal{G}_i, i = 1, 2, 3$. Because transporting masses between two null gates takes no cost, the solution to (18) is the same with $y_i = \left[\{lm(u)\}_{u\in\mathcal{L}_i}, \sum_{j\neq i} tm(\mathcal{G}_j)\right]^T$. This is to make the total mass $\sum_{i=1}^3 tm(\mathcal{G}_i)$ for all three pairs, hence without generality we prove the triangle inequality assuming the total mass is 1. We follow the classic proof in (Rubner et al., 2000). Let $\{z_{ij}^{(12)}\}$ be the optimal flow from $\mathcal{G}_1$ to $\mathcal{G}_2$ and $\{z_{jk}^{(23)}\}$ the optimal flow from $\mathcal{G}_2$ to $\mathcal{G}_3$, i.e. the solutions to the corresponding optimal transport problems. Let s_{ijk} be the mass that moves from gate $u_i \in \mathcal{L}_1$ to gate $v_j \in \mathcal{L}_2$ and from $v_j \in \mathcal{L}_2$ to $w_k \in \mathcal{L}_3$. Define a flow $z^{(13)ik}$ from $\mathcal{G}_1$ to $\mathcal{G}_3$ by

$$z_{ik}^{(13)} = \sum_j s_{ijk} \tag{41}$$

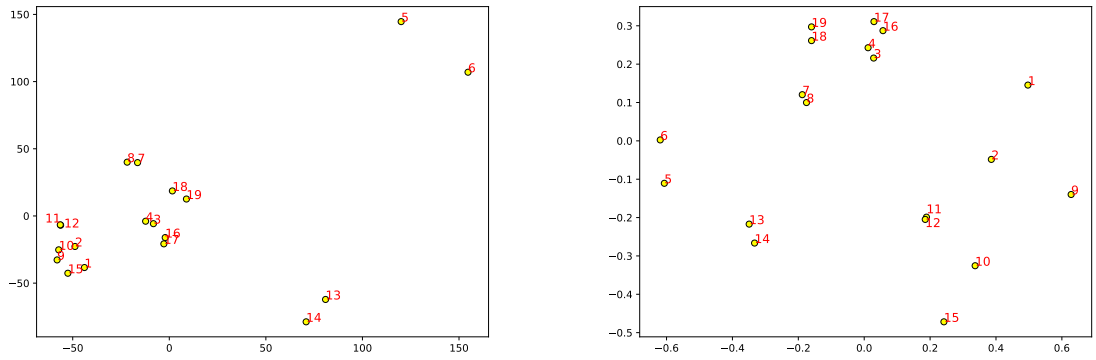
This flow is feasible since

$$\begin{aligned}
\sum_i z_{ik}^{(13)} &= \sum_{i,j} s_{ijk} = \sum_j z_{jk}^{(23)} = lm(w_k), \\
\sum_k z_{ik}^{(13)} &= \sum_{j,k} s_{ijk} = \sum_j z_{ij}^{(12)} = lm(u_i)
\end{aligned} \tag{42}$$

where $lm(u_i)$ ($lm(w_k)$) is the gate mass if u_i (w_k) is a quantum gate or $tm(\mathcal{G}_2) + tm(\mathcal{G}_3)$ ($tm(\mathcal{G}_1) + tm(\mathcal{G}_3)$) if it is a null gate. Then

$$\begin{aligned}
d(\mathcal{G}_1, \mathcal{G}_3) &\leq \sum_{i,k} z_{ik}^{(13)} c_{u_i,w_k} \\
&\leq \sum_{i,j,k} s_{ijk} c_{u_i,w_k} \\
&\leq \sum_{i,j,k} s_{ijk}(c_{u_i,v_j} + c_{v_j,w_k}) \\
&= \sum_{i,j} z_{ij}^{(12)} c_{u_i,v_j} + \sum_{j,k} z_{jk}^{(23)} c_{v_j,w_k} \\
&= d(\mathcal{G}_1, \mathcal{G}_2) + d(\mathcal{G}_2, \mathcal{G}_3)
\end{aligned} \tag{43}$$

The inequality in the third line comes from the fact that the cost between gates, including null ones, is metric. Because the cost is a weighted sum of core distance, shape distance, structural dissimilarity cost, and non-assignment cost, it suffices to show each component bears similar properties. The core distance with nuclear norm is a metric. Numerical results of the shape distance show it is a pseudo-metric for our set of gates as remarked in Section 3. Regarding the non-assignment cost, if there is at least a null gate among u_i, v_j, w_k, the costs that involve the null gate are either 0 or 1 which leads to the



(a) MDS: Circuit Distance

(b) MDS: Normalized Circuit Distance

Figure 5: The two dimensional embedding from pairwise (normalized) distance with $\nu = 0.5$ of 19 circuit templates. The circuits are labeled with the same ID shown in Figure 2 in (Sim et al., 2019). Circuits with notable similarity are close to each other such as (3,4), (5,6), (7,8), (11,12), (13,14), (16,17), (18,19). Remaining circuits (1,2,9,10,15) have fewer parametrized gates, hence small total mass and are placed within a neighborhood.

triangle inequality; otherwise the term for non-assignment cost disappears. The structural dissimilarity cost satisfies the triangle inequality because the following holds trivially for real numbers $\delta_t^{s,q}(\cdot)$:

$$|\delta_t^{s,q}(i) - \delta_t^{s,q}(k)| \leq |\delta_t^{s,q}(i) - \delta_t^{s,q}(j)| + |\delta_t^{s,q}(j) - \delta_t^{s,q}(k)| \tag{44}$$

The optimal-transport distance is therefore a metric in the strict sense. Two quantum circuits must be the same when they have zero distance. Every gate in one circuit then must have an exact counterpart gate in the other, where they have the same relative position to all of the qubit wires and are of the same gate type. This could only happen when the two circuits are identical.

D. Illustration of Quantum Circuits Metric

First, we show the circuit metric as in (18) agree with our intuition through a visualization of pairwise distances of some common circuit architectures using Multidimensional Scaling (MDS). The MDS assigns each architecture a point $(x_1, \ldots, x_d) \in \mathbb{R}^d$ so that two points are close to each other if the distance between the two corresponding is small (Mead, 1992). Figure 5 shows a two-dimensional embedding of 19 circuit templates with 4 qubits surveyed in literature (Sim et al., 2019).

Second, we demonstrate that the circuit metric aligns with the performance of QNNs in a simple training task. We examine 300 pairs of random QNNs with 3 qubits and a maximum of 20 gates. The QNNs are trained to mimic the action of a Quantum Fourier Transform (QFT) by maximizing the fidelity between the output states with ideal outcome states with the anchor states $|\psi_k\rangle$ as the input states, $\max_\theta \frac{1}{K} \| \langle \psi_k | \, \mathrm{QFT}^\dagger U(\theta) \, | \psi_k \rangle \|^2$. Figure 6 shows 45000 pairwise distance and the difference in the circuits' performance. It can be seen that circuits with a small distance are likely to have the same learning capacity in terms of the optimal fidelity. However, it is inconclusive when the distance is large, since two good circuits, even with substantial architectural difference, could result in the same optimal fidelity.

E. Implementation Details

E.1. Optimizing the acquisition function

We optimize the acquisition function (5) with an evolution approach. Beginning with a pool of $k(t)$ circuits, we generate $N_{\text{mut}}(t)$ samples by letting each initial circuit to reproduce $N_{\text{off}}(t)$ offsprings through mutation. We evaluate the acquisition function on $N_{\text{mut}}(t)$ samples. Those with high values are kept to the next generation. To be exact, half of the next generation are ones with highest values; the other half contains some of the remaining circuits such that those with higher values are more likely to be selected. Repeat the process through τ generations. We set $\tau = \Theta(\sqrt{t})$ and $k(t), N_{\text{off}}(t) = \Theta(\sqrt{\tau})$ by

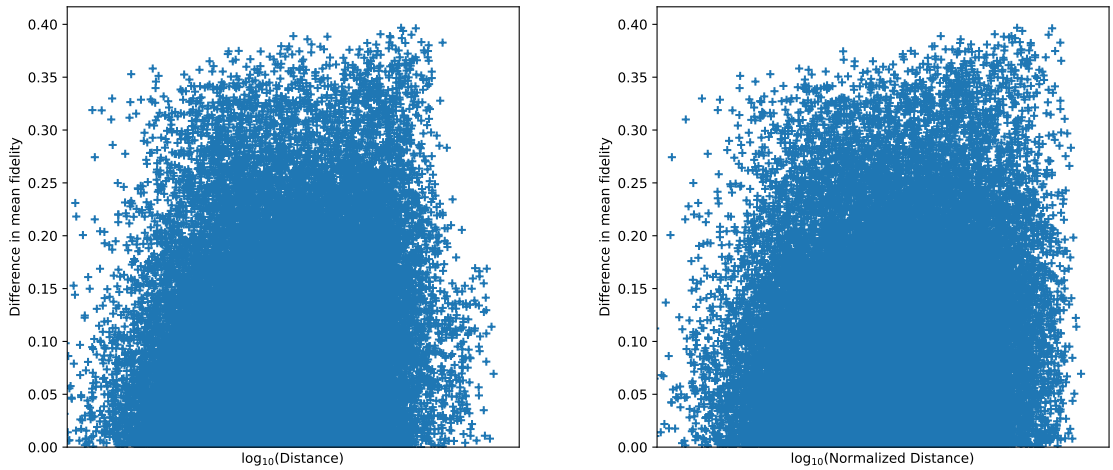


Figure 6: Each point in the scatter plot represents the distance and the performance after training of a pair of circuits.

default. For each mutation, a circuit can change at most 4 components. The number of changes from 1 to 4 is chosen from the distribution $[0.4, 0.3, 0.2, 0.1]$. A change can be either replacing the gate at a random position or switching the qubit wires connected to a random gate.

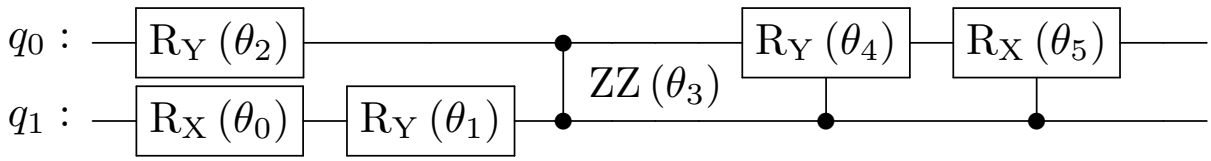
E.2. Training Quantum Neural Networks

In all of our experiments, the black-box function involves optimizing quantum neural networks. The optimization problem is straightforward for QFT and MaxCut problems since the optimal value obtained from the training problem is also the black-box function. In particular, as discussed in 4, the objective function is $\max_\theta \frac{1}{K}|\langle\psi_k| U_{\text{QFT}}^\dagger U(\theta) |\psi_k\rangle|^2$ for the QFT experiment and $\frac{1}{M}\sum_{m=1}^{M} \max_\theta \frac{\langle 0|U^\dagger(\theta)H_m U(\theta)|0\rangle}{C_m}$ for the MaxCut experiment. Training quantum circuits for these tasks with the L-BFGS-B optimizer is fairly robust.

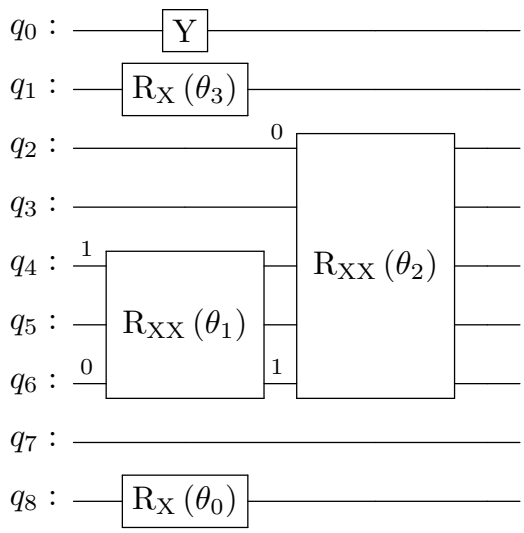
On the other hand, although we use the relative entropy D_{KL} as the black-box function for BO, optimizing QGAN requires a different approach. Given m training data x^l and m generated data g^l, the quantum generator G_θ and the classical discriminator D_ϕ are updated by $\min L_G$ and $\max L_D$ (Zoufal et al., 2019), where

$$\begin{aligned} L_G(\phi,\theta) =& \frac{1}{m}\sum_{l=1}^{m} \log D_\phi(g^l), \\ L_D(\phi,\theta) =& \frac{1}{m}\sum_{l=1}^{m} [\log D_\phi(x^l) + \log\big(1 - D_\phi(g^l)\big)]. \end{aligned} \tag{45}$$

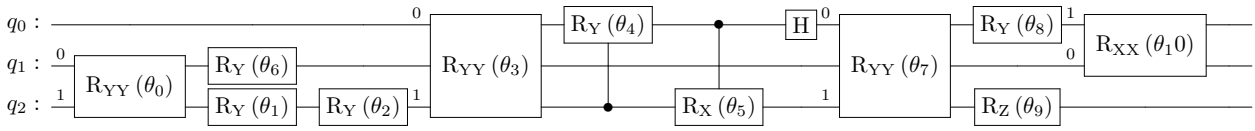
We train the QGAN objective using ADAM optimizer through 200 epochs with batch size 100 and learning rate 10^{-3}. Parameters are initialized uniformly at random in the interval $(-0.1, 0.1)$.



(a) QFT



(b) MaxCut



(c) QGAN

Figure 7: Optimal circuits by BO for three learning tasks