MATCHING POINT SETS WITH QUANTUM CIRCUIT LEARNING

*Mohammadreza Noormandipour** *Hanchen Wang*

University of Cambridge

ABSTRACT

In this work, we propose a parameterised quantum circuit learning approach to point set matching problem. In contrast to previous annealing-based methods, we propose a quantum circuit-based framework whose parameters are optimised via descending the gradients w.r.t a kernel-based loss function. We formulate the shape matching problem into a distribution learning task; that is, to learn the distribution of the optimal transformation parameters. We show that this framework is able to find multiple optimal solutions for symmetric shapes and is more accurate, scalable and robust than the previous annealing-based method.

***Index Terms*—** Quantum Circuits, Quantum Machine Learning

1. INTRODUCTION

Recent advances in Noisy Intermediate Scale Quantum (NISQ) [1] technologies have validated the potentials to achieve a quantum advantage (i.e., supremacy [2]) with tens of (noisy) qubits. As a subset of such, quantum machine learning (QML) [3–5] explores the venues to deploy machine learning algorithms on quantum systems [6–8].

The structure of QML algorithms is found to be fundamentally homogeneous with that of classical kernels [9]. Furthermore, kernel-based quantum models are supposed to benefit from a more efficient training than the variational variants [9] and therefore need fewer training parameters. This is of importance in hybrid quantum-classical models (i.e., quantum models with classical optimisation routines) on NISQ devices, since the circuits should be kept as shallow as possible due to existence of decoherence. On this basis, we are particularly interested in advancing QML methods with kernels.

In this paper, we propose a parameterised quantum circuit model (PQC) with a classical and a quantum kernel-based loss to match the point sets. The solution is inferred from the output distribution of the PQC and the training is performed via classical gradient descent. In comparison with the prior method [10], our method, qKC (Quantum Kernel Correlation), is: 1) a gate-based quantum computation model rather than an annealing-based model; 2) empirically scaled to 3D shapes; 3) differentiable trainable; 4) with smaller generalisation error; and 5) capable of finding multiple solutions for the symmetric shapes.

2. PROBLEM STATEMENT

In this section, we lay out the basic problem setting and notation. We denote the two point sets as $\mathcal{M}$ (model) and $\mathcal{S}$ (scene), where $\mathcal{M} = \{\mathbf{m}_i\}_{i=1}^{N}$ and $\mathcal{S} = \{\mathbf{s}_i\}_{i=1}^{N'}$. $\mathcal{M}$ is assumed to be transformed from $\mathcal{S}$ via a rigid transformation: $\mathcal{T} = [\mathbf{r}_{\mathcal{MS}}, \mathbf{t}_{\mathcal{MS}}]$, where $\mathbf{r}_{\mathcal{MS}} \in$ SO(2) or SO(3) and $\mathbf{t}_{\mathcal{MS}} \in \mathbb{R}^2$ or $\mathbb{R}^3$. The matching task objective is to minimise the summation of mean-squared error $\mathcal{L}$, given a ground true point pair correspondence between $\mathcal{M}$ and $\mathcal{S}$.

* Correspondence: mrn31@cam.ac.uk. Authors contributed equally.

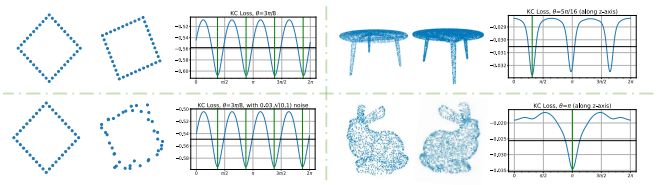

Fig. 1: Diagram of the kernel correlation (KD) loss. The left column shows that KC loss is capable of finding *all* the four optimal matching angles for a symmetric 2D square even in the presence of noise (bottom-left). The right column shows the effectiveness of KC loss for more complicated 3D shapes, namely a symmetric table (top-right) and an asymmetric bunny (bottom-right).

A point pair correspondence function $L_{\mathcal{MS}}$ is then defined to fetch the corresponding point(s) in $\mathcal{S}$ for each queried point $\mathbf{m}_i$ in $\mathcal{M}$: $L_{\mathcal{MS}} : \mathcal{M} \rightarrow \mathcal{S}$. Usually, $L_{\mathcal{MS}}$ is assumed to be bijective ($N = N'$): $L_{\mathcal{MS}}(\mathbf{m}_i) = \mathbf{s}_j$, which is mostly used in prior works. An alternative design is to return multiple corresponding points in $\mathcal{S}$ for each query point $\mathbf{m}_i$ in $\mathcal{M}$: $L_{\mathcal{MS}}(\mathbf{m}_i) = \{\mathbf{s}_{j_k}\}_{k=1}^{K}$. This multi-linked design is utilised in works including EM-ICP [11], Soft-Assignment [12], as well as ours. With $L_{\mathcal{MS}}$ defined, we can directly solve the optimal transformation $\mathcal{T}^{opt}$ in a deterministic fashion:

$$\left(\frac{\partial}{\partial \mathcal{T}} \sum_{\mathbf{m}_i \in \mathcal{M}} ||\mathcal{T}\mathbf{m}_i - L_{\mathcal{MS}}(\mathbf{m}_i)||_2 \right) \Bigg|_{\mathcal{T}=\mathcal{T}^{opt}} = 0 \quad (1)$$

where $\mathcal{T}\mathbf{m}_i := \mathbf{r}_{\mathcal{MS}} \cdot \mathbf{m}_i + \mathbf{t}_{\mathcal{MS}}$. The ground truth of the transformation $\mathcal{T}^{gt}$ is also computed in this fashion based on the provided rather than the predictive correspondence. Clearly it is with no challenge if the predictive correspondence is the same as the ground truth. However, finding such correspondence mapping for a perfect matching is known as a NP-hard problem. Therefore, we tackle the matching problem under the context of fully-connection where the all-to-all correspondence is given by kernel correlations [13].

3. RELEVANT WORKS

Besides recent progress such as self supervised learning [14] on 3D point cloud, quantum computation [3, 15] provides a novel perspective for efficient processing and optimisation. While the previous attempts [10, 16–18] are based on the annealing-based quantum system (adiabatic quantum computer, AQC), we instead consider a gate-based quantum circuit learning framework.

To the best of our knowledge, we are the first to apply QCL to 3D vision. The most relevant work is QA proposed by (Golyanik & Theobalt [10]). QA formulates the matching objective as a quadratic unconstrained binary optimisation problem (QUBOP), where the measured qubit states are expected to provide solution for the predefined quadratic energy. However, their method is not empirically applicable to 3D data and cannot find multiple optimal rotations when

the shapes are symmetric. In contrast, we formulate this task as a distribution learning problem realised by a quantum circuit system. On the same downstream task, our method is more robust, and with less transformation discrepancy and alignment error. We have also provide a novel technique for quantum kernel encoding, which can cope with classically-hard-to-learn feature maps in RKHS.

4. KERNEL CORRELATION

In this section, we introduce how prior methods [13, 19] match two point sets with a kernel correlation (KC) loss. We provide a diagram of the KC loss for 2D/3D, (a)symmetric shapes in Figure 1.

Kernel Correlation (KC) [13] extends the correlation techniques to point set matching. It is a measure of affinity as well as a function of the entropy. KC between two points, $\mathbf{x}_i$ and $\mathbf{x}_j$, is defined as:

$$\mathcal{KC}(\mathbf{x}_i, \mathbf{x}_j) = \int_{\mathbb{R}^k} \kappa(\mathbf{x}, \mathbf{x}_i)\kappa(\mathbf{x}, \mathbf{x}_j)\mathrm{d}\mathbf{x} \tag{2}$$

where $\kappa(\cdot, \cdot)$ is a kernel such as Gaussian and quantum. The homologic Gaussian KC is written as follows, where α and σ are constants:

$$\mathcal{KC}_G(\mathbf{x}_i, \mathbf{x}_j) = \alpha \exp(-||\mathbf{x}_i - \mathbf{x}_j||_2^2/\sigma^2) \tag{3}$$

The KC between two point sets, $\mathcal{M}$ and $\mathcal{S}$, is defined as:

$$\begin{aligned}\mathcal{KC}(\mathcal{M}, \mathcal{S}) &= \sum_{\mathbf{s}\in\mathcal{S}} \sum_{\mathbf{m}\in\mathcal{M}} \mathcal{KC}(\mathcal{T}\mathbf{m}, \mathbf{s}) \\ &= \mathbb{E}_{\mathbf{s}\sim\mathcal{S}\,\mathbf{m}\sim\mathcal{M}}[\mathcal{KC}(\mathbf{m}, \mathbf{s})] * \|\mathcal{M}\| * \|\mathcal{S}\|\end{aligned} \tag{4}$$

where $\|\cdot\|$ is the set cardinality. If the two point sets are closely aligned, the KC is large. Therefore, the optimal transformation $\mathcal{T}^{opt}$ is solved via finding the *minima* of the negative KC value:

$$\mathcal{T}^{opt} = \arg\min \mathcal{L}_{\mathcal{KC}}, \qquad \text{where } \mathcal{L}_{\mathcal{KC}} = -\mathcal{KC}(\mathcal{T} \circ \mathcal{M},\ \mathcal{S}) \tag{5}$$

Notice that in Equation 5 each transformed model point $\mathbf{m}$ is interacting with all the scene points. We call it a fully-linked registration cost function. This is in contrast to the methods like ICP [20, 21] and prior quantum method [10], where each model point is connected to a subset of scene points. It is clear that the objective defined in Equation 5 satisfies the minimum requirement for a registration algorithm. That is, $\mathcal{T}^{opt}$ corresponds to one of the global minima of the cost. Same as previous work [10], we assume the translation between two point sets is resolved by aligning the centres of mass, in the following sections, we focus on solving the optimal rotations with QCL.

5. QUANTUM CIRCUIT LEARNING

In this section, we describe how we tackle the problem in the context of Quantum Circuit Learning (QCL) [15, 22]. In Section 5.1, we summarise how previous work utilises Born Machine circuit, a subset of PQCs for distribution learning. In Section 5.2, we define a new loss and its gradient, bridging the distribution learning and the minimisation of kernel correlation loss. In appendix, we provide theorems and details for implementing a quantum kernel, which can offer potential advantages. It is worth noting that our method is *fundamentally* different from the prior attempt [10] and its follow-ups [16–18] regarding the model/approach of quantum computation at work.

In gate-based quantum computation models, quantum circuits are composed of qubits and logic (parameterised) quantum gates. From computational point of view, a qubit is a mathematical object composed of a linear combination of two basis states that the information is encoded in: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where $\alpha, \beta \in \mathbb{C}$, $\|\alpha\| + \|\beta\| = 1$ and $|0\rangle$ and $|1\rangle$ are the basis states in Dirac notation. The quantum

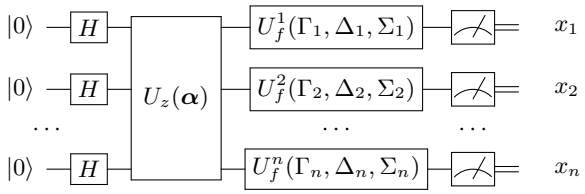

Fig. 2: Schematic of a PQC, from [24]

gates are unitary (norm-preserving) matrices that act on a normalised initial quantum state and cause a unitary evolution of it. The choice of the unitary gates is done in such a way that the final evolved quantum state provides the solution of a specific problem upon measurements. For a more comprehensive introduction, we recommend [15, 23].

5.1. Quantum Circuit and Ising Born Machine

A generic PQC (in Figure 2) consists of three stages: state preparation, evolution and measurement. During preparation, n qubits are initialised in $|0\rangle$ states and then are acted upon by Hadamard gates to produce a super-position of possible states with equal probabilities:

$$\boldsymbol{H}^{\otimes n}|0\rangle^{\otimes n} = |+\rangle^{\otimes n} \tag{6}$$

It is then followed by a unitary evolution by m operators, each acting on the qubits in the set S_j:

$$\boldsymbol{U}_z(\boldsymbol{\alpha}) \coloneqq \prod_{j=1}^{m} U_z(\alpha_j, S_j) \coloneqq \prod_{j=1}^{m} \exp\left(\mathrm{i}\alpha_j \bigotimes_{k\in S_j} Z_k\right) \tag{7}$$

The observation is made via the measurement unitary $U_f(\cdot, \cdot, \cdot)$, a set of single qubit gates:

$$U_f\left(\boldsymbol{\Gamma}, \boldsymbol{\Delta}, \boldsymbol{\Sigma}\right) \coloneqq \exp\left(\mathrm{i}\sum_{k=1}^{n} \Gamma_k X_k + \Delta_k Y_k + \Sigma_k Z_k\right) \tag{8}$$

where X_k, Y_k, Z_k are the canonical Pauli operators [23] acting on the k-th qubit, $\Gamma_k, \Delta_k, \Sigma_k$ are parameters that can be specifically chosen to realise different circuit families incl. instantaneous quantum polynomial time (IQP) [25, 26] and quantum approximate optimisation algorithm (QAOA) [27, 28]. Essentially here learning with quantum circuits is to optimise the parameters $\boldsymbol{\alpha}$ in the unitary evolution given a particular task and a specific circuit design (e.g., interconnect).

A Born Machine [29, 30] is a quantum-classical hybrid generative model for distribution learning. Concretely, it utilises a classical optimisation tactic (i.e., gradient descent) to alter the parameters $\boldsymbol{\alpha}$ of a quantum system, thus the output binary vector $\boldsymbol{x}$ of quantum system (circuit) is able to produce samples according to the target distribution. According to the Born's Rule [1], the output probability distribution of a quantum circuit (parameterised by $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \boldsymbol{\Gamma}, \boldsymbol{\Delta}, \boldsymbol{\Sigma}\}$) is:

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \left|\langle\mathbf{x}|\, \boldsymbol{U}_f\left(\boldsymbol{\Gamma}, \boldsymbol{\Delta}, \boldsymbol{\Sigma}\right) \boldsymbol{U}_z(\boldsymbol{\alpha}) \boldsymbol{H}^{\otimes n}|0\rangle^{\otimes n}\right|^2 \tag{9}$$

where the raw output of the n-qubit circuit, $\mathbf{x} = [x_1, x_2, .., x_n] \in \{0, 1\}^n$, is a n-dimensional binary vector. Usually, the Born Machine is restricted into an *Ising* version [24] in order to be implementable on the NISQ devices. That is, all the unitary gates act on either one or two qubits, i.e., $|S_j| \leq 2$.

On this basis, we can rewrite the unitary evolution (7) as an Ising Hamiltonian [31, 32], $\boldsymbol{U}_z(\boldsymbol{\alpha}) = \exp\left(\mathrm{i}\mathcal{H}_z\right)$, and $\boldsymbol{U}_z$ is:

$$\mathcal{H}_z = \sum_{i<j} J_{ij} Z_i Z_j + \sum_{k=1}^{n} b_k Z_k \tag{10}$$

where $Z_{i,j,k}$ are the z-basis Pauli operators acting on the $\{i,j,k\}$-th qubit, $\boldsymbol{\alpha} = \{J_{ij}, b_k\}$ are the parameters we need to optimise w.r.t. input distribution[1]. $\{J_{ij}, b_k\}$ can be viewed as the pairwise coupling and local magnetic fields, respectively. For a n-qubit Ising Born Machine model, the number of parameters are: $n*(n-1)/2+n = n*(n+1)/2$. Note that the prior work [10] does not have *any* parameters to be optimised, their quadratic objective is fixed and universal for all point sets.

As mentioned, the output of a n-qubit Ising Born Machine (IBM) is a n-dimensional binary vector which has a total of 2^n possible values. We now define the mapping from these binary output to the solution of the point matching problem. We use the same setting as previous attempt [10], assuming the translation in the $\mathcal{T}^{\text{opt}}$ is resolved by aligning the mass centres of the two point sets, and only considering rotation along *one* axis. Therefore we simply require to transform the distribution of the output binary vectors to a *single* rotation angle. Specifically, we equally split the range $[0, 2\pi)$ of the rotation angle into 2^n number of bins, where each binary vector corresponds to a unique bin.

For instance, if the measurement outcome of a 4-qubit IBM is $\mathbf{x} = (0,1,0,1)$, the corresponding rotation angle is the median value of the 6-th bin, which is $\frac{(1\times2^0+1\times2^2)*2+1}{2^4*2}\times 2\pi = \frac{11}{16}\pi$. The resolution for the angle distribution is $2\pi/2^n$, which will grow exponentially as the number of qubits increases. However, due to the hardness of using classical simulation on quantum systems [32, 33], in experiment we mainly consider 4/6-qubits shallow circuits.

5.2. Training Procedures

We first describe how we design a kernel-based loss, to bridge the distribution learning and optimal alignment. Then we present the gradient calculation procedure to enable differentiable training.

To utilise Born Machine models for distribution learning, a differentiable loss function is usually required which measures the closeness between the model output distribution $p_{\boldsymbol{\theta}}(\mathbf{x})$ and the ground true distribution $\pi(\mathbf{y})$. The first metric to be considered is KL Divergence:

$$\mathcal{L}_{\text{KL}} = -\sum_{\mathbf{x}} \pi(\mathbf{x}) \log\left(p_{\boldsymbol{\theta}}(\mathbf{x})\right) = -\,\mathbb{E}_{\mathbf{x}\sim\pi} \log\left(p_{\boldsymbol{\theta}}(\mathbf{x})\right) \tag{11}$$

However, as noted in [24, 30, 33], calculating the gradient of $\mathcal{L}_{\text{KL}}$ is #P-hard, thus make it a suboptimal choice. So we instead use an alternative discrepancy measure called Integral Probability Metrics (IPMs) [34], among which a well-known example is Maximum Mean Discrepancy (MMD) [35, 36][2]. It is widely used in applications incl. homogeneity testing [37] and independence testing [38, 39].

A general form of an IPM $\gamma_{\mathcal{F}}(\mathcal{P}, \mathcal{Q})$ is defined as:

$$\sup_{\phi\in\mathcal{F}} \left| \int_{\mathcal{M}} \phi \,\mathrm{d}\,\mathcal{P} - \int_{\mathcal{M}} \phi \,\mathrm{d}\,\mathcal{Q} \right| = \sup_{\phi\in\mathcal{F}} \left(\mathbb{E}_{\mathcal{P}}[\phi] - \mathbb{E}_{\mathcal{Q}}[\phi]\right) \tag{12}$$

where $\mathcal{P}$ and $\mathcal{Q}$ are defined on the same probability space $\mathcal{M}$. $\mathcal{F}$ is a set of real bounded measurable functions on $\mathcal{M}$. $\phi(\cdot)$ defines a feature map from the input space $\mathcal{X}$ to the feature space $\mathcal{H}$ (i.e. Reproducing Kernel Hilbert Space, RKHS [40]) with a corresponding kernel function $\kappa(\cdot,\cdot)$:

$$\phi : \mathcal{X} \to \mathcal{H}, \qquad \kappa(\mathbf{x},\mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle_{\mathcal{H}}, \quad \forall \mathbf{x},\mathbf{y} \in \mathcal{X} \tag{13}$$

[1] some works [24] treat $(\boldsymbol{\Gamma}, \boldsymbol{\Delta}, \boldsymbol{\Sigma})$ as learnable parameters as well.

[2] examples of IPMs incl. Wasserstein Distance, Total Variation, Stein Discrepancy and Sinkhorn Divergence.

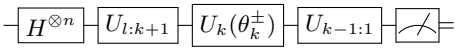

Fig. 3: a Shifted PQC, from [24, 30]

$\gamma_{\mathcal{F}}(\cdot,\cdot)$ becomes MMD when $\mathcal{F}$ is confined in a unit hyper-sphere: $\mathcal{F} = \{\phi : \|\phi\|_{\mathcal{H}} \le 1\}$. It has been proven that MMD equates the mean embedding differences between $\mathcal{P}$ and $\mathcal{Q}$ [31]:

$$\gamma_{\text{MMD}}(\mathcal{P},\mathcal{Q}) = \|\mu_{\mathcal{P}} - \mu_{\mathcal{Q}}\|_{\mathcal{H}} = \|\mathbb{E}_{\mathbf{x}\sim\mathcal{P}}[\phi(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\sim\mathcal{Q}}[\phi(\mathbf{x})]\|_{\mathcal{H}} \tag{14}$$

Born Machine [24, 29, 30] thus defines the MMD training loss:

$$\begin{aligned}\mathcal{L}_{\text{MMD}} = \gamma^2_{\text{MMD}} =& \mathbb{E}_{\mathbf{x}\sim\mathcal{P},\,\mathbf{y}\sim\mathcal{P}}[\kappa(\mathbf{x},\mathbf{y})] + \mathbb{E}_{\mathbf{x}\sim\mathcal{Q},\,\mathbf{y}\sim\mathcal{Q}}[\kappa(\mathbf{x},\mathbf{y})] \\ &- 2\,\mathbb{E}_{\mathbf{x}\sim\mathcal{P},\,\mathbf{y}\sim\mathcal{Q}}[\kappa(\mathbf{x},\mathbf{y})]\end{aligned} \tag{15}$$

We can re-write the $\mathcal{L}_{\text{MMD}}$ using Kernel Correlation in Equation 4:

$$\begin{aligned}\mathcal{L}_{\text{MMD}} * N^2 =& \mathbb{E}_{\mathbf{x}\sim\mathcal{P},\,\mathbf{y}\sim\mathcal{P}}[\mathcal{KC}(\mathbf{x},\mathbf{y})] + \mathbb{E}_{\mathbf{x}\sim\mathcal{Q},\,\mathbf{y}\sim\mathcal{Q}}[\mathcal{KC}(\mathbf{x},\mathbf{y})] \\ &- 2\,\mathbb{E}_{\mathbf{x}\sim\mathcal{P},\,\mathbf{y}\sim\mathcal{Q}}[\mathcal{KC}(\mathbf{x},\mathbf{y})]\end{aligned} \tag{16}$$

here we assume $\|\mathcal{P}\| = \|\mathcal{Q}\| = N$, i.e., the two sets have the same N points. Note that the kernel $\kappa(\cdot,\cdot)$ should be *bounded, measurable* and *characteristic* [34], which are satisfied by Gaussian kernels.

If $\mathbf{x}$ and $\mathbf{y}$ are sampled from the same distribution (i.e., a surface of rigid object), the $\mathcal{KC}$ term essentially represents the compactness of that shape, which is constant. Therefore, Equation 16 tells us minimising the $\mathcal{L}_{\text{MMD}}$ substantially equates the diminution of the KC loss (Equation 5).

Recall that the raw output from the quantum circuit serves as the distribution of the rotation angles $p_\theta(\cdot)$, $\mathcal{P}$ and $\mathcal{Q}$ are identical. Thus we can re-write Equation 16 in terms of p_θ:

$$\mathcal{L}_{\text{MMD}} * N^2 = -2\,\mathbb{E}_{\mathbf{m}\sim\mathcal{M},\,\mathbf{s}\sim\mathcal{S},\,\mathbf{x}\sim p_\theta}[\mathcal{KC}(\mathcal{T}_{\mathbf{x}}\mathbf{m},\mathbf{s})] + \text{constant} \tag{17}$$

In Equation 17, we have shown that using the same distribution learning framework under (Ising) Born Machine models, we can minimise the KC loss [13] to align two point sets. We name our method **qKC**. Furthermore, since the IBM solver outputs a distribution rather than a scalar/vector, qKC is able to learn *multiple* optimal angles particularly for symmetric shapes (shown in Figure 4).

The gradients of the Born Machine [24, 30] can be written as[3]:

$$\frac{\partial p_{\boldsymbol{\theta}}(\mathbf{x})}{\partial \boldsymbol{\theta}_k} = p^{-}_{\boldsymbol{\theta}_k}(\mathbf{x}) - p^{+}_{\boldsymbol{\theta}_k}(\mathbf{x}) \tag{18}$$

where $p^{\pm}_{\boldsymbol{\theta}_k}$ is the measured output distribution with a shift on the k-th circuit parameter $p_{\boldsymbol{\theta}_k}$ [41]:

$$p^{\pm}_{\boldsymbol{\theta}_k} = p_{\boldsymbol{\theta}_k \pm \pi/2} \tag{19}$$

Measuring a parameter shifted circuit (Figure 3), the derivative is:

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{MMD}}}{\partial \boldsymbol{\theta}_k} =& 2\,\mathbb{E}_{\mathbf{m}\sim\mathcal{M},\,\mathbf{s}\sim\mathcal{S},\,\mathbf{x}^+\sim p^+_{\boldsymbol{\theta}_k}}\left[\mathcal{KC}\left(\mathcal{T}_{\mathbf{x}^+}\mathbf{m},\mathbf{s}\right)\right] \\ &- 2\,\mathbb{E}_{\mathbf{m}\sim\mathcal{M},\,\mathbf{s}\sim\mathcal{S},\,\mathbf{x}^-\sim p^-_{\boldsymbol{\theta}_k}}\left[\mathcal{KC}\left(\mathcal{T}_{\mathbf{x}^-}\mathbf{m},\mathbf{s}\right)\right]\end{aligned} \tag{20}$$

[3] A more precise notation is to use $\mathbf{z}$ instead of $\mathbf{x}$ [22], since the measurement is on the $\mathbf{z}$-computational basis (i.e., electron spin around the $\hat{\mathbf{z}}$-direction): $|\mathbf{z}\rangle\langle\mathbf{z}|$ (right after $U^k_f(\cdot,\cdot,\cdot)$ in Figure 2). For simplicity, here we stick to $\mathbf{x}$ to represent the measured bit string from the quantum circuit.

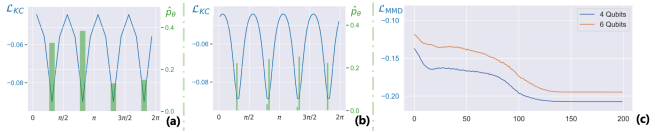

Fig. 4: Matching a 2D (symmetric) square in the xy plane with qKC, with a rotation of $5\pi/16$ around the $\hat{z}$-axis. (a) and (b): KC loss (in blue line) and output rotation distribution (in green bar) from the 4-qubit and 6-qubit qKC, respectively, (c): training curves of qKC.

The empirical estimation of the loss and gradient are:

$$\tilde{\mathcal{L}}_{\text{MMD}} = \tilde{\gamma}^2_{\text{MMD}} = \frac{-2}{N^2\|\mathcal{X}\|} \sum_{\mathbf{s}\in\mathcal{S}} \sum_{\mathbf{m}\in\mathcal{M}} \sum_{\mathbf{x}\in\mathcal{X}} \mathcal{KC}\left(\mathcal{T}_{\mathbf{x}}\mathbf{m}, \mathbf{s}\right)$$

$$\frac{\partial\tilde{\mathcal{L}}_{\text{MMD}}}{\partial\boldsymbol{\theta}_k} = \frac{2}{N^2\|\mathcal{X}\|} \sum_{\substack{\mathbf{s}\in\mathcal{S}\\ \mathbf{m}\in\mathcal{M}}} \Big(\sum_{\mathbf{x}\in\mathcal{X}^+} \mathcal{KC}(\mathcal{T}_{\mathbf{x}^+}\mathbf{m}, \mathbf{s}) - \sum_{\mathbf{x}\in\mathcal{X}^-} \mathcal{KC}(\mathcal{T}_{\mathbf{x}^-}\mathbf{m}, \mathbf{s})\Big)$$

where $\mathcal{X}$ is the number of samples from the circuit at each iteration, $\|\cdot\|$ is the cardinality, and $\|\mathcal{X}\| = \|\mathcal{X}^+\| = \|\mathcal{X}^-\|$. Intuitively the larger the sample size is, the better it can represent $p_{\boldsymbol{\theta}}(\mathbf{x})$ and $p_{\boldsymbol{\theta}^\pm}(\mathbf{x})$, as well as the loss and gradient. [34] shows this empirical estimation on the MMD loss and gradient of qKC is unbiased and with a quadratic convergence rate:

$$\begin{aligned} &|\tilde{\gamma}_{\text{MMD}}\left(\mathcal{P}_M, \mathcal{Q}_N\right) - \gamma_{\text{MMD}}(\mathcal{P}, \mathcal{Q})| \\ &= O_{\mathcal{P},\mathcal{Q}}\left(\frac{1}{\sqrt{M}} + \frac{1}{\sqrt{N}}\right) \xrightarrow{a.s.} 0 \;\text{ as }\; M, N \to \infty \end{aligned} \tag{21}$$

where M is number of samples from distribution $\mathcal{P}$: $M = \|\mathcal{P}_M\|$.

However, in qKC both $\mathcal{P}$ and $\mathcal{Q}$ (or $\mathcal{M}$ and $\mathcal{S}$) represent the same underlying shape. Empirically we don't observe that sampling more points results in a faster convergence and better solution. We therefore investigate how $\|\mathcal{X}\|$ (i.e., batch size) will affect the training. In experiments we find that $\|\mathcal{X}\|$ is not as critical as other parameters such as learning rate for the optimisation process. We will have more discussions on this parameter tuning in next section and appendix.

Besides Equation 18, there is an alternative strategy [22] for the gradient estimation of Born Machine models. It requires additional measurements on the imaginary parts of the unitary via additional auxiliary qubits and Hadamard gates. Our framework by contrast is more implementation friendly. We have also explored solutions with quantum kernels which we have included in the arXiv appendix.

6. EXPERIMENTS

In this section, we first describe the architectures and measurement procedures of qKC. We then present the set-up details and compare the experimental results with the prior attempt [10].

Circuit architecture We use QAOA circuit [42], it is an algorithm to approximately prepare a desired quantum state via a p-depth quantum circuit. We use the same design of IBM [24] (described in Section 5.1), to implement a shallowest version of a QAOA circuit, we provide detail description in the appendix.

Training setting Same as previous works [10, 22, 24], our experiments are limited by the fact that only small (in terms of number of operating qubits) quantum devices are less costly to simulate and experimentally available. So we only use 4-qubit and 6-qubit QAOA architectures. Further details can be found in the appendix.

Evaluation metric We use the same metrics as the prior [10] to quantify the preciseness of matching results, detailed in appendix.

2D symmetric shape We first evaluate qKC on the synthesised 2D symmetric polygons. Each polygon is centred at the origin and inscribed by a unit circle. We use the canonical and rotated shapes for the scene $\mathcal{S}$ and model $\mathcal{M}$ sets, respectively. Each side of these polygons consists of 10 points. In Figure 4, we plot the KC loss, output distribution and training curve for matching a pair of squares, with a rotation of $5\pi/16$ (the other optima are $13\pi/16, 21\pi/16, 29\pi/16$).

We show that Gaussian kernel loss can capture all the optimal rotations. For the 4-qubit qKC, all the output samples (1000 predictions) fall in the optimal solutions. As the resolution (i.e., number of qubits) increases, over 85% of samples belong to the optima and 99% are covered by their neighbourhoods. Compare with previous attempts [24, 30] on learning with Quantum Born Machine, our loss curve is with more stability while decreasing. We observe that the quantum kernel provides a faster convergence, as shown in appendix.

	Golyanik & Theobalt, QA [10]					Ours, qKC	
	K					κ_G	
	10	20	30	40	50	$n_Q = 4$	$n_Q = 6$
e_{2D}	0.026	0.041	0.078	0.17	0.3	**0**	**0.0097**
σ_{2D}	0.013	0.012	0.012	0.012	0.013	**0**	**0.0199**
e_R	0.062	0.083	0.22	0.47	0.764	**0**	**0**
σ_R	0.044	0.041	0.036	0.031	0.03	**0**	**0**

Table 1: Performance on the fish dataset (2D). We compare 4-qubit and 6-qubit (n_Q=4 or 6) qKC of Gaussian kernel ($\kappa_G, \sigma^2 = 0.01$) and quantum kernel ($\kappa_Q$), with all the configurations (characterised by the number of corresponding points, K) of QA.

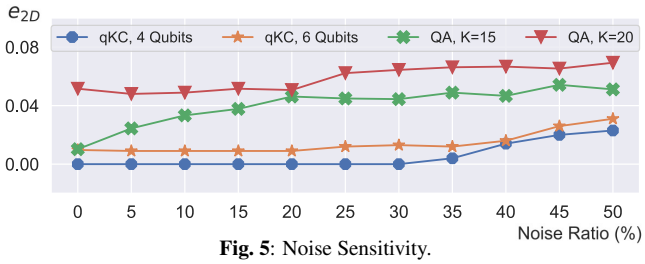

Fig. 5: Noise Sensitivity.

Fish We compare qKC with QA [10] on matching the 2D shapes from the fish dataset in Table 1. Though the two methods are based on fundamentally different architectures, we stick to the same experimental settings as [10]. We use the rotation with the highest output probability for benchmarking. We have demonstrated that qKC can solve the matching problem with more preciseness.

Noise sensitivity We use the same procedures as QA [10] to evaluate the robustness of qKC when the data (from fish dataset) is accompanied with noise. The ratio of iid Gaussian noise ranges from 5% to 50%. We report the averaged e_{2D} for every noise ratio over 50 runs in Figure 5. We observe both methods have increasing misalignment when the noise level rises, and qKC has smaller generalisation error.

7. DISCUSSION

In this work, for the first time, a generic QCL framework is applied to the point set matching problem, whose solution is represented as a distribution of rotations. We have shown our proposed framework named qKC has several benefits in comparison with the prior work. We believe qKC is potentially capable of showing quantum computational advantage upon efficient implementation on quantum hardware in the near future. We hope that our work will bring insights for future research in quantum machine learning and related optimisation problems in the 3d vision community.

8. ACKNOWLEDGEMENT

We would like to thank Weiyang Liu, Shuai Yuan, Clarice D. Aiello and Joan Lasenby for valuable discussions. HW is partially supported by the Cambridge Trust Scholarship, the Cathy Xu Fellowship, the CAPA Research Grant and the Cambridge Philosophical Society.

References

[1] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum*, vol. 2, pp. 79, 2018.

[2] T. F. Rønnow, Z. Wang, J. Job, et al., "Defining and detecting quantum speedup," *science*, vol. 345, no. 6195, pp. 420–424, 2014.

[3] J. Biamonte, P. Wittek, N. Pancotti, et al., "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, 2017.

[4] M. Schuld, I. Sinayskiy, and F. Petruccione, "An introduction to quantum machine learning," *Contemporary Physics*, 2015.

[5] P. Wittek, *Quantum Machine Learning: What Quantum Computing Means to Data Mining*, Academic Press, 2014.

[6] F. Arute, K. Arya, R. Babbush, et al., "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.

[7] H.-S. Zhong, H. Wang, Y.-H. Deng, et al., "Quantum computational advantage using photons," *Science*, 2020.

[8] M. Noormandipour, "Implementation of a quantum perceptron in intel-qs," 2020.

[9] M. Schuld, "Quantum machine learning models are kernel methods," *arXiv preprint arXiv:2101.11020*, 2021.

[10] V. Golyanik and C. Theobalt, "A quantum computational approach to correspondence problems on point sets," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2020.

[11] S. Granger and X. Pennec, "Multi-scale em-icp: A fast and robust approach for surface registration," in *ECCV*, 2002.

[12] A. Rangarajan, H. Chui, and F. L. Bookstein, "The softassign procrustes matching algorithm," in *Biennial International Conference on Information Processing in Medical Imaging (IPMI)*, 1997.

[13] Y. Tsin and T. Kanade, "A correlation-based approach to robust point set registration," in *ECCV*, 2004.

[14] H. Wang, Q. Liu, X. Yue, J. Lasenby, and M. J. Kusner, "Unsupervised point cloud pre-training via view-point occlusion, completion," *arXiv preprint arXiv:2010.01089*, 2021.

[15] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum circuit learning," *Physical Review A*, vol. 98, 2018.

[16] M. S. Benkner, V. Golyanik, C. Theobalt, and M. Moeller, "Adiabatic quantum graph matching with permutation matrix constraints," in *IEEE International Conference on 3D Vision, 3DV*, 2020.

[17] M. S. Benkner, Z. Lähner, V. Golyanik, et al., "Q-match: Iterative shape matching via quantum annealing," *arXiv preprint arXiv:2105.02878*, 2021.

[18] T. Birdal, V. Golyanik, C. Theobalt, and L. Guibas, "Quantum permutation synchronization," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2021.

[19] Y. Shen, C. Feng, Y. Yang, and D. Tian, "Mining point cloud local structures by kernel correlation and graph pooling," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2018.

[20] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI*, vol. 14, no. 2, pp. 239–256, 1992.

[21] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *ICRA*, 1991.

[22] E. Farhi and H. Neven, "Classification with quantum neural networks on near term processors," *arXiv preprint arXiv:1802.06002*, 2018.

[23] M. A. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.

[24] B. Coyle, D. Mills, V. Danos, and E. Kashefi, "The born supremacy: quantum advantage and training of an ising born machine," *npj Quantum Information*, vol. 6, no. 1, 2020.

[25] D. Shepherd and M. J. Bremner, "Temporally unstructured quantum computation," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 465, no. 2105, pp. 1413–1439, 2009.

[26] M. J. Bremner, A. Montanaro, and D. J. Shepherd, "Achieving quantum supremacy with sparse and noisy commuting quantum computations," *Quantum*, vol. 1, pp. 8, 2017.

[27] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint arXiv:1411.4028*, 2014.

[28] E. Farhi and A. W. Harrow, "Quantum supremacy through the quantum approximate optimization algorithm," *arXiv preprint arXiv:1602.07674*, 2016.

[29] S. Cheng, J. Chen, and L. Wang, "Information perspective to probabilistic modelling: Boltzmann machines versus born machines," *Entropy*, vol. 20, no. 8, 2018.

[30] J.-G. Liu and L. Wang, "Differentiable learning of quantum circuit born machines," *Physical Review A*, vol. 98, 2018.

[31] M. J. Bremner, A. Montanaro, and D. J. Shepherd, "Average-case complexity versus approximate simulation of commuting quantum computations," *Physical Review Letters*, vol. 117, no. 8, 2016.

[32] K. Fujii and T. Morimae, "Commuting quantum circuits and complexity of ising partition functions," *New Journal of Physics*, vol. 19, no. 3, pp. 033003, 2017.

[33] M. J. Bremner, R. Jozsa, and D. J. Shepherd, "Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 467, no. 2126, pp. 459–472, 2011.

[34] B. K. Sriperumbudur, K. Fukumizu, A. Gretton, B. Schölkopf, and G. R. Lanckriet, "On integral probability metrics,\phi-divergences and binary classification," *arXiv preprint arXiv:0901.2698*, 2009.

[35] B. K. Sriperumbudur, K. Fukumizu, and G. R. Lanckriet, "Universality, characteristic kernels and rkhs embedding of measures.," *Journal of Machine Learning Research, JMLR*, vol. 12, no. 7, 2011.

[36] B. K. Sriperumbudur, A. Gretton, K. Fukumizu, B. Schölkopf, and G. R. Lanckriet, "Hilbert space embeddings and metrics on probability measures," *Journal of Machine Learning Research, JMLR*, vol. 11, pp. 1517–1561, 2010.

[37] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola, "A kernel method for the two-sample-problem," in *Advances in neural information processing systems, NeurIPS*, 2006.

[38] A. Gretton, K. Fukumizu, C. H. Teo, et al., "A kernel statistical test of independence.," in *Advances in neural information processing systems, NeurIPS*, 2007.

[39] K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf, "Kernel measures of conditional dependence.," in *Advances in neural information processing systems, NeurIPS*, 2007.

[40] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American mathematical society*, vol. 68, no. 3, pp. 337–404, 1950.

[41] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, "Evaluating analytic gradients on quantum hardware," *Physical Review A*, vol. 99, no. 3, pp. 032331, 2019.

[42] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint arXiv:1411.4028*, 2014.