SLIQ: Quantum Image Similarity Networks on Noisy Quantum Computers

Daniel Silver, Tirthak Patel, Aditya Ranjan, Harshitta Gandhi, William Cutler, Devesh Tiwari

Northeastern University
{silver.da, patel.ti, ranjan.ad, gandhi.ha, cutler.wi, d.tiwari}@northeastern.edu

Abstract

Exploration into quantum machine learning has grown tremendously in recent years due to the ability of quantum computers to speed up classical programs. However, these efforts have yet to solve unsupervised similarity detection tasks due to the challenge of porting them to run on quantum computers. To overcome this challenge, we propose SLIQ, the first open-sourced work for resource-efficient quantum similarity detection networks, built with practical and effective quantum learning and variance-reducing algorithms.

Introduction

Brief Overview and Motivation. Rapid advancements in quantum machine learning (QML) have increasingly allowed researchers to leverage the benefits of quantum computing in solving ML problems. Although the degree of advantage for different ML tasks is being explored, the recent advances suggest that classification and solving high energy physics problems are among the most promising candidates (Huang et al. 2021b; Guan et al. 2021). In particular, recent efforts have focused on developing high-quality classification circuits for quantum computers. While the resulting classifiers have been highly effective, they are restricted because classification, like all other supervised learning methods, requires labeled data. In many real-world scenarios, labeled data is either not readily available (e.g., diagnosing complex diseases based on medical images without quantifiable ground truth (Utkin et al. 2019)) or not feasible (e.g., a visual sketch of a suspect) (Wan, Gao, and Lee 2019). In such scenarios, comparison across unlabeled inputs is critical for learning, prediction, and ground truth generation – hence the popularity of similarity detection for various tasks including recommendation systems (Wei et al. 2019). However, there is no QML circuit designed to predict similarity on unlabeled data currently available.

SLIQ: Solution and Approach. To bridge this gap, a naïve design for similarity detection might create pairs of similar and dissimilar inputs from the training dataset, much like classical Siamese and Triplet networks (Chicco 2021; Li, Kan, and He 2020) (e.g., Anchor-Positive and Anchor-Negative pairs), and pass them sequentially over a variational quantum circuit (VQC)(Cerezo et al. 2021) to minimize the loss function. While straightforward, this design is resource-inefficient and does not fully leverage the unique properties of quantum computing. To address this gap in unsupervised QML, we propose SLIQ, which mitigates these challenges with multiple novel key design elements.

First, SLIQ addresses resource inefficiency by training both images in the pair at once via a superimposed state. This provides multiple advantages: (1) it reduces the overall quantum resource requirements by reducing the number of runs, and (2) it allows the VQC to learn more effectively since the superposition provides explicit hints about the similarity between the data. Next, to take advantage of entanglement in quantum computing systems, SLIQ interweaves the features of both inputs and explicitly entangles them at each layer in the VQC, decreasing the distance of corresponding features in Hilbert space. SLIQ's design ensures that interwoven features from different inputs are embedded into all physical qubits of the learning circuit so that the entanglement effects are captured in the measurement qubits. To ensure that SLIQ is practical and effective on current error-prone quantum computers, SLIQ keeps the number of parameters in the learning circuit minimal to mitigate the compounding noise effects on real quantum computers.

Unfortunately, incorporating superposition and entanglement properties creates new challenges. The identities of individual inputs in a pair (e.g., Anchor input in the Anchor-Positive pair) are indistinguishable due to entanglement, and the projection of the same input on the classical space is inconsistent across different runs. SLIQ introduces a new training "loss" estimation and improves quantum embedding methods to reduce projection variance, resulting in more robust training and a network that is resilient to hardware errors on real quantum systems. Overall, SLIQ demonstrates that the combination of training on entangled pairs and utilizing a projection variance-aware loss estimation yields effective similarity detection, even on current noisy quantum computers.

Contributions of SLIQ.

I. To the best of our knowledge, SLIQ is the first method to build a practical and effective *quantum learning circuit for similarity detection on NISQ-era quantum computers*. SLIQ is available as open-source framework at `https://github.com/SilverEngineered/SliQ`.

II. SLIQ's design demonstrates how to exploit the superposition and entanglement properties of quantum computing systems for similarity detection. It builds a *resource-efficient training pipeline by creating interwoven, entangled input pairs on a VQC, and it applies new robust methods for the quantum embedding of classical inputs.*

III. Our simulations and real-computer evaluations demonstrate that SLIQ achieves a 31% point improvement in similarity detection over a baseline quantum triplet network on a real-world, unlabeled dataset (Chen, Lai, and Liu 2018), while prior state-of-the-art works in QML only perform classification and require labeled input data (Huang et al. 2021b; Silver, Patel, and Tiwari 2022). We also show that SLIQ performs competitively for classification tasks on labeled data, despite not being a primary objective a similarity network.

Background

Qubits, Quantum Gates, and Quantum Circuits. A quantum bit (qubit) has the ability to attain a *superposition* of its two basis states: $|\Psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$. Here, $|0\rangle$ and $|1\rangle$ are the two basis states, α_0, and α_1 are normalized, complex-valued coefficients and Ψ is the overall qubit state in superposition. For an n-qubit computing system, the overall system state is represented as: $|\Psi\rangle = \sum_{k=0}^{k=2^n-1} \alpha_k |k\rangle$. When this state is *measured*, the *superposition collapses*, and the system is observed in state $|k\rangle$ with probability $\|\alpha_k\|^2$.

A qubit can be put in arbitrary superpositions using the $R3(p_1, p_2, p_3)$ *quantum gate*. The single-qubit $R3$ gate has three parameters (p_1, p_2, p_3) that can be adjusted to achieve the desired state (Cerezo et al. 2021). Multiple qubits can be *entangled* together to form an n-qubit system using two-qubit gates (e.g., the CX gate). These non-parameterized two-qubit gates and tunable $R3$ gates may be combined to achieve any n-qubit computation.

A sequence of quantum gates applied to a system of qubits forms a *quantum circuit*, at the end of which the qubits are measured to obtain the circuit's output. Fig. 1 shows an example of a quantum circuit in the "Variational Quantum Circuit (VQC)" box. The horizontal lines represent five qubit states, to which the $R3$ and two-qubit gates are applied over time, and the measurement gates are applied at the end.

Variational Quantum Circuits and Quantum Machine Learning. Whereas the gates in a normal quantum circuit are deterministic and predefined, a *Variational Quantum Circuit (VQC)* is a quantum circuit that utilizes parameterized gates that are tuned to optimize a certain objective (Cerezo et al. 2021). This objective can take many forms, from finding the minimal energy of a molecule's Hamiltonian to maximizing the rate of return of a financial portfolio or, in SLIQ's case, optimizing the loss function of a *quantum machine learning (QML)* task (Preskill 2018; Aaronson 2015). These circuits are "variational" because the gates' parameter values vary during the optimization process. These gates are adjusted according to a classical optimizer running on a classical machine, while the variational circuit itself is executed on a quantum computer. Fig. 1 demonstrates this hybrid feedback approach between
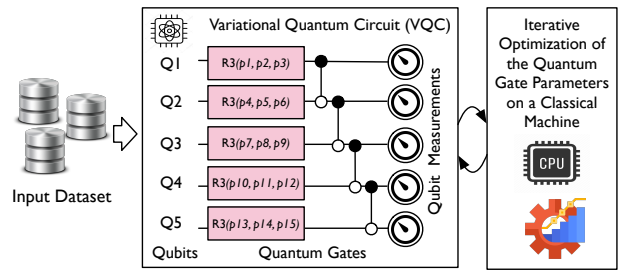

Figure 1: Hybrid quantum-classical procedure of executing and optimizing a variational quantum circuit (VQC).

quantum execution and classical optimization.

Although the optimization of VQC parameters is performed classically, a quantum advantage can be obtained from the circuit's execution on quantum hardware, which means the circuit has far fewer parameters to optimize compared to the classical version of the same algorithm (Chen et al. 2020). This advantage is gained by utilizing the superposition and entanglement properties on quantum computers that are not available on classical computers. For example, a classical image classification neural network typically consists of millions of parameters while a well-designed quantum network only requires hundreds (Cerezo et al. 2021). However, the accuracy of such quantum networks has been limited due to prevalent noise on current quantum computers (Preskill 2018), especially for unsupervised learning (Cerezo et al. 2021) – this paper aims to overcome this barrier.

Noise on NISQ Computers. Contemporary quantum computers suffer from noise during program execution due to various imperfections in the hardware of physical qubits, causing errors in the program output. SLIQ aims to achieve effective results in the face of these challenges.

SLIQ: Design and Implementation

In this section, we discuss the design of SLIQ and its key design elements. Before we discuss the design details of SLIQ, we first describe a base design of a quantum machine learning circuit for similarity detection. We refer to this design as *"Baseline"*. First, we discuss how the baseline design leverages widely-used variational quantum circuits to build a quantum learning network to perform the similarity detection task for labeled/unlabeled data. Next, we discuss the baseline design's resource inefficiency and its inability to exploit the power of superposition and entanglement. SLIQ's design addresses those limitations and provides superior performance, as discussed in our evaluation.

Baseline Design. The baseline design has three major components. The first component is generating the input which consists of triplets of Anchor, Positive, and Negative inputs – similar to classical Siamese-based and Triplet models which are widely used in the classical similarity networks (Veit, Belongie, and Karaletsos 2017; Ma et al. 2020). Then the encoding of the input features to the physical qubits is performed. To achieve this, we perform amplitude embedding (Schuld and Petruccione 2018) on the inputs one by one
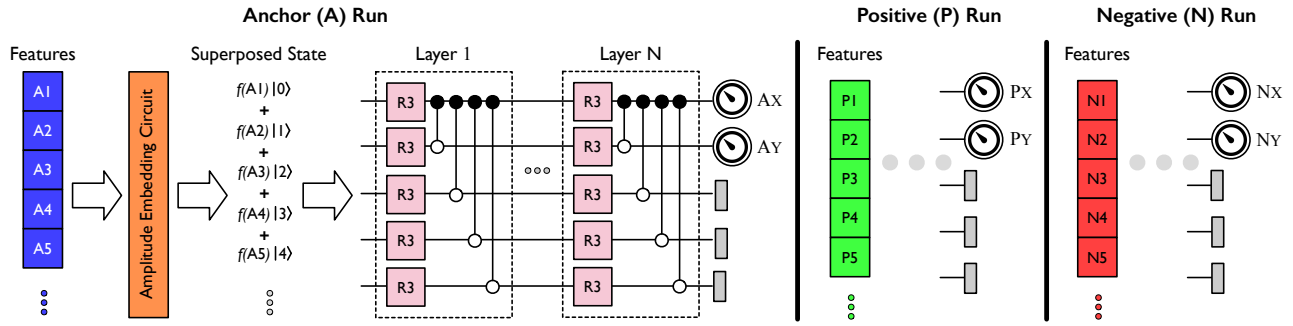
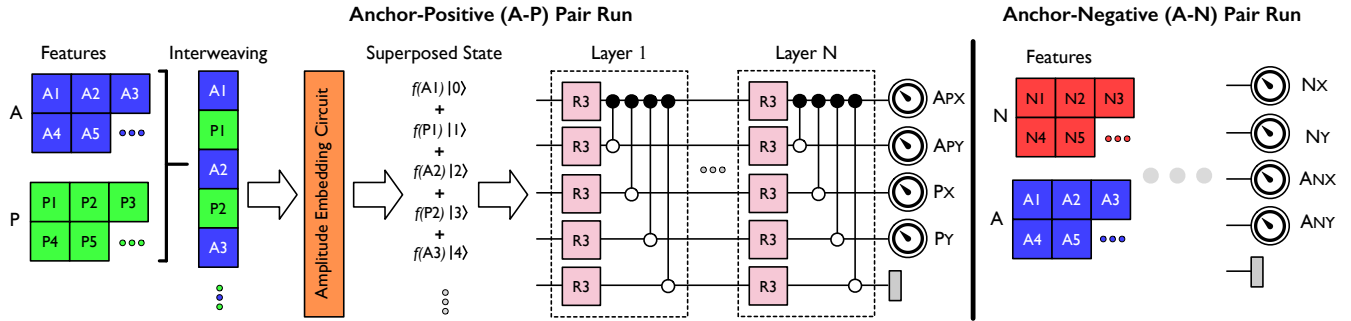Figure 2: The baseline design inputs one image at a time, requiring three separate runs for A, P, and N.

Figure 3: SLIQ's procedure of combining A-P and A-N inputs into two runs, interweaving their feature space, and updating the variational structure to leverage the properties of quantum superposition and entanglement to reduce the number of runs and generate better results.

for all three inputs (Fig. 2). The amplitude embedding procedure embeds classical data as quantum data in a Hilbert Space. Although recent prior works have utilized principal component analysis (PCA) prior to amplitude embedding for feature encoding (Silver, Patel, and Tiwari 2022), the baseline does employ PCA because higher performance is observed with keeping features intact and padding 0's as necessary to maintain the full variance of features. The second component is to feed these encoded features to a VQC for training and optimizing the VQC parameters to minimize the training loss (Fig. 2). The training loss is estimated via calculating the distance between the projection of inputs on a 2-D space – the projection is obtained via measuring two qubits at the end of the VQC circuit (this is the third and final component). The loss is calculated as the squared distance between the projection of the anchor, A, to the positive projection, P, taking the absolute distance between the anchor projection and negative projection N. This is an $L2$ variant of Triplet Embedding Loss and is formally defined as

$$L_2 = \left\{ (A_x - P_x)^2 + (A_y - P_y)^2 \right\} - \left\{ (A_x - N_x)^2 + (A_y - N_y)^2 \right\} \tag{1}$$

We note that the effectiveness of training can be adapted to any choice of dimension and shape of the projection space (2-D square box bounded between -1 and 1, in our case) as long as the choice is consistent among all input types (Anchor, Positive, and Negative inputs). A more critical feature is the repeating layers of the VQC which the baseline design chooses to be the same as other widely-used VQCs to make it competitive (Chicco 2021; Li, Kan, and He 2020).

SLIQ: Key Design Elements

SLIQ builds off the baseline design and introduces multiple novel design aspects to mitigate the limitations of the baseline design. First, SLIQ introduces the concept of training an input pair in the same run to leverage the superposition and entanglement properties of quantum computing systems.

Input Feature Entanglement and Interweaving. Recall that in the baseline design, each image type (Anchor, Positive, Negative) traverses through the variational quantum circuit one-by-one. The corresponding measurements at the end of each image-run produce coordinates on a 2-D plane that allows us to calculate similarity distance between A-P and A-N inputs. This allows us to calculate the loss that is targeted to be minimized over multiple runs during training. Unfortunately, this procedure requires performing three runs before the loss for a single (A, P, N) triplet input can be estimated, which is resource inefficient.

SLIQ's design introduces multiple new elements to address this resource inefficiency. The first idea is to create two training pairs (Anchor-Positive and Anchor-Negative), and each pair is trained in a single run (demonstrated visually in Fig. 3). This design element improves the resource-efficiency of the training process – instead of three runs (one for each image type), SLIQ requires only two runs. Note that, theoretically, it is possible to combine all three input types and perform combined amplitude embedding. However, in practice, this process is not effective because it be-
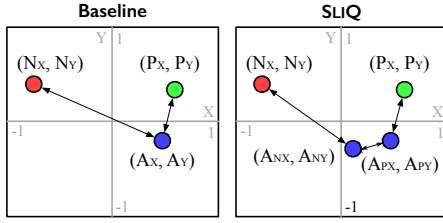
Figure 4: While the baseline design has zero projection variance, SLIQ has to take additional steps to mitigate it.
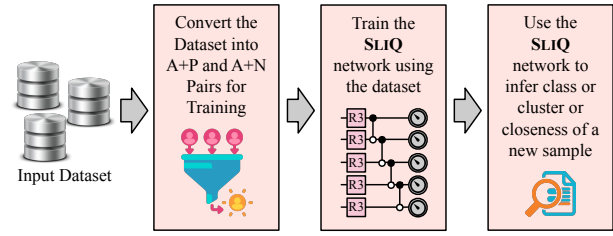

Figure 5: Overview of the design of SLIQ including the steps of preparing the data for training, training the quantum network, and using the network for inference post-training.

comes challenging for the quantum network to learn the distinction between the positive and negative input relative to the anchor input. Creating two pairs provides an opportunity for the quantum circuit to learn the similarity and dissimilarity in different pairs without dilution.

The second idea is to interweave the two input types in a pair before performing the amplitude embedding, and then feeding the output of the amplitude embedding circuit to the quantum circuit (Fig. 3). Interweaving provides the advantage of mapping features from different inputs to different physical qubits. This is particularly significant to mitigate the side-effects of noise on the current NISQ-era quantum machines where different physical qubits suffer from different noise levels (Tannu and Qureshi 2019). If interweaving of images is not performed, we risk the network not learning direct comparison between positionally equivalent features. SLIQ's interweaving mitigates this risk to make it effective on NISQ-era quantum computers which we found when compared to layering images.

As a final remark, we note that all these ideas are combined to leverage the power of entanglement and superposition of quantum systems – by training multiple inputs together, interweaving them, and creating superposition, and entangling them. While SLIQ's design to exploit superposition and entanglement is useful, it creates new challenges too. Next, we discuss the challenges of attaining projection invariance and novel solutions to mitigate the challenge.

Projection Variance Mitigation (PVM). Recall that in the baseline design, we measure two qubits and project the input's position in a 2-D space. Over three runs, we receive three separate coordinates in 2-D space, which we can use to calculate the loss – as shown in Fig. 4 (left). Our objective is to minimize the overall loss, defined as below:

$$L_{obj} = \left(|A_x - P_x| + |A_y - P_y| \right) - \left(|A_x - N_x| + |A_y - N_y| \right)$$
(2)

Optimizing for the above objective function is relatively straightforward. However, this loss function becomes nontrivial when SLIQ introduces the idea of training input pairs. Recall that the inputs are interweaved (Fig. 3), and hence, our measurements need to capture the outputs of anchor and positive/negative features separately. SLIQ resolves these issues by increasing the number of qubits we measure. Instead of two qubits per run, SLIQ measures four qubits. In Fig. 3, these qubits are denoted at the end of both runs. To distinguish the anchor input, SLIQ enforces two apriori-designated qubits measurements to correspond to the anchor

in both runs. We note it is not critical which qubits are chosen to "represent" the anchor input as long as our choice is treated as consistent. For example, qubits 1 and 2 could be tied to the anchor image, or qubits 1 and 3. So long as the choice does not change through training and evaluation, these options are computationally identical. However, this idea creates a major challenge – the coordinates corresponding to the anchor input may not project to the same point in our 2-D space. This is visually represented by two points (A_{NX}, A_{NY}) and (A_{PX}, A_{PY}) in Fig. 4. Ideally, these two points should project on the same coordinates.

The baseline design inherently has zero projection variance because it only has one measurement corresponding to the anchor input, and the loss for the positive and negative input was calculated from this absolute pivot. To mitigate this challenge, SLIQ designs a new loss function that accounts for minimizing this projection variance over training. As shown below, SLIQ's novel loss function has two components: (1) the traditional loss between the positive/negative input and the anchor input, and (2) new consistency loss. Consistency loss enforces positional embeddings to separate the entangled images at the time of measurement.

$$L_{pvm} = |A_{px} - A_{nx}| + |A_{py} - A_{ny}|$$
(3)

$$L_{total} = \alpha * L_{obj} + \beta * L_{pvm}$$
(4)

In Eq. 4, the parameters α and β are hyperparameters that denote weights for the objective function to balance the objective of accurate embeddings and precise embeddings. For L_{obj}, we use (A_{px}, A_{py}) and (A_{nx}, A_{ny}) for the positive and negative anchor values respectively. Additionally, to ensure robustness across the circuit, the samples are reversed for the negative case. The pair (A,P) is run along with the pair (N,A). The consistency is then applied between the mappings of the anchor image which now lie on different parts of the circuit. This additional measure ensures robustness by making the entire output of the circuit comply with the decided-upon separability as opposed to just a few qubits. This technique also enables scaling to entanglement of more than 2 images on a single machine.

In Fig. 5, we show an overview for the design of SLIQ. The first step is to take the input dataset and create pairs of the form (Anchor, Positive) and (Anchor, Negative) for training and testing. Once these are formed, the network trains on the dataset classically as part of the hybrid quantum-classical model used in VQCs. Once the data is
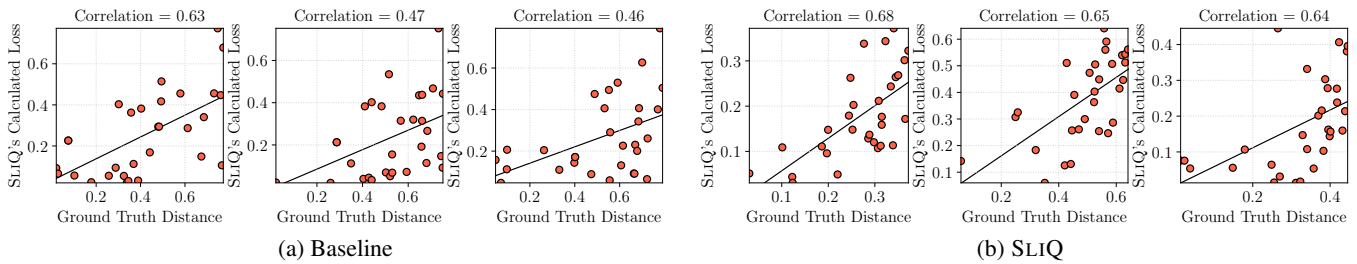
| (a) Baseline | (b) SLiQ |

Figure 6: SLiQ's loss normalized against ground truth distance between color frequencies. The anchor image was compared to 100 images. *Only the top 3 correlations are shown.* The steep drop in correlation metric shows that the baseline is not effective.

trained, SLiQ performs similarity detection by performing inference on new pairs of data. This inference can be used to identify the most similar samples to one another, the most distant samples, and even serve as a classifier if an unsupervised clustering algorithm is used to generate clusters.

Experimental Methodology

Training and Testing Datasets. SLiQ is evaluated on NIH AIDS Antiviral Screen Data (Kramer, De Raedt, and Helma 2001), MNIST (Deng 2012), Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017), and Flickr Landscape (Chen, Lai, and Liu 2018). These datasets are chosen because (1) they cover both unlabeled (e.g., Flickr Landscape) and labeled datasets (e.g., AIDS, MNIST, Fashion-MNIST), (2) they represent different characteristics (e.g., medical dataset, images, handwritten characters) and have real-world use cases (e.g., AIDS detection). We note that the size of the Flickr Landscape dataset is $\approx25\times$ larger than the commonly used quantum machine learning image datasets of MNIST and Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017; Silver, Patel, and Tiwari 2022). This presents additional scaling challenges that we mitigate with the scalable design of SLiQ.

Flickr Landscape is an unlabeled dataset consisting of 4300 images of different landscapes spanning general landscapes, mountains, deserts, seas, beaches, islands, and Japan. The images are different sizes, but are cropped to $80\times80\times3$ for consistency. The center is kept intact with all color channels. This dataset is unlabeled and serves to show how SLiQ performs on unlabeled data. NIH AIDS Antiviral Screen Data contains 50,000 samples of features alongside a label to indicate the status of the Aids virus ("CA" for confirmed active, "CM" for confirmed moderately active, and "CI" for confirmed inactive). The MNIST dataset is a grayscale handwritten digit dataset (Deng 2012) where we perform binary classification on '1's and '0's. The Fashion-MNIST dataset contains the same number of pixels and classes as MNIST, but the images are more complex in nature. In all datasets, 80% of the data is reserved for training and 20% reserved for testing.

Experimental Framework. The environment for SLiQ is Python3 with Pennylane (Bergholm et al. 2018) and Qiskit (Aleksandrowicz et al. 2019) frameworks. Our quantum experiment evaluations are simulated classically, and the inference results which are collected on the real IBM quantum

machines are specified in the text. For setting up the datasets for testing and training, triplets are created in the form of an anchor image, a positive image, and a negative image.

For the unlabeled dataset used, three images are selected at random. The color frequency is evaluated for each image on R, G, and B channels individually, then placed into eight bins for each color. The 24 total bins for each image are used to establish a ground truth similarity between the images, where the first image is the anchor and the images closest in L1 norm to the anchor is set as the positive image and the further away image is the negative image. Once the triplets are created, the pixels within the images are interwoven with one another. The image is then padded with 0s to the nearest power of 2 for the amplitude embedding process.

In the case of the labeled datasets, the anchor and positive are chosen to be from the same class, while the negative image is chosen at random from another class. For evaluation, an anchor and positive are passed into the embedding and the four dimensions are used in a Gaussian Mixture Model to form clusters which are then evaluated for accuracy. We use a batch size of 30 for all experiments with a GradientDescentOptimizer and a learning rate of 0.01. We train for 500 epochs on a four-layer network. The size of the network changes based on the dataset used, a four-qubit circuit is used for the Aids dataset, 14 for Flickr Landscape, and 11 for MNIST and Fashion-MNIST. For the baseline, one less qubit is used for all circuits as it does not necessitate the additional qubit to store an additional sample on each run.

Competing Techniques. Our baseline scheme is the same as described earlier: it is a quantum analogue of a triplet network, where weights are shared and use a single network for training and inference. Although SLiQ is not designed for classification tasks on labeled data, we provide a quantitative comparison with also the state-of-the-art quantum machine learning classifier: Projected Quantum Kernel (PQK) (published in Nature'2021) (Huang et al. 2021b) and Quilt (published in AAAI'2022) (Silver, Patel, and Tiwari 2022). The former, PQK, trains a classical network on quantum features generated by a specially designed quantum circuit. Datasets are modified to have quantum features that show the quantum advantage in training. While not feasible to implement on current quantum computers, we modify PQK's architecture to use fewer intermediate layers to reduce the number of gates and training time. The other comparison used, Quilt, is

Percentile	Baseline	SLIQ	Percentile	Baseline	SLIQ
25th	-0.06	0.23	75th	0.19	0.48
50th	0.05	0.36	100th	0.63	0.68

Table 1: Spearman correlation results show that SLIQ outperforms the baseline in similarity detection. SLIQ's median Spearman correlation is 0.36 vs. the Baseline's 0.05.
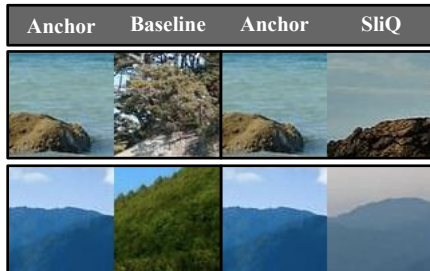

Figure 7: Anchor image, Baseline-identified similar image, and SLIQ-identified similar image for the same anchor image, using the unlabeled Flickr Dataset. SLIQ is effective in identifying images with similar color frequencies.

an image classifier built on an ensemble of variational quantum circuits which uses the ensemble to mitigate noise in the NISQ-era of quantum computing.

Figures of Merit. We categorize the figures of merit by dataset type: unlabeled or labeled. As image similarity is inherently visual, in addition to quantitative success, we demonstrate qualitative success by including a few relevant snapshots. For our unlabeled results, we also examine quantitatively how well the overall predictions match with the ground truth similarities, showing how well SLIQ learns over a distribution. The specific details of the metrics are described near the description of the results. For qualitative success in the unlabeled dataset, we show the model's predicted images for most similar to a given anchor image. For our labeled datasets, we report the accuracy of SLIQ. Because SLIQ performs an embedding and does not classify, accuracy can not be obtained directly from the output. For accuracy metrics, Gaussian Mixture Models are used to assign clusters for classification.

Evaluation and Analysis

SLIQ effectively detects the similarity of samples in unlabeled datasets using quantum networks; SLIQ is the first work to target this area. As SLIQ is the first work to target quantum similarity detection for unlabeled data, we compare against the results for the baseline quantum triplet model. Using the Flickr Landscape dataset, we rank the image similarity based on triplets formed from color frequency histograms. For each image in a set of 100 images, we treat the image as an anchor and compute the ground truth distance in color frequency between the anchor and every other image. We compare this ground truth distance to the distance identified by the model and correlate the rankings.

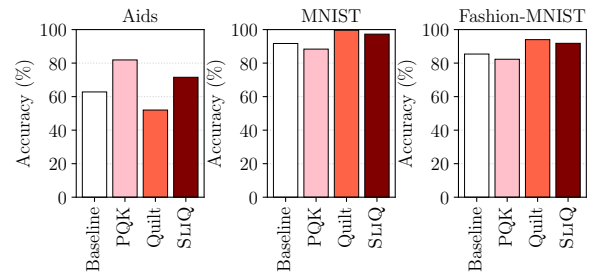We use Spearman correlation (Myers and Sirois 2004),

Figure 8: SLIQ performs competitively against the comparative classification techniques for all datasets, despite classification not being a learned objective of SLIQ.

which performs ranked correlation of two random variables, to interpret the relationship between ground truth and the model's estimations. Spearman correlation is commonly used to perform this type of analysis, for example (Reimers and Gurevych 2019) uses Spearman correlation in ranking sentence embedding in similarity networks. SLIQ *has much better correlations over the baseline triplet model, with a median Spearman correlation of 0.36 compared to a median Spearman correlation of 0.05, which shows that* SLIQ *is 0.31 or 31% points more accurately correlated than Baseline.* In Table **??**, we show the distribution of Spearman correlations for SLIQ compared to the baselines. At every percentile measured, SLIQ has notable improvements in similarity detection, which demonstrates SLIQ's overall improvement over an entire distribution.

This trend is also visually observed in Fig. 6(a) for Baseline and Fig. 6(b) for SLIQ. The x-axis denotes the ground truth distance, where the points further to the left indicate true similarity. The y-axis denotes the calculated loss of the model, indicating SLIQ's ground truth distance estimate. Points closer to the diagonal line denote accurate estimations. Fig. 6 show only the top-3 correlation examples for easier visual interpretation. We note that SLIQ tends to cluster more around the ground truth correlation line, and its top-3 correlation drop from 0.68 to 0.64; in contrast, the baseline, drops from 0.63 to 0.46.

Additionally, we show some of the closest predictions identified in Fig. 7. These demonstrate the different scenes of the Landscape dataset. For example, the demonstrated landscapes include mountains, aerial perspectives, forests, and oceans. SLIQ is able to match similar landscapes efficiently, demonstrating its effectiveness at similarity detection – better than the baseline's relative picks.

Although SLIQ was not designed to act as a classifier, it is effective at detecting the similarity of samples in labeled datasets and is competitive with prior state-of-the-art quantum classifiers. On its own, SLIQ performs embeddings, not classification, but we can use SLIQ as a classifier by performing reasonable clustering on its output. To demonstrate its classification ability, we compare against state-of the art quantum classification methods (Huang et al. 2021b; Silver, Patel, and Tiwari 2022). Our results (Fig. 8) indicate that SLIQ yields competitive accuracy compared to the advanced quantum classifiers on a task that SLIQ was

Environment	Baseline	PQK	Quilt	SLIQ
Simulation	60.8%	81.6%	51%	71.54%
Real Computer	64.4%	N/A	21.4%	68.8%

Table 2: SLIQ's real quantum computer results for the AIDS dataset are consistent with the simulation results, showing its resilient to hardware noise due to its low number of parameters. The PQK column is denoted as N/A circuit is prohibitively deep for it to be feasible and run on error-prone NISQ computers (30× more parameters than SLIQ).
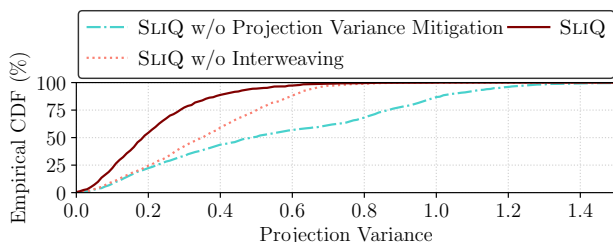

Figure 9: SLIQ achieves a lower projection variance compared to when it is run without projection variance mitigation or without interweaving of images.

never trained on (classification) and demonstrates the broad applicability of embeddings. To perform this clustering, we employ a Gaussian Mixture Model on our embeddings. The model is initialized with the true number of classes in the dataset and is fit to 1000 samples to form clusters. For classification, each permutation of the clusters to labels is considered, as these models do not provide labels. The permutation with the highest accuracy is considered to be the correct label. With these techniques, our results show SLIQ performs well on a variety of datasets, averaging up to 96.44% accuracy on binary MNIST classification. We show the full classification accuracy results in Fig. 8 for different techniques.

In Table **??**, we show how SLIQ performs on real quantum computers today. SLIQ achieves a 68.8% accuracy on the AIDS dataset, running on IBM Oslo. SLIQ *significantly outperforms the state-of-the-art quantum classifier (QUILT), even though* SLIQ *was not originally designed for classification.* The reason is because SLIQ's design is noise-aware to be effective on error-prone NISQ computers. In particular, SLIQ is designed with few parameters for current NISQ computers, where error compounds at each step and quickly explodes. Other quantum machine learning architectures tend to have higher degradation in accuracy on real computers as they require larger architectures with ample opportunity for compounding error. For the AIDS dataset, SLIQ has 48 tunable parameters, while Quilt requires 375, and PQK requires 1,633 parameters. As a result of more parameters, the hardware error compounds, explaining the degradation shown above.

Why does SLIQ perform effectively? By mitigating projection variance, SLIQ is able to map the anchor input at the same position to a close location regardless of the second input in the pair. This is necessary as the images get entangled together throughout the entire circuit and will not be separated in the output unless a constraint is put in place to enforce this. This separability can be demonstrated in Fig. 9 where SLIQ is compared to a trained version without consistency loss. SLIQ has more precise outputs throughout the entire CDF, evaluated over 1000 MNIST test images. Enforcing consistency amounts to an average decrease of 80% in projection variance when changing the order of the inputs – demonstrating the effectiveness of SLIQ's projection invariance method. As shown in Fig. 9, interweaving increasing robustness, leading to a decrease in projection variance.

Related Work

Classical Similarity Networks. Siamese networks and triplet networks are commonly-used classical similarity networks (Johnson, Douze, and Jegou 2021; Koch et al. 2015; Schroff, Kalenichenko, and Philbin 2015; Roy et al. 2019; Li, Kan, and He 2020; Gichane et al. 2020; Li et al. 2020; Hoffer and Ailon 2015; Patel, Merali, and Wetzel 2022), as they known are to be the best choice for complex datasets (Chicco 2021). As an instance, using the Riemannian geometry to train the Siamese network, Roy et al. (Roy et al. 2019) get effective results for image classification, while FaceNet (Schroff, Kalenichenko, and Philbin 2015) achieves representational efficiency using an embedding for face recognition and clustering. On the other hand, TrimNet (Li et al. 2020) uses a graph-based approach toward enabling a triplet network to learn molecular representations. However, while these works are effective classically, quantum theory enables the enhancement of machine learning workloads by reducing their size and speeding them up (Aaronson 2015; Daley et al. 2022).

Quantum Machine Learning. Extensive work has been performed toward porting a wide variety of machine learning tasks to quantum computing (Huang et al. 2021a; Li and Kais 2021; Tiwari and Melucci 2019; Li, Song, and Wang 2021; Khairy et al. 2020; Lockwood and Si 2020; Heidari, Grama, and Szpankowski 2022; Lloyd et al. 2020; Radha and Jao 2022; Nandy Pal, Banerjee, and Sarkar 2022). This includes porting workloads such as generalized neural networks (Beer et al. 2020), convolutional neural networks (Hur, Kim, and Park 2022), and even application-specific networks such as models used to learn the metal-insulator transition of VO_2 (Li and Kais 2021).

Quantum Image Similarity Detection (Liu, Qi, and Liu 2022) and (Liu et al. 2019) have also worked on quantum image similarity detection; notably, these did not take a machine-learning approach to identify similarity.

Conclusion

In this work, we present SLIQ, a resource-efficient quantum similarity network, which is the first method to build a practical and effective quantum learning circuit for similarity detection on NISQ-era computers. We show that SLIQ improves similarity detection over a baseline quantum triplet network by 31% points for Spearman correlation. SLIQ is available at: https://github.com/SilverEngineered/SliQ.

Acknowledgements

References

Aaronson, S. 2015. Read the fine print. *Nature Physics*, 11(4): 291–293.

Aleksandrowicz, G.; Alexander, T.; Barkoutsos, P.; Bello, L.; Ben-Haim, Y.; Bucher, D.; Cabrera-Hernández, F. J.; Carballo-Franquis, J.; Chen, A.; Chen, C.-F.; et al. 2019. Qiskit: An open-source framework for quantum computing. *Accessed on: Mar*, 16.

Beer, K.; Bondarenko, D.; Farrelly, T.; Osborne, T. J.; Salzmann, R.; Scheiermann, D.; and Wolf, R. 2020. Training deep quantum neural networks. *Nature Communications*, 11(1): 808.

Bergholm, V.; Izaac, J.; Schuld, M.; Gogolin, C.; Alam, M. S.; Ahmed, S.; Arrazola, J. M.; Blank, C.; Delgado, A.; Jahangiri, S.; et al. 2018. Pennylane: Automatic differentiation of hybrid quantum-classical computations. *arXiv preprint arXiv:1811.04968*.

Cerezo, M.; Arrasmith, A.; Babbush, R.; Benjamin, S. C.; Endo, S.; Fujii, K.; McClean, J. R.; Mitarai, K.; Yuan, X.; Cincio, L.; et al. 2021. Variational quantum algorithms. *Nature Reviews Physics*, 3(9): 625–644.

Chen, S. Y.-C.; Yang, C.-H. H.; Qi, J.; Chen, P.-Y.; Ma, X.; and Goan, H.-S. 2020. Variational quantum circuits for deep reinforcement learning. *IEEE Access*, 8: 141007–141024.

Chen, Y.; Lai, Y.-K.; and Liu, Y.-J. 2018. Cartoongan: Generative adversarial networks for photo cartoonization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 9465–9474.

Chicco, D. 2021. *Siamese Neural Networks: An Overview*, 73–94. New York, NY: Springer US. ISBN 978-1-0716-0826-5.

Daley, A. J.; Bloch, I.; Kokail, C.; Flannigan, S.; Pearson, N.; Troyer, M.; and Zoller, P. 2022. Practical quantum advantage in quantum simulation. *Nature*, 607(7920): 667–676.

Deng, L. 2012. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6): 141–142.

Gichane, M. M.; Byiringiro, J. B.; Chesang, A. K.; Nyaga, P. M.; Langat, R. K.; Smajic, H.; and Kiiru, C. W. 2020. Digital Triplet Approach for Real-Time Monitoring and Control of an Elevator Security System. *Designs*, 4(2).

Guan, W.; Perdue, G.; Pesah, A.; Schuld, M.; Terashi, K.; Vallecorsa, S.; and Vlimant, J.-R. 2021. Quantum machine learning in high energy physics. *Machine Learning: Science and Technology*, 2(1): 011003.

Heidari, M.; Grama, A.; and Szpankowski, W. 2022. Toward Physically Realizable Quantum Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(6): 6902–6909.

Hoffer, E.; and Ailon, N. 2015. Deep metric learning using triplet network. In *Similarity-Based Pattern Recognition: Third International Workshop, SIMBAD 2015, Copenhagen, Denmark, October 12-14, 2015. Proceedings 3*, 84–92. Springer.

Huang, H.-Y. R.; Broughton, M. B.; Cotler, J.; Chen, S.; Li, J.; Mohseni, M.; Neven, H.; Babbush, R.; Kueng, R.; Preskill, J.; and McClean, J. R. 2021a. Quantum advantage in learning from experiments. *Science*, 376: 1182 – 1186.

Huang, H.-Y. R.; Broughton, M. B.; Mohseni, M.; Babbush, R.; Boixo, S.; Neven, H.; and McClean, J. R. 2021b. Power of data in quantum machine learning. *Nature Communications*, 12: 2631.

Hur, T.; Kim, L.; and Park, D. K. 2022. Quantum convolutional neural network for classical data classification. *Quantum Machine Intelligence*, 4(1): 3.

Johnson, J.; Douze, M.; and Jegou, H. 2021. Billion-Scale Similarity Search with GPUs. *IEEE Transactions on Big Data*, 7(03): 535–547.

Khairy, S.; Shaydulin, R.; Cincio, L.; Alexeev, Y.; and Balaprakash, P. 2020. Learning to Optimize Variational Quantum Circuits to Solve Combinatorial Problems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(03): 2367–2375.

Koch, G.; Zemel, R.; Salakhutdinov, R.; et al. 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille.

Kramer, S.; De Raedt, L.; and Helma, C. 2001. Molecular feature mining in HIV data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 136–143.

Li, G.; Song, Z.; and Wang, X. 2021. VSQL: Variational Shadow Quantum Learning for Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9): 8357–8365.

Li, J.; and Kais, S. 2021. Quantum cluster algorithm for data classification. *Materials Theory*, 5(1): 6.

Li, P.; Li, Y.; Hsieh, C.-Y.; Zhang, S.; Liu, X.; Liu, H.; Song, S.; and Yao, X. 2020. TrimNet: learning molecular representation from triplet messages for biomedicine. *Briefings in Bioinformatics*, 22(4). Bbaa266.

Li, Y.; Kan, S.; and He, Z. 2020. Unsupervised Deep Metric Learning with Transformed Attention Consistency and Contrastive Clustering Loss. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *Computer Vision – ECCV 2020*, 141–157. Cham: Springer International Publishing.

Liu, X.; Zhou, R.-G.; El-Rafei, A.; Li, F.-X.; and Xu, R.-Q. 2019. Similarity assessment of quantum images. *Quantum Information Processing*, 18(8): 1–19.

Liu, Y.-h.; Qi, Z.-d.; and Liu, Q. 2022. Comparison of the similarity between two quantum images. *Scientific reports*, 12(1): 1–10.

Lloyd, S.; Schuld, M.; Ijaz, A.; Izaac, J.; and Killoran, N. 2020. Quantum embeddings for machine learning. *arXiv preprint arXiv:2001.03622*.

Lockwood, O.; and Si, M. 2020. Reinforcement Learning with Quantum Variational Circuit. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 16(1): 245–251.

Ma, Z.; Dong, J.; Long, Z.; Zhang, Y.; He, Y.; Xue, H.; and Ji, S. 2020. Fine-grained fashion similarity learning by attribute-specific embedding network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 11741–11748.

Myers, L.; and Sirois, M. J. 2004. Spearman correlation coefficients, differences between. *Encyclopedia of statistical sciences*, 12.

Nandy Pal, M.; Banerjee, M.; and Sarkar, A. 2022. Retrieval of exudate-affected retinal image patches using Siamese quantum classical neural network. *IET Quantum Communication*, 3(1): 85–98.

Patel, Z.; Merali, E.; and Wetzel, S. J. 2022. Unsupervised learning of Rydberg atom array phase diagram with Siamese neural networks. *New Journal of Physics*, 24(11): 113021.

Preskill, J. 2018. Quantum Computing in the NISQ era and beyond. *Quantum*, 2: 79.

Radha, S. K.; and Jao, C. 2022. Generalized quantum similarity learning. *arXiv preprint arXiv:2201.02310*.

Reimers, N.; and Gurevych, I. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Roy, S.; Harandi, M.; Nock, R.; and Hartley, R. 2019. Siamese Networks: The Tale of Two Manifolds. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 3046–3055.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 815–823.

Schuld, M.; and Petruccione, F. 2018. *Supervised learning with quantum computers*, volume 17. Springer.

Silver, D.; Patel, T.; and Tiwari, D. 2022. QUILT: Effective Multi-Class Classification on Quantum Computers Using an Ensemble of Diverse Quantum Classifiers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 8324–8332.

Tannu, S. S.; and Qureshi, M. K. 2019. Not all qubits are created equal: A case for variability-aware policies for NISQ-era quantum computers. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, 987–999.

Tiwari, P.; and Melucci, M. 2019. Binary Classifier Inspired by Quantum Theory. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01): 10051–10052.

Utkin, L.; Meldo, A.; Kovalev, M.; and Kasimov, E. 2019. An ensemble of triplet neural networks for differential diagnostics of lung cancer. In *2019 25th Conference of Open Innovations Association (FRUCT)*, 346–352. IEEE.

Veit, A.; Belongie, S.; and Karaletsos, T. 2017. Conditional similarity networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 830–838.

Wan, W.; Gao, Y.; and Lee, H. J. 2019. Transfer deep feature learning for face sketch recognition. *Neural Computing and Applications*, 31(12): 9175–9184.

Wei, Y.; Cheng, Z.; Yu, X.; Zhao, Z.; Zhu, L.; and Nie, L. 2019. Personalized hashtag recommendation for micro-videos. In *Proceedings of the 27th ACM International Conference on Multimedia*, 1446–1454.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.