QTML 2021

Quantum Techniques in Machine Learning

Booklet of Extended Abstracts

POSTERS - Part I

QUANTUM STATE DISCRIMINATION FOR MULTICLASS CLASSIFICATION

ROBERTO GIUNTINI, HECTOR FREYTES, DANIEL K. PARK, CARSTEN BLANK, FEDERICO HOLIK, KENG LOON CHOW, AND GIUSEPPE SERGIOLI

Abstract. In this talk we focus on the connection between quantum information theory and machine learning. In particular, we show how quantum state discrimination can be used as a tool to address the standard classification problem in machine learning. Previous studies have shown that the optimal quantum measurement theory developed in the context of quantum information theory and quantum communication, can inspire a new binary classification algorithm that can achieve higher inference accuracy for various datasets. Here we propose a model for arbitrary multiclass classification inspired by quantum state discrimination, which is performed by encoding classical data in the space of linear operators on a Hilbert space. Given that our algorithm is quantum-inspired, it can be implemented on purely classical hardware, thereby allowing for immediate applications.

1. Long astract

Quantum theory gives place to a new paradigm for information processing and provides unconventional ways to address computational problems. Advances in quantum computing have led to the development of algorithms that utilize the peculiar properties of quantum systems to solve certain problems dramatically faster than any foreseeable classical hardware [1, 2, 3, 4]. A commercially relevant family of problems for which the application of quantum algorithms promises certain computational benefits are found in the domain of machine learning. This gave birth to the new discipline known as quantum machine learning (QML). Several quantum machine learning algorithms have been proposed with clear quantum advantages [5, 6, 7, 8, 9, 10]. However, the practical application of these algorithms is limited by the development of quantum hardware, which remains a long-term prospect.

Advances in quantum computing have led to another intriguing stream of research which aims to develop new classical algorithms inspired by the mathematical structure of quantum theory (in particular quantum information processing) to outperform existing methods, namely quantum-inspired classical algorithms [11, 12, 13]. The implications of this approach are significant, not only in the context of computational complexity theory, but also in practical applications. The difficulties involved in implementing quantum machine learning algorithms on quantum hardware and the emergence of quantum-inspired classical algorithms in the NISQ era motivated the development of quantum-inspired machine learning (QIML) [14]. In principle, QIML deals only with "mathematically quantum objects": objects that are formally represented by different elements of the quantum formalism (such as density operators), but are not necessarily connected to actual quantum systems. Thus, the information stored in those objects can be formally managed by a classical computer. Recent findings show that the well-developed field of quantum state discrimination in quantum information theory and quantum communication, can inspire new pattern recognition algorithms that can improve the binary classification accuracy of extant methods [15, 16, 17, 18].

In this talk, we will first explain the connection between quantum state discrimination and the problem of quantum-inspired binary classification, and then propose a quantum-inspired supervised machine learning algorithm for arbitrary multiclass classification. Our algorithm makes use of the mathematical framework of quantum mechanics to represent data and a quantum state discrimination technique known as *Pretty Good Measurement.* We show the theoretical derivation of this measurement strategy in the context of multiclass classification tasks in machine learning. We also show how the classification accuracy of this quantum-inspired multiclass classifier can be improved by increasing the number of copies of the quantum object that encodes the data – at the cost of increasing the computational time. It is important to stress the fact that, since our approach does not require quantum hardware, it can be immediately implemented on existing classical hardware.

In the last part of the talk we show some experiments performed over artificial datasets, where we compare the quantum-inspired multiclass classifier with other standard classifiers by considering time complexity, accuracy and other relevant statistical quantities. The results of the experiment show how the performances of the quantum inspired classifier are quite promising compared with other standard classifiers. This leads to a double potential benefit: the first is the improvement of the classification accuracy for certain datasets and the second is to have the opportunity to run the algorithm over a real quantum computer with a potential reduction of the complexity.
Part of the results to be presented are summarized in $[arXiv : 2104.00971v1]$.

References

[1] Seth Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, 1996.

[2] Christof Zalka. Simulating quantum systems on a quantum computer. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1969):313–322, 1998.

[3] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[4] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103:150502, Oct 2009.

[5] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Phys. Rev. Lett.*, 113:130503, Sep 2014.

[6] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631–633, 2014.

[7] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. Prediction by linear regression on a quantum computer. *Phys. Rev. A*, 94:022342, Aug 2016.

[8] Nana Liu and Patrick Rebentrost. Quantum machine learning for quantum anomaly detection. *Phys. Rev. A*, 97:042315, Apr 2018.

[9] Carsten Blank, Daniel K. Park, June-Koo Kevin Rhee, and Francesco Petruccione. Quantum classifier with tailored quantum kernel. *npj Quantum Information*, 6(1):41, May 2020.

[10] Daniel K. Park, Carsten Blank, and Francesco Petruccione. The theory of the quantum kernel-based binary classifier. *Physics Letters A*, 384(21):126422, 2020.

[11] Ewin Tang. A quantum-inspired classical algorithm for recommendation systems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 217–228, New York, NY, USA, 2019. Association for Computing Machinery.

[12] Ewin Tang. Quantum-inspired classical algorithms for principal component analysis and supervised clustering, 2019. arXiv:1811.00414.

[13] Juan Miguel Arrazola, Alain Delgado, Bhaskar Roy Bardhan, and Seth Lloyd. Quantum-inspired algorithms in practice. *Quantum*, 4:307, August 2020.

[14] Federico Holik, Giuseppe Sergioli, Hector Freytes, and Angelo Plastino. Pattern recognition in non-kolmogorovian structures. *Foundations of Science*, 23(1):119–132, 2018.

[15] Giuseppe Sergioli, Roberto Giuntini, and Hector Freytes. A new quantum approach to binary classification. *PLOS ONE*, 14(5):1–14, 05 2019.

[16] Giuseppe Sergioli, Gustavo Martin Bosyk, Enrica Santucci, and Roberto Giuntini. A quantum-inspired version of the classification problem. *International Journal of Theoretical Physics*, 56(12):3880–3888, 2017.

[17] Giuseppe Sergioli, Giorgio Russo, Enrica Santucci, Alessandro Stefano, Sebastiano Emanuele Torrisi, Stefano Palmucci, Carlo Vancheri, and Roberto Giuntini. Quantum-inspired minimum distance classification in a biomedical context. *International Journal of Quantum Information*, 16(08):1840011, 2018.

[18] Giuseppe Sergioli, Carmelo Militello, Leonardo Rundo, Luigi Minafra, Filippo Torrisi, Giorgio Russo, Keng Loon Chow, and Roberto Giuntini. A quantum-inspired classifier for clonogenic assay evaluations. *Scientific Reports*, 11(1):2830, 2021.

(R. Giuntini) University of Cagliari, Via Is Mirrionis 1, I-09123 Cagliari, Italy.
Email address: giuntini@unica.it

(H. Freytes) University of Cagliari, Via Is Mirrionis 1, I-09123 Cagliari, Italy.
Email address: hfreytes@gmail.com

(D.K. Park) Sungkyunkwan University Advanced Institute of Nanotechnology, Suwon, Korea.
Email address: dkp.quantum@gmail.com

(C. Blank) data cybernetics, Landsberg, Germany
Email address: blank@data-cybernetics.com

(F. Holik) University of La Plata, Physics Institute
Email address: olentiev2@gmail.com

(K.L. Chow) University of Cagliari, Via Is Mirrionis 1, I-09123 Cagliari, Italy.
Email address: leo.chow11@gmail.com

(G. Sergioli) University of Cagliari, Via Is Mirrionis 1, I-09123 Cagliari, Italy.
Email address: giuseppe.sergioli@gmail.com

Scaling properties of pre-trained neural-network-based quantum state tomography

Sanjaya Lohani[1, 2, *], **Brian T. Kirby**[3, 4], **Ryan T. Glasser**[4], **and Thomas A. Searles**[1]

[1]University of Illinois Chicago, Chicago, IL 60607, USA
[2]IBM-HBCU Quantum Center, Howard University, Washington, DC 20059, USA
[3]United States Army Research Laboratory, Adelphi, MD 20783, USA
[4]Tulane University, New Orleans, LA 70118, USA
[*]slohan3@uic.edu

Abstract

We determine the scaling properties of a neural-network-based quantum state tomography technique and find a significant reduction in the inference time compared to both maximum likelihood estimation and adaptive neural network methods. The results indicate that state reconstruction with pre-trained networks is a practical approach for rapidly characterizing NISQ devices.

Introduction

The reconstruction of a physically valid density matrix from tomographic measurements is a resource-intensive problem with broad applicability in quantum information science[1,2]. Recently, several competing methods for state reconstruction based on machine learning have been proposed and demonstrated[3–12]. These techniques fall into two categories, those which are entirely pre-trained and those which adapt based on measurement results[13,14]. A benefit of the former is the ability to front-load expensive computations, offering a potentially dramatic reduction in inference time as compared to standard reconstruction methods[12,15].

Here we discuss the scaling properties of both training and inference for a pre-trained machine-learning-based quantum state reconstruction system. We explicitly limit our system to pure states in an effort to achieve both the highest possible reconstruction fidelity and the most favorable resource scaling. We find that our networks reconstruct states of up to four qubits comparably to an MLE process restricted to pure states but with a higher variance. Based on our results, we estimate that training time increases exponentially with qubit number, but that inference time remains linear.

Methods

We consider the problem of reconstructing the quantum state of a d qubit system most consistent with a set of a complete set of repeated tomographic measurements. In particular, we assume the state under investigation has been measured repeatedly by a set of projectors $\hat{\Pi}$ containing the 6^d eigenvectors associated with the 3^d combinations of operators $\{X,Y,Z\}_1 \otimes \{X,Y,Z\}_2 \otimes \ldots \otimes \{X,Y,Z\}_d$ where the subscripts indicate qubit number. In the limit of having infinite copies of the unknown state with which to build measurement statistics, the measurement probabilities associated with each projector would be given by $\bar{n}_i = \mathrm{Tr}\big(\rho \hat{\Pi}_i\big)$, where 'Tr' represents the trace. In practice the measured probabilities include statistical noise from having finite copies and noise from imperfect experimental implementations. The problem of inverting measured probabilities to find a physical valid ρ is known to be a resource intensive problem[1,2].

To perform this reconstruction we build a custom-designed convolutional neural network with a convolutional unit of kernel size (2, 2), strides of 1, ReLU as an activation function, and 25 filters. The filters are followed by a max-pooling layer with pool-size (2, 2) and a second convolutional unit with the same configuration in a row. We then connect two dense layers, each followed by a dropout layer with a rate of 0.5, which is finally attached to an output layer that predicts ρ. Then, the predicted ρ are forwarded to an error layer, where mean square loss between the target and predicted ρ is evaluated and fed back to optimize the network's training using the Adagrad optimizer with a learning rate of 0.005 for up to 300 epochs. We also note that network is constrained to reconstruct physically valid density matrices through the use of a Cholesky decomposition[16]. Additionally, while our system is trained exclusively on pure states it is not actually constrained to only pure states and can in principle reconstruct mixed

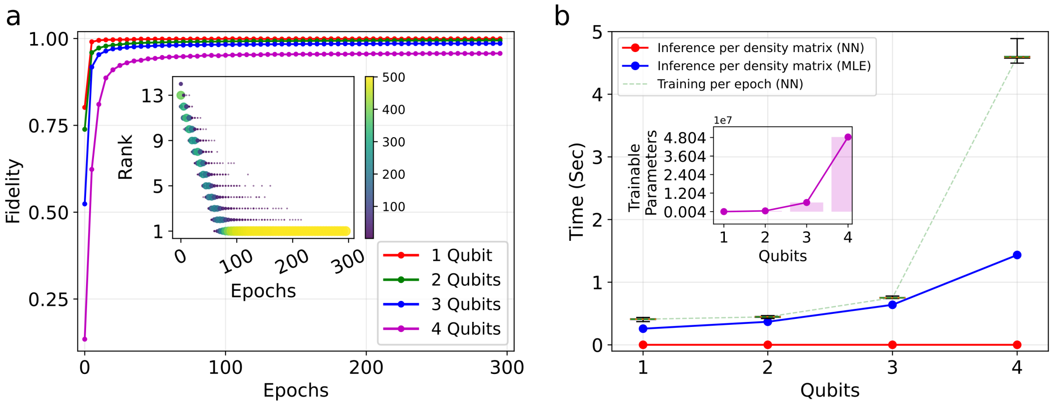


Figure 1. (a) Neural network reconstruction fidelity for simulation pure quantum states as a function of training epochs. Inset shows the rank of the reconstructed state for the four-qubit network. (b) Resource scaling for training and inference for the neural network and inference time for reconstruction of the same states using maximum likelihood estimation. The inset shows the trainable parameters of the networks used in this demonstration.

states.

We train the network using 35,500 random quantum states according to the Haar measure and the associated 6^d tomographic measurement for systems with qubit numbers ranging from one to four. We split the simulated data into a training set of size 35,000 and a validation set of size 500 to cross-validate the network performance. After training, we generate test sets that are entirely unknown to the trained network. Note that we keep the depth (number of hidden layers) of the network fixed for all cases; instead, we increase the number of the neurons in the dense layers with increasing qubit number. Finally, for comparison purposes, we also implement maximum likelihood estimation (MLE) with a multinomial likelihood constrained to reconstruct only pure-states.

Results and Discussion

We now consider the scaling properties of our neural network in terms of both training and inference and compare this with maximum likelihood estimation. In order to evaluate the fidelity (F) between the density matrix predicted by our network (ρ_{nn}) and the target (ρ_t), we use $F = \left|\mathrm{Tr}\sqrt{\sqrt{\rho_{nn}}\rho_t\sqrt{\rho_{nn}}}\right|^2$. In Fig. 1(a), we plot the fidelity F as a function of training epochs for simulated systems of sizes one through four. We see that after approximately 100 epochs, all four networks have reached maximum fidelity, with the network for a single-qubit requiring fewer than ten epochs. In the inset, we have included the rank of the reconstructed density matrix for our four-qubit system as a function of epochs. The size and color of the markers are proportional to the number of states found with a given fidelity. Despite our network being only constrained to reconstruct physical and not pure density matrices, we see that the network quickly learns to construct states of low rank.

In Fig. 1(b), we now show the inference and training time as a function of the number of qubits in the system for both our neural network and maximum likelihood estimation. We see that in terms of inference, the neural network (red) scales linearly with a four-qubit reconstruction requiring only 0.8ms. Comparatively, the maximum likelihood estimation for the same states (blue) scales unfavorably with system size and takes orders of magnitude longer. Of course, the rapid inference times come at the cost of imperfect reconstruction fidelity, as seen in Fig. 1(a), and an upfront training cost shown by the green line in Fig 1(b). The training time scales exponentially, with the four-qubit system requiring over four seconds per epoch. Therefore, both maximum likelihood estimation and our pre-trained neural network method require exponential resources in the size of the system; however, the network has the advantage that these resources can be expended ahead of time and only once.

We acknowledge the funding under contract/grant numbers W911NF-19-2-0087, W911NF-20-2-0168, and DE-SC0012704. We also thank the IBM-HBCU Quantum Center at Howard University.

References

1. Hou, Z. *et al.* Full reconstruction of a 14-qubit state within four hours. *New J. Phys.* **18**, 083036, DOI: 10.1088/1367-2630/18/8/083036 (2016).
2. Smolin, J. A., Gambetta, J. M. & Smith, G. Efficient Method for Computing the Maximum-Likelihood Quantum State from Measurements with Additive Gaussian Noise. *Phys. Rev. Lett.* **108**, 070502, DOI: 10.1103/PhysRevLett.108.070502 (2012).
3. Carrasquilla, J., Torlai, G., Melko, R. G. & Aolita, L. Reconstructing quantum states with generative models. *Nat. Mach. Intell.* **1**, 155–161 (2019).
4. Cha, P. *et al.* Attention-based quantum tomography. *arXiv preprint arXiv:2006.12469* (2020).
5. Tiunov, E. S., Tiunova, V., Ulanov, A. E., Lvovsky, A. & Fedorov, A. Experimental quantum homodyne tomography via machine learning. *Optica* **7**, 448–454 (2020).
6. Torlai, G. *et al.* Integrating neural networks with a quantum simulator for state reconstruction. *Phys. review letters* **123**, 230504 (2019).
7. Palmieri, A. M. *et al.* Experimental neural network enhanced quantum tomography. *npj Quantum Inf.* **6**, 1–5 (2020).
8. Torlai, G. *et al.* Neural-network quantum state tomography. *Nat. Phys.* **14**, 447–450 (2018).
9. Neugebauer, M. *et al.* Neural-network quantum state tomography in a two-qubit experiment. *Phys. Rev. A* **102**, 042604 (2020).
10. Lohani, S., Kirby, B., Brodsky, M., Danaci, O. & Glasser, R. T. Machine learning assisted quantum state estimation. *Mach. Learn. Sci. Technol.* (2020).
11. Xu, Q. & Xu, S. Neural network state estimation for full quantum state tomography. *arXiv preprint arXiv:1811.06654* (2018).
12. Lohani, S. *et al.* Improving application performance with biased distributions of quantum states. *arXiv preprint arXiv:2107.07642* (2021).
13. Ahmed, S., Muñoz, C. S., Nori, F. & Kockum, A. F. Quantum state tomography with conditional generative adversarial networks. *Phys. Rev. Lett.* **127**, 140502 (2021).
14. Ahmed, S., Muñoz, C. S., Nori, F. & Kockum, A. F. Classification and reconstruction of optical quantum states with deep neural networks. *Phys. Rev. Res.* **3**, 033278 (2021).
15. Lohani, S., Searles, T. A., Kirby, B. T. & Glasser, R. On the experimental feasibility of quantum state reconstruction via machine learning. *IEEE Transactions on Quantum Eng.* (2021).
16. Altepeter, J. B., Jeffrey, E. R. & Kwiat, P. G. Photonic state tomography. *Adv. At. Mol. Opt. Phys.* **52**, 105–159 (2005).

Machine learning with quantum fields

Gert Aarts,[1,2,*] Dimitrios Bachtis,[3,†] and Biagio Lucini[3,4,‡]

[1] *Department of Physics, Swansea University,*
Singleton Campus, SA2 8PP, Swansea, Wales, UK

[2] *European Centre for Theoretical Studies in Nuclear Physics and Related Areas (ECT*)*
& Fondazione Bruno Kessler Strada delle Tabarelle 286, 38123 Villazzano (TN), Italy

[3] *Department of Mathematics, Swansea University,*
Bay Campus, SA1 8EN, Swansea, Wales, UK

[4] *Swansea Academy of Advanced Computing, Swansea University,*
Bay Campus, SA1 8EN, Swansea, Wales, UK

The adoption of Euclidean quantum field theories in machine learning algorithms makes inference and learning possible using quantum field dynamics. We demonstrate that the ϕ^4 scalar field theory satisfies the Hammersley-Clifford theorem, therefore recasting it as a machine learning algorithm within the mathematically rigorous framework of Markov random fields. We illustrate the concepts by minimizing an asymmetric distance between the probability distribution of the ϕ^4 theory and that of target distributions, by quantifying the overlap of statistical ensembles between probability distributions and through reweighting to complex-valued actions with longer-range interactions. Neural networks architectures are additionally derived from the ϕ^4 theory which can be viewed as generalizations of conventional neural networks and applications are presented. Our aims are two-fold: the approach can provide a new perspective on machine learning with continuous degrees of freedom using the language of quantum fields, while also providing a new look at quantum fields when employed as building blocks in neural networks.

This presentation is based on Phys. Rev. D 103 (2021) 074510.

[*] g.aarts@swansea.ac.uk, presenter

[†] dimitrios.bachtis@swansea.ac.uk

[‡] b.lucini@swansea.ac.uk

Probably approximately correct quantum source coding

Armando Angrisani[1], Brian Coyle[2], Elham Kashefi[1,2]

[1] *Laboratoire d'Informatique de Paris 6, CNRS, Sorbonne Université, 4 place Jussieu, 75005 Paris, France*
[2] *School of Informatics, University of Edinburgh, EH8 9AB Edinburgh, United Kingdom*

Contact e-mail: armando.angrisani@lip6.fr

Abstract

Information-theoretic lower bounds are often encountered in several branches of computer science, including learning theory and cryptography. In the quantum setting, Holevo's and Nayak's bounds give an estimate of the amount of classical information that can be stored in a quantum state. Here we combine these information-theoretic tools with a counting argument, establishing the notion of *Probably Approximately Correct (PAC) Source Coding.* We remark that a similar approach was proposed in [1] in the context of distribution-free quantum PAC learning. Building upon this work, we show two novel applications in quantum learning theory and delegated quantum computation with a purely classical client. In particular, we give a lower bound on the sample complexity of a quantum learner for arbitrary functions under the Zipf's distribution, and we improve the security guarantees of the delegation protocol proposed in [2].

Extended Abstract

The term *source coding* refers to the process of encoding information produced by a given source in a way that it may be later decoded. The initial result of this topic, the source coding theorem was derived in the seminal work of Shannon [3], which describes how many bits are required to encode independent and identically distributed random variables, without loss of information. Nayak's bound [4] is a generalisation of the Holevo bound [5] which are fundamental results in the area of *quantum* source coding, particularly in the encoding and decoding of classical information encoded into quantum states. Informally, Nayaks bound states that the probability of successfully decoding an n bit string, encoded into $m < n$ qubits decreases exponentially in the difference between n and m. In this work, we generalise Nayak's bound, to not require that the encoded and decoded bits agree exactly, but only that the decoded bit string is only partially correct, which we quantify via the Hamming distance between the encoded and decoded bitstrings, X and Z respectively. We do this via some tools from *learning* theory, and specifically using the notion of *probably approximately correct* (PAC) learning. As a result, we dub our generalised bound as the PAC Nayak bound, which can be stated as follows:

Lemma 1 (PAC Nayak's bound)**.** *If X is an n-bit binary string, we send it using m qubits, and decode it via some mechanism back to an n-bit string Z , then our probability of correct decoding up to an error $\varepsilon n \geq 0$ in Hamming distance is given by*

$$\Pr[d_H(X, Z) \leq \varepsilon n] \leq \frac{2^m}{2^{n(1-\varepsilon-H(\varepsilon))}}.$$

Using this bound, we demonstrate two use cases in two apparently distinct application areas. The first is a use case in quantum learning theory, in which we prove a lower bound on the sample complexity of supervised learning algorithms relative to a specific distribution. The second is in the area of *delegated* quantum computation, where we use the bound to analyse and refine the security of the protocol of [2]. This protocol describes a method for a resource-limited 'client' to delegate a quantum computation to a powerful quantum 'server' in a manner to provide security guarantees to the data and information of the delegating client. This scenario is extremely relevant not only for the future quantum-enabled communication, but is also a practical concern in the modern day, where noisy-intermediate scale quantum (NISQ) [**?**] computers are only accessibly via a 'quantum cloud' [6]. To begin, we first prove the following lemma, which is a consequence of Holevo's bound:

Lemma 2 (Learning a string with quantum data)**.** *Assume X is an n-bit binary string sampled with probability p, we send it using m copies of a quantum state $\rho \in \mathbb{C}^{\ell \times \ell}$ and decode it via some mechanism back to an n-bit string Z. Let $d_H(X, Z) \leq \varepsilon n$ with probability $1 - \delta$. Then,*

$$m \geq \frac{(1-\delta)(1-H(\varepsilon))n - H(\delta)}{\log \ell}$$

Now, let us discuss briefly quantum learning theory, and how these above results can be useful there. In PAC learning, one considers a *concept class*, usually Boolean functions, $\mathcal{C} \subseteq \mathcal{F}_n := \{f | f : \{0,1\}^n \rightarrow \{0,1\}\}$. For a given concept, $c \in \mathcal{C}$, the goal of a PAC learner is to output a 'hypothesis', h, which is 'probably approximately' correct. In other words, the learner can output a hypothesis which agrees with the output of chosen function almost always. In order to be a PAC learner for a concept class, the learner must be able to do this for all $c \in \mathcal{C}$. In this process, learners are given access to an oracle, O, which (in one model) outputs a sample from a distribution, $x \sim D$, along with the corresponding concept evaluation at that datapoint, $c(x)$.

Regards *quantum* PAC learning, one question of key interest is whether quantum learners may be able to learn concepts with a lower sample complexity than is possible classically. Early results in this direction were both positive and negative, with the distribution from which the examples are sampled being a crucial ingredient. For example, it was shown [**?**] that exponential advantages for PAC learners were possible under the *uniform* distribution. On the other hand, if one requires the quantum PAC learner to succeed relative to *all* distributions, there is only a marginal improvement that quantum can hope to provide [1]. For our purposes, we analyse the sample complexity of quantum PAC learners, and use our tools to derive a lower bound on the learning problem, relative to another specific distribution (not simply the uniform one). The distribution in question is the *Zipf* distribution (which is a long-tailed distribution relevant in many practical scenarios, see [**?**, **?**, **?**] for example) over $\{0,1\}^n$, defined as follows:

$$\mathsf{Zipf}(k) := \frac{1}{kH_N}, \; H_K := \sum_{k=1}^{K} \frac{1}{k} \tag{1}$$

Taking Lemma 2, and combining it with Lemma 1, we can prove the following lower bound on the sample complexity of a PAC learner relative to the Zipf distribution.

Theorem 1. *For every $\varepsilon \in [0, 1/3)$ and $\delta \in [0,1]$, every $(\varepsilon, \delta, \mathsf{Zipf})$-PAC quantum learner for $\mathcal{F}_n$ has sample complexity $\Omega\left(N^{1-\varepsilon}/n\right)$.*

Now, moving to the second application; delegated quantum computation. Here we revisit the protocol of [2], which uses *flow ambiguity* to allow a client, who has no quantum resources, to delegate a computation blindly to a (potentially malicious) powerful quantum server. To do so, special properties of measurement-based quantum computation are utilised to gain security guarantees. By applying Lemma 1 to their protocol, we can prove the following theorem, which extends the results of [2] from the server being required to guess the *entire* computation, to only requiring them to gain knowledge of a part of it. This is important since in many realistic scenarios, an adversary leaking even part of a computation can be a serious threat to the client's privacy.

Theorem 2. *Let $\mathcal{C}$ be a family of computations, such that each $C \in \mathcal{C}$ can be encoded in a N-bit string. Assume that a client delegates $C \in \mathcal{C}$ to a server using $\lfloor 0.59N \rfloor$ bits of communication. Then the following bounds hold:*

1. *For every $\varepsilon \in [0,1]$, the server can guess C up to an error εN in Hamming distance with probability at most $2^{(-0.41+\varepsilon+H(\varepsilon))N}$. This gives an exponentially small probability for every $\varepsilon < 0.06$.*

2. *Assume that server outputs C' such that $d_H(C, C') \leq \varepsilon N$ with probability 1. Then $\varepsilon > 0.08$.*

Here, the family of computations, $\mathcal{C}$, are different possible *flows* through the MBQC path. The 'true' flow is only known to the client, and the hiding of this flow information is what gives the security guarantees. Given the two diverse applications we have demonstrated here, we believe the bounds we provide in this work may find applications in other areas of quantum information science.

References

[1] Srinivasan Arunachalam and Ronald de Wolf. Optimal Quantum Sample Complexity of Learning Algorithms. *Journal of Machine Learning Research*, 19(71):1–36, 2018.

[2] Atul Mantri, Tommaso F. Demarie, Nicolas C. Menicucci, and Joseph F. Fitzsimons. Flow Ambiguity: A Path Towards Classically Driven Blind Quantum Computation. *Phys. Rev. X*, 7(3):031004, July 2017.

[3] C. E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[4] A. Nayak. Optimal lower bounds for quantum automata and random access codes. In *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)*, pages 369–376, October 1999.

[5] Alexander Holevo. Bounds for the Quantity of Information Transmitted by a Quantum Communication Channel. *Problems of Information Transmission*, 9(3):177–183, 1973.

[6] Ryan LaRose. Overview and Comparison of Gate Level Quantum Software Platforms. *Quantum*, 3:130, March 2019.

Quantum statistical query model and local differential privacy

Armando Angrisani[1], Elham Kashefi[1,2]

[1] *Laboratoire d'Informatique de Paris 6, CNRS, Sorbonne Université, 4 place Jussieu, 75005 Paris, France*
[2] *School of Informatics, University of Edinburgh, EH8 9AB Edinburgh, United Kingdom*

Contact e-mail: armando.angrisani@lip6.fr

Abstract

The problem of private learning has been extensively studied in classical computer science. A striking equivalence between local differentially private algorithms and statistical query algorithms has been shown in [1]. In addition, the statistical query model has been recently extended to quantum computation in [2]. In this work, we give a formal definition of quantum local differential privacy and we extend the aforementioned result of [1], addressing an open problem posed in [3]. Our approach builds upon the seminal work of [4], which provided a notion of quantum differential privacy and showed a remarkable connection with quantum gentle measurements.

Extended Abstract

Differential privacy is a relatively recent field of computer science, that witnessed a tremendous growth over the last decade [5, 6]. Privacy issues arise in a number of computational tasks, notably in learning problems. In this context, differentially private mechanisms ensure that the final output does not depend too heavily on any one input or specific training example. In the standard model, a trusted curator collects the raw data of the individuals and it's responsible of their privacy. On the contrary, in the *local* model the curator is possibly malicious, and hence each individual submits her own privatized data. More formally, consider a statistical database, i.e. a vector $x = (x_1, \ldots, x_n)$ over a domain X, where each entry $x_i \in X$ represents information contributed by one individual. Databases x and x' are neighbors if $x_i \neq x'_i$ for exactly one $i \in [n]$. A randomized algorithm $\mathcal{A}$ is α-differentially private if for any two neighbor databases x, x' and for every subset F of the possible outcomes of $\mathcal{A}$ we have

$$\Pr[\mathcal{A}(z) \in F] \leq \exp(\alpha) \Pr[\mathcal{A}(z') \in F].$$

We now turn our attention to the *local model*. Following the notation used in [1], we say that a randomized algorithm over databases is α-local differentially private if it's an α-differentially private algorithm that takes in input a database of size $n = 1$.

An extension to quantum computation of the former notion is presented in the following sections. Moreover, we provide a precise characterization of *quantum* local private algorithms In particular, our result implies that concept class is learnable efficiently by a quantum local algorithm if and only if it is learnable efficiently in the quantum statistical query model.

Quantum statistical queries. We present here a definition of QSQ oracle that returns an approximation of the expectation value of any POVM. This definition generalizes the one given in [2], which is stated with respect to projective measurements. Recall that a POVM measurement M is given by a list of $d \times d$ positive semidefinite matrices $E_1, \ldots, E_k$, which satisfy $\sum_i E_i = \mathbb{I}$. The measurement rule is:

$$\Pr[M(\rho) \text{ returns outcome } i] = \mathrm{Tr}(E_i \rho)$$

and hence $\mathbb{E}[M(\rho)] = \sum_i i \mathrm{Tr}(E_i \rho)$.

Definition 1 (QSQ oracle)**.** *A quantum statistical query oracle $QSQ_\rho(\cdot, \cdot)$ for some d-dimensional mixed state ρ receives as inputs a tolerance parameter $\tau \geq 0$ and a POVM measurement $M = (E_0, \ldots, E_{k-1})$. Such oracle outputs a number α satisfying*

$$|\alpha - \mathbb{E}[M(\rho)]| \leq \tau.$$

A QSQ algorithm accesses a quantum state ρ via the quantum statistical query oracle QSQ_ρ. QSQ algorithms that prepare all their queries to QSQ_ρ before receiving any answers are called nonadaptive; otherwise, they are called adaptive.

Let $\mathcal{C} \subseteq \{c : \{0,1\}^n \to \{0,1\}\}$ be a concept class and $D : \{0,1\}^n \to [0,1]$ be a distribution. A case of particular interest is the one in which the state ρ is a quantum sample associated to a concept class, i.e. $\rho = |\psi_c\rangle \langle\psi_c|$, where $|\psi_c\rangle := \sum_{x\in\{0,1\}^n} \sqrt{D(x)}\,|x, c(x)\rangle$.

Quantum differential privacy. We recall the definition of quantum differential privacy given in [4]. For the sake of simplicity, we assume that the input state is a product state.

Definition 2 (DP measurement)**.** *Two product states $\rho = \rho_1 \otimes \ldots \otimes \rho_n$ and $\sigma = \sigma_1 \otimes \cdots \otimes \sigma_n$ are neighbors if there exists exactly one $i \in \{1, \ldots, n\}$ such that $\rho_i \neq \sigma_i$. A POVM measurement M is α-differentially private on some subset S of product states if for all states $\rho, \sigma \in S$ that are neighbors and every possible outcome y of M we have that*

$$\Pr[M(\rho) = y] \leq \exp(\alpha) \Pr[M(\sigma) = y].$$

Definition 3 (Trivial measurement)**.** *A measurement M is α-trivial on some subset S if for all states $\rho, \sigma \in S$ and every possible output y of M we have that*

$$\Pr[M(\rho) = y] \leq \exp(\alpha) \Pr[M(\sigma) = y].$$

We provide here a formal definition of quantum *local* differential privacy (LDP), inspired by the its classical countepart introduced in [1].

Definition 4 (QLDP oracle)**.** *Let $\rho = \rho_1 \otimes \cdots \otimes \rho_n$ be a product state. An α-quantum local differentially private (QLDP) oracle $QL_\rho(\cdot,\cdot)$ gets an index $i \in \{1, \ldots, n\}$ and an α-trivial measurement $M = E_1, \ldots, E_k$. Such oracle outputs $j \in [k]$ with probability $Tr(E_j\rho_i)$.*

Given a product state $\rho = \rho_1 \otimes \ldots \otimes \rho_n$, we say that an algorithm is α-QLDP if it accesses the state ρ via the oracle QL_ρ and the following restriction holds: for all $i \in \{1, \ldots, n\}$, if $QL_\rho(i, M_1), \ldots, QL_\rho(i, M_k)$ are the algorithm's invocations of QL_ρ on state ρ_i, where each M_j is an α_j-local differentially private measurement, then $\sum_{j=1}^k \alpha_j \leq \alpha$. QLDP algorithms that prepare all their queries to QL_ρ before receiving any answers are called noninteractive; otherwise, they are interactive.

The equivalence

In this section, we relate the QSQ model and the QLDP model. Specifically, we show that a QSQ algorithm that queries the oracle QSQ_ρ can be simulated by a QLDP algorithm that queries the oracle $QL_{\rho^{\otimes n}}$. Moreover, the expected query complexity is preserved up to polynomial factors. For the sake of simplicity, we restrict our analysis to noninteractive QLDP algorithms and nonadaptive QSQ algorithms.

Theorem 1 (Simulation of QSQ algorithms by QLDP algorithms)**.** *Let $\mathcal{A}$ be a nonadaptive QSQ algorithm that makes at most t queries to a QSQ oracle QSQ_ρ, each with tolerance at least τ, with respect to some POVMs with at most k outcomes. There exists a noninteractive $\ln\left(\frac{1+\alpha k}{1-\alpha}\right)$-QLDP algorithm that makes $n = t\frac{k^2\ln(2t/\beta)}{2\tau^2\alpha^2}$ queries to $QL_{\rho^{\otimes n}}$ and simulates $\mathcal{A}$ correctly with probability at least $1 - \beta$.*

Theorem 2 (Simulation of QLDP algorithms by QSQ algorithms)**.** *Let $\mathcal{A}$ be a noninteractive ε-QLDP algorithm that makes at most t queries to a QLDP oracle $QL_{\rho^{\otimes n}}$. There exists a nonadaptive QSQ algorithm $\mathcal{B}$ that makes in expectation at most $O(te^\varepsilon)$ queries to QSQ_ρ with tolerance $\tau := \beta/(3t)$ and the statistical difference between $\mathcal{B}$'s and $\mathcal{A}$'s output distributions is at most β.*

References

[1] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM J. Comput.*, 40(3):793–826, June 2011.

[2] Srinivasan Arunachalam, Alex B. Grilo, and Henry Yuen. Quantum statistical query learning, 2020.

[3] Srinivasan Arunachalam, Yihui Quek, and John Smolin. Private learning implies quantum stability, 2021.

[4] Scott Aaronson and Guy N. Rothblum. Gentle measurement of quantum states and differential privacy. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 322–333, New York, NY, USA, 2019. Association for Computing Machinery.

[5] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography*, TCC'06, page 265–284, Berlin, Heidelberg, 2006. Springer-Verlag.

[6] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. 9(3–4):211–407, August 2014.

Diagnostic Decision Support using Quantum Machine Learning

Amit Chakraborty[1] and Kameshwar Rao JV[2]
[1,2]HCL Technologies LTD
amitc8250@gmail.com, kameshjvkr@gmail.com

Abstract- Diagnostic Decision Support using Quantum Machine Learning (DDSQML) is an innovative solution used for detecting the diseases from Chest X-ray images. DDSQML architecture comprises of the quantum convolution layer and classical convolutional neural network. The DDSQML model has obtained a validation accuracy of 89.13% and a training accuracy of 98.78%.

I. DIAGNOSTIC DECISION SUPPORT USING QUANTUM MACHINE LEARNING

The recommended Diagnostic Decision Support using Quantum Machine Learning (DDSQML) model aims to enhance the performance of traditional CNNs classification for x-ray images and predict COVID-19 or other diseases. The central idea of the DDSQML model is predicated on quantum and classical hybrid computation [1] to enhance the performance of classical learning [2]. The DDSQML model is split into two parts: first, the quantum part has used the quantum Convolution layer, presented by Henderson et al. [3]. Second, a traditional convolutional neural network model for the ultimate classification of the model. The DDSQML model has implemented quantum convolution given by the Maxwell Henderson et al. [3]. The Quantum convolution operation [7] acts as random quantum circuits (RQCs) to compute the convolution operation and implement near-term quantum devices. Here circuit has been applied with local regions of input images to extract significant and informative features. The quantum convolution layer consists of three stages:

- Quantum encoding of the input file
- A random quantum circuit as quantum computation step
- Decoding the output of the random circuit (Measurement)

The encoding of the classical data to quantum state could be a hurdle in quantum machine learning (QML) [4]. Various encoding methods are discussed in [5]. We've used angle encoding to convert pixels' values of input images into rotation angles of quantum states. A quantum circuit could be a group of quantum unitary operations and measurements connected via wires just like the traditional convolutional layer, the quantum convolutional layer composed of quantum kernels applies to the input image. The central idea of quantum convolution operation is to utilize random quantum circuits to divide input images into small local regions to extract meaningful features. Measurements are taken from the circuit, and also the expectation values are decoded into classical bits. Here we are using the Pauli matrices as measurement methods [6]. By the quantum convolution step, we are able extract more complex features from the low dimension input images (28x28). After the quantum convolution, the info is fed into the classical convolution neural network(CNN). Convolution operation can extract feature maps from the input images by using the inner product between the chosen local region of the input image data and therefore the values of the kernel matrix. In this architecture, every convolution layer is followed by the relu activation function to feature the activation states within the network. Two Fully Connected Layer (FC) implemented after the convolution layer within the classical CNN. The FC layers perform the classification task following the flattening layer by implemented weights to predict classes. The output layer is applied after the fully connected (FC) layer with a softmax activation function with an adam optimizer with binary cross-entropy loss function.

We experimented with different types of algorithms for the same dataset; we used the SVM classifier as a classical machine learning algorithm and DenseNet121 architecture as a transfer learning. Training accuracy and validation accuracy are as follows.

Algorithm	Training Accuracy	Validation Accuracy
DenseNet121	99.86%	81.42%
SVM	85.95%	73.12%
DDSQML	98.78%	89.13%

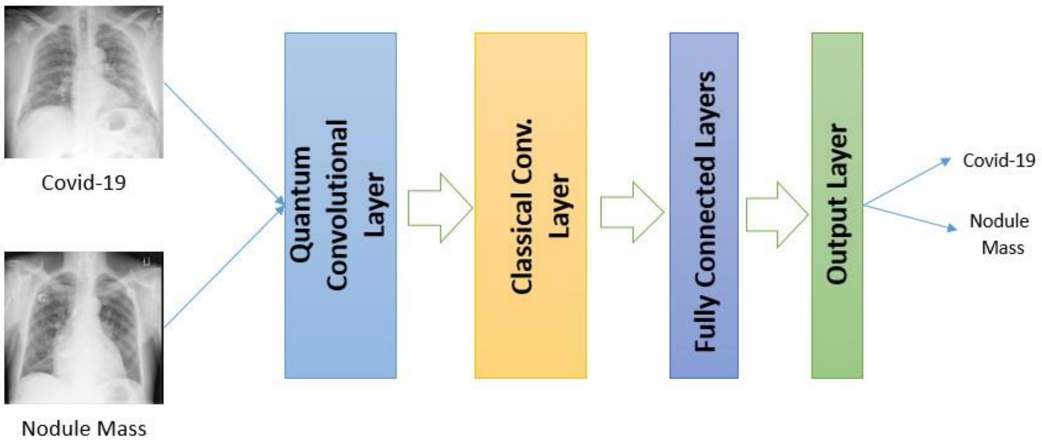


Fig.1 - Overall Workflow of the DDSQML

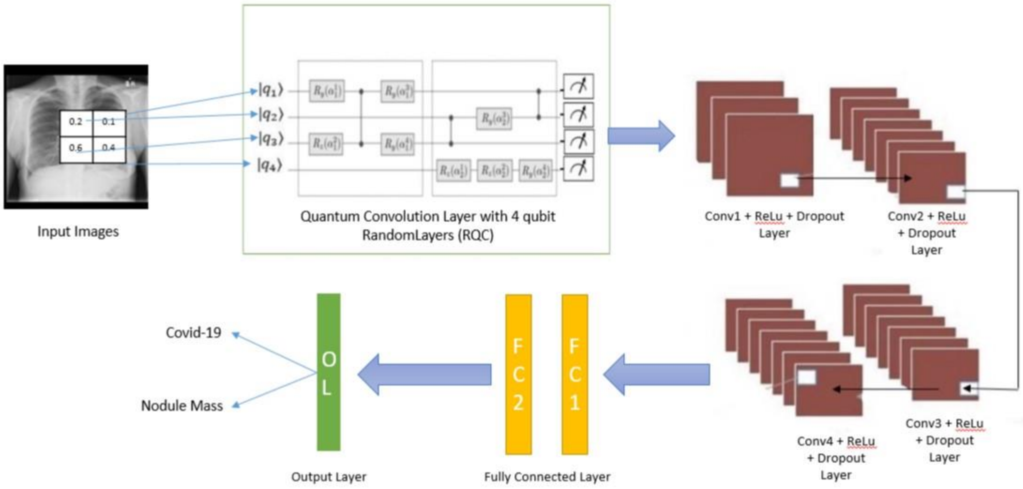

0.2
0.1
0.6
0.4
Input Images
$|q_1\rangle$
$|q_2\rangle$
$|q_3\rangle$
$|q_4\rangle$
Quantum Convolution Layer with 4 qubit RandomLayers (RQC)
Conv1 + ReLu + Dropout Layer
Conv2 + ReLu + Dropout Layer
Conv3 + ReLu + Dropout Layer
Conv4 + ReLu + Dropout Layer
F C 1
F C 2
Fully Connected Layer
O L
Output Layer
Covid-19
Nodule Mass

Fig 2 - Architecture of DDSQML model

REFERENCES

[1] Essam H. Houssein, Zainab Abohashima, Mohamed Elhoseny, Waleed M. Mohamed. Hybrid quantum convolutional neural networks model for COVID-19 prediction using chest X-Ray images. https://arxiv.org/abs/2102.06535v1

[2] Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, Carsten Blank, Keri McKiernan, and Nathan Killoran. Pennylane: Automatic differentiation of hybrid quantum-classical computations. arXiv preprint arXiv:1811.04968, 2018.

[3] M. Henderson, S. Shakya, S. Pradhan, and T. Cook. Quanvolutional neural networks: powering image recognition with quantum circuits. Quantum Machine Intelligence, 2:1–9, 2020.

[4] Yuki Takeuchi, Tomoyuki Morimae, and Masahito Hayashi. Quantum computational universality of hypergraph states with pauli-x and z basis measurements. Scientific reports, 9(1):1–14, 2019.

[5] Seth Lloyd, Maria Schuld, Aroosa Ijaz, Josh Izaac, and Nathan Killoran. Quantum embeddings for machine learning. arXiv preprint arXiv:2001.03622, 2020

[6] Yuki Takeuchi, Tomoyuki Morimae, and Masahito Hayashi. Quantum computational universality of hypergraph states with pauli-x and z basis measurements. Scientific reports, 9(1):1–14, 2019.

[7] Quanvolutional Neural Networks — PennyLane

Topological quantum phase transitions retrieved through unsupervised machine learning

Authors:

Yanming Che (1, 2), Clemens Gneiting (1), Tao Liu (1), Franco Nori (1, 2)

1. Theoretical Quantum Physics Laboratory, RIKEN
2. Department of Physics, University of Michigan

Abstract：

Detecting topological features in physics with unsupervised machine learning has been attracting increasing attention recently. It provides a viable approach for the study of topological phase transitions, without prior knowledge and labeled training examples of the system.

Here we show with several prototypical and relevant models that topological quantum phase transitions can be automatically retrieved with unsupervised manifold learning, requiring only a very limited number of hyperparameters. Inspired by the fidelity-susceptibility indicator for topological quantum phase transitions as well as the non-Euclidean structure of the data set, we argue that the widely used choice of a Euclidean distance can be in general suboptimal to discover topological transitions in momentum and real space. On the other hand, we can show that The Chebyshev distance (for momentum space data) and the (dual) L1-norm distance (for real space data) sharpen the topological quantum transitions, which are better (approximative) distance metrics supporting the retrieval of the critical points through nonlinear dimensionality reductions. The pursued method has the potential to deepen our understanding of topological features in quantum systems, providing fresh perspectives and possible applications for studying topological quantum phase transitions, especially when no additional prior information is available.

Reference: Y. Che, C. Gneiting, T. Liu, F. Nori, Phys. Rev. B 102, 134213 (2020).

URL: https://journals.aps.org/prb/abstract/10.1103/PhysRevB.102.134213

Quantum Discriminator with Gaussian Smoothing for Binary Classification

Wyatt Smith, Undergraduate Researcher
University of Tennessee, Knoxville
Knoxville, TN, 37919
Email: wsmith97@vols.utk.edu

Prasanna Date, Research Scientist
Oak Ridge National Laboratory
Oak Ridge, TN, 37830
Email: datepa@ornl.gov

Abstract: Quantum-classical hybrid algorithms have been shown to greatly speed up various machine learning tasks. In this work, we implement a novel, hybrid algorithm for binary classification, the Quantum Discriminator, on a 3x3 bars and stripes dataset and explore means of increasing the algorithm's generalizability via a process of Gaussian smoothing.

Keywords: Quantum Learning; Quantum Machine Learning; Binary Classification.

Machine learning models have been widely used for numerous scientific, business and consumer applications in the recent past with great success [1]. Presently, machine learning models are run on classical computing platforms containing CPUs, GPUs or FPGAs and trained using large amounts of data. However, this will not be sustainable in the future. In the future, while data available for training is expected to increase significantly and rapidly, the classical computing platforms are expected to stagnate in terms of speed and computing power owing to the inevitable end of Moore's law [2]. This will severely restrict our ability to build end-to-end machine learning applications. In order to continue developing machine learning applications in the post-Moore's law era, we must resort to non-conventional computing platforms like quantum computing and neuromorphic computing [3, 4].

Advances in quantum computing have opened up a new frontier in which existing machine learning techniques can be greatly improved and entirely new algorithms can be developed [5, 6]. In our previous work, we have developed adiabatic quantum computing approaches for widely used machine learning models such as linear regression, support vector machine, and k-means clustering [7, 8, 9]. In this work, we turn our attention to universal quantum computers and propose a novel quantum machine learning model called the Quantum Discriminator.

The quantum discriminator is a technique designed to perform the task of binary classification— a problem whereby a model is tasked with sorting data into two classes (often denoted Class 0 and Class 1). The quantum discriminator is designed to perform binary classification on binary data in particular. It consists of a training step, which follows a quantum-classical hybrid algorithm, and an inferencing step, which is performed on a universal quantum computer. For a data set of size N, training the quantum discriminator requires $\mathcal{O}(N \log N)$ classical bits, $\mathcal{O}(\log N)$ quantum bits, and can be done in $\mathcal{O}(N \log N)$ time. Training yields a list of $\mathcal{O}(N)$ model parameters which are then used to construct the inferencing circuit, requiring $\mathcal{O}(\log N)$ quantum bits [10].

The quantum discriminator had yet to be implemented and tested on any real-world data set; however, it is in theory highly prone to overfitting and can generalize quite poorly on complex datasets depending the feature extraction process. The discriminator should inherently perform quite well on the data seen during training, with the capability to obtain 100% accuracy on a separable training set [10]. Despite this inherent accuracy, discrepancies between the size of the training set and that of our binary feature space can lead to poor performance on data reserved for validation. In particular, the discriminator fails to identify Class 1 data in regions of our feature space that are not sampled in training; meaning our model should suffer from poor recall when the size of our feature space greatly exceeds $\log N$.

In this work, we implement the quantum discriminator for the purpose of benchmarking and propose a technique which, in principle, should improve the generalizability issues of our discriminant model. Specifically, we introduce a new step to the training phase of the discriminator technique whereby we smooth our extracted model parameters in a Gaussian fashion. During the smoothing process, we introduce a hyperparameter, σ, that controls the standard deviation of our Gaussian peaks. Both the unaltered and smoothed discriminator algorithms were benchmarked on a bars and stripes data set, containing 50 bars and 50 stripes in a 3 by 3 grid. Inferencing was performed both on IBM Q quantum computers and via noiseless simulation using Qiskit; various performance metrics such as accuracy, precision, and recall were gathered in each case.

The results of our experiment indicate that the Quantum Discriminator fares extremely well on conventional, $80/20$ train/test splits of our data, obtaining an average validation accuracy of 99.15% across 300 trials in simulation. A histogram displaying the distribution of the discriminator's accuracy on validation in this case can be found in Fig. 1. Moreover, it was found that the Gaussian smoothing procedure allowed the discriminator to locate Class 1 data in cases where the training data set does not adequately sample our relevant feature space. For example, in the case of an $11/89$ train/test split, it was found that the discriminator obtained an average validation accuracy of 71.02% before smoothing, whereas the smoothed models obtained average validation accuracies nearing 76% with an appropriate choice of smoothing parameter, σ. A histogram depicting the distribution of model accuracy for the discriminator without smoothing on an $11/89$ split can also be found in Fig. 1, and charts comparing the performance of smoothed and unsmoothed models on this same split can be found in Fig 2.

Figure 1. Histograms depicting the validation accuracies for unsmoothed models using a conventional $80/20$ split (left) and a sparse, $11/89$ split (right):

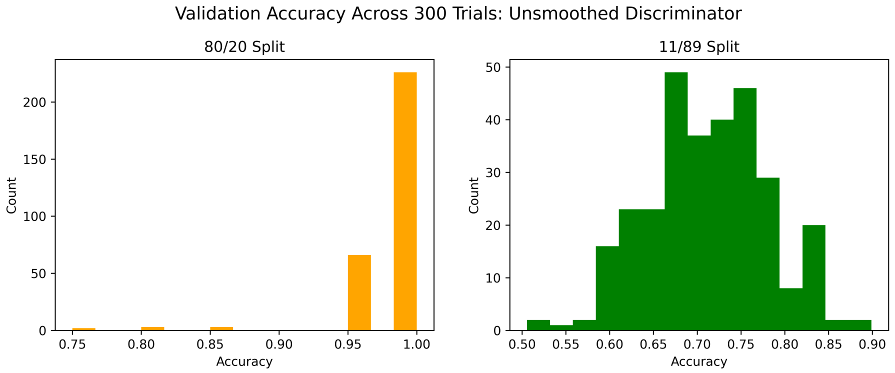
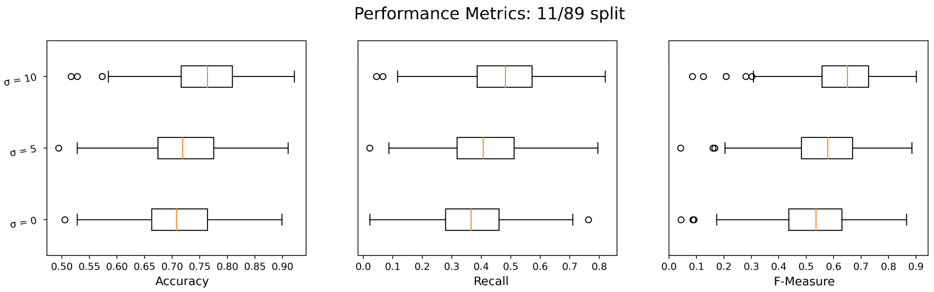


Figure 2. Box and whisker plots displaying the distribution of model accuracy, recall, and F-measure on an $11/89$ split for different values of smoothing parameter:



References:

[1] Junfei Qiu, Qihui Wu, Guoru Ding, Yuhua Xu, and Shuo Feng. A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, 2016(1):67, 2016.

[2] James B Aimone. Neural algorithms and computing beyond moore's law. *Communications of the ACM*, 62(4):110–110, 2019.

[3] Prasanna Date, Robert Patton, Catherine Schuman, and Thomas Potok. Efficiently embedding qubo problems on adiabatic quantum computers. *Quantum Information Processing*, 18(4):1–31, 2019.

[4] Prasanna Date, Catherine Schuman, Bill Kay, Thomas Potok, et al. Neuromorphic computing is turing-complete. *arXiv preprint arXiv:2104.13983*, 2021.

[5] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.

[6] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. An introduction to quantum machine learning. *Contemporary Physics*, 56(2):172–185, 2015.

[7] Prasanna Date, Davis Arthur, and Lauren Pusey-Nazzaro. Qubo formulations for training machine learning models. *Scientific Reports*, 11(1):1–10, 2021.

[8] Prasanna Date and Thomas Potok. Adiabatic quantum linear regression. *arXiv preprint arXiv:2008.02355*, 2020.

[9] Davis Arthur and Prasanna Date. Balanced k-means clustering on an adiabatic quantum computer. *arXiv preprint arXiv:2008.04419*, 2020.

[10] Prasanna Date. Quantum discriminator for binary classification.

Machine learning and optical quantum information

K. Bartkiewicz[1,2]

🖂 bark@amu.edu.pl

[1] Institute of Spintronics and Quantum Information, Faculty of Physics, Adam Mickiewicz University in Poznań, Uniwersytetu Poznańskiego 2, 61-614 Poznań, Poland

[2] RCPTM, Joint Laboratory of Optics of Palacký University and Institute of Physics of Czech Academy of Sciences,17. listopadu 12, 771 46 Olomouc, Czech Republic

Classical programming means writing explicit instructions so that a program processes the input data and correctly answers our questions. Machine learning (ML) is a branch of artificial intelligence research that uses implicit programming, where the program does not receive explicit instructions. This method is particularly suitable for problems that are intuitive to humans but difficult to convert to set of machine instructions. Some complex problems resist known ML methods, especially in quantum systems [1,3]. E.g. designing new drug molecules or supervising quantum communication networks, which under certain assumptions should be protected from eavesdropping by the laws of quantum physics. These tasks quickly become unfeasible as the complexity of the problem increases. Solutions to such problems must be sought using quantum computing for ML [1,2]. This is the original motivation to combine ML and quantum physics [1]. However, there are many other reasons to do so. In particular, ML can be used to motivate theoretical and experimental research in quantum information, quantum state engineering, classification and detection. In some cases surpassing the best human-designed solutions. To illustrate this, I will discuss a few assorted examples of combining ML and photonic quantum information processing, including [2,4-7]. These results include kernel and Hilbert-Schmidt distance based methods, supervised and unsupervised learning.

Short Abstract

Quantum ML can be used to motivate theoretical and experimental research in quantum information, quantum state engineering, classification and detection. I will discuss a few assorted examples of combining ML and photonic quantum information processing, including [2,4-7]. These results include kernel and Hilbert-Schmidt distance based methods, supervised and unsupervised learning. The also covey examples where human-designed classification methods for quantum states are surpassed by our ANNs.

Keywords

optical quantum information, quantum machine learning, kernel trick, classification, clustering, Hilbert-Schmidt distance, variational quantum circuits, entanglement detection

References

[1] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, S. Lloyd, Nature **549**, 195 (2017).
[2] V. Trávníček, K. Bartkiewicz, A. Černoch, K. Lemr, Phys. Rev. Lett. **123**, 260501 (2019).
[3] G. Carleo et al., Rev. Mod. Phys. **91**, 045002 (2019).
[4] K. Bartkiewicz, C. Gneiting, A. Černoch, K. Jiráková, K. Lemr, F. Nori, Sci. Rep. **10**, 12356 (2020).
[5] J. Jašek, K. Jiráková, K. Bartkiewicz, A. Černoch, T. Fürst, K. Lemr, Opt. Express **27**, 32454 (2019).
[6] V. Trávníček, K. Bartkiewicz, A. Černoch, K. Lemr, Phys. Rev. Applied, **14**, 064071, (2020).
[7] J. Roik, K. Bartkiewicz, A. Černoch, K. Lemr, Phys. Rev. Applied, **15**, 054006, (2021)

Optimisation-free Classification with Quantum Circuit Measurements

Extended Abstract Submitted to QTML2021

Vladimir Vargas-Calderón[1], **Fabio A. González**[2], and **Herbert Vinck-Posada**[1]

[1]Grupo de Superconductividad y Nanotecnología, Departamento de Física, Universidad Nacional de Colombia, Bogotá, Colombia

[2]MindLab Research Group, Departamento de Ingeniería de Sistemas e Industrial, Universidad Nacional de Colombia, Bogotá, Colombia

Abstract

We present a novel supervised machine learning method based on preparing quantum states, through quantum feature maps (QFMs), and on making measurements on those states, corresponding to the training and prediction phases, respectively [1] . Remarkably, our proposal does not require learning parameters. The method is implemented on real quantum devices.

Introduction.—In recent years the area of quantum machine learning has been gaining popularity amongst researchers because of the intrinsic connections between quantum physics and machine learning [2, 3]. There are different approaches to quantum machine learning depending on whether a classical or quantum system generates the data, and whether the data processing device is a classical or a quantum computer [4]. Some quantum versions of different classical machine learning algorithms have been implemented with an emphasis on showing an advantage, at least theoretically, of the quantum version in terms of speedup [5, 6, 7]. We focus on another research branch which has been less explored, where quantum information tools are used to formulate machine learning methods that take advantage of the quantum conceptual machinery.

Classification tasks have wide academic and industry applications. Several algorithms have been devised to utilise quantum computers for classification [8, 9, 10]. The task consists of learning a function $f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{Y}$ that maps a feature vector $\boldsymbol{x}$ to an estimated class or label $\hat{y}$, where $\hat{y} = f_{\boldsymbol{\theta}}(\boldsymbol{x}) \in \mathcal{Y} = \{1, 2, \ldots, K\}$ is one of K possible labels to classify the feature vector into. The subscript $\boldsymbol{\theta}$, indicates that there is a set of parameters that define a family of functions – parameterised by $\boldsymbol{\theta}$. The learning process is usually achieved by collecting a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, where $y_i \in \mathcal{Y}$ are called the true label associated to the feature vector $\boldsymbol{x}_i$. Then, through some algorithm, learning occurs by minimising an error function, e.g. $\min_{\boldsymbol{\theta}} E = \min_{\boldsymbol{\theta}} \sum_{i=1}^N d(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i)$, where d is some distance function.

This error minimisation strategy has been considered by most of the literature, both in classical and, more recently, quantum machine learning [11]. However, there are non-optimisation-based strategies that estimate (either explicitly or implicitly), the joint probability distribution of the data $P(\mathcal{D}) = P(X, Y)$. In this work, we report the implementation of an optimisation-free classification algorithm on quantum circuits.

The method.—The classification algorithm consists of three basic steps [1]: *i)* computing a QFM for each data sample ($\boldsymbol{x} \mapsto |\psi_{\mathcal{X}}(\boldsymbol{x})\rangle$ and $y \mapsto |\psi_{\mathcal{Y}}(y)\rangle$, so that each data sample $(\boldsymbol{x}_i, y_i)$ is mapped to $|\psi_i\rangle = |\psi_{\mathcal{X}}(\boldsymbol{x}_i)\rangle \otimes |\psi_{\mathcal{Y}}(y_i)\rangle$) in order to build a training dataset state $|\psi_{\text{train}}\rangle$ by considering the superposition of all data samples, i.e. $|\psi_{\text{train}}\rangle = \sum_i |\psi_i\rangle / \|\sum_i |\psi_i\rangle\|$ *ii)* building the quantum state for a new sample $\boldsymbol{x}_\star$ to be classified using the QFM, i.e. $|\psi_\star\rangle =$

$|\psi_{\mathcal{X}}(\boldsymbol{x}_\star)\rangle$, *iii)* projecting the dataset quantum state onto the new sample quantum state, so that the density matrix corresponding to the $\mathcal{Y}$ subsystem– $\rho'_{\mathcal{Y}} = \mathrm{Tr}_{\mathcal{X}}[\rho']$, where $\rho' = \pi_\star |\psi_{\text{train}}\rangle\langle\psi_{\text{train}}| \pi_\star / \mathrm{Tr}[\pi_\star |\psi_{\text{train}}\rangle\langle\psi_{\text{train}}| \pi_\star]$, with $\pi_\star = |\psi_\star\rangle\langle\psi_\star| \otimes \mathrm{Id}_{\mathcal{Y}}$– contains the probabilities of classifying the new sample into each label. We highlight that calculating the training dataset state $|\psi_{\text{train}}\rangle$ does not require parameter optimisation and the computational cost is linear on the size of the training dataset.

The circuit.—We have successfully applied the previous method both to toy data sets [1], as well as more challenging data sets [12], without the need to build quantum circuits. However, the method can also be realised on qubit-based quantum devices[1], as we show in this work by implementing it on the IBM Bogotá quantum computer.

Let U_d be the circuit that prepares the dataset state $|\psi_{\text{train}}\rangle$ with some given QFM. Such dataset state has n_x qubits corresponding to the $\mathcal{X}$ subsystem, and n_y qubits corresponding to the $\mathcal{Y}$ subsystem[2]. Thus, U_d is a quantum circuit acting on $n_x + n_y$ qubits. On the other hand, let $U_\star$ be the circuit that prepares the state for the sample to be classified, so that it only acts on the first n_x qubits. Classification is achieved by estimating $\left|\langle 0^{n_x}| \otimes \langle 0^{n_y}| U_d(\mathcal{D}) U_\star^\dagger(\boldsymbol{x}_\star) |0^{n_x}\rangle \otimes |q_{\mathcal{Y}}\rangle\right|^2$, which is a circuit form normally used to estimate quantum kernels [14]. Thus, the probability of obtaining a label y for the sample $\boldsymbol{x}_\star$ is computed by setting $|q_{\mathcal{Y}}\rangle$ to $|\psi_{\mathcal{Y}}(y)\rangle$.

The circuits are built by first computing the state classically. Then, such a state can be deterministically prepared with algorithms such as the one presented in Ref. [15]. Note that for QFMs that require many qubits, computing the state classically can be intractable; thus, such QFMs are not usable with our framework. Nonetheless, we can approximate any QFM with random Fourier features, as shown in Ref. [12].

Results.—We validated our proposal with a toy XOR data set, which is a set of 4 blobs of points in 2D in a chequerboard pattern, as shown in fig. 1 (only the test data set is shown). Each coordinate of each sample is mapped to a qubit using the QFM $|\psi_{\mathcal{X}}(x_1, x_2)\rangle = \bigotimes_{i=1}^{2} (\sin \pi x_i |0\rangle + \cos \pi x_i |1\rangle)$, whereas its corresponding label is mapped to $|0\rangle$ for red points and $|1\rangle$ for blue points. The background of the panels in fig. 1 shows the probability assigned by the exactly simulated circuit, the noisy simulated circuit, and the real circuit, from left to right. IBM reports that the noisy simulation is based on one- and two-qubit gate errors consisting of depolarising and thermal relaxation errors, as well as individual qubit readout errors. The large discrepancy between simulated and real circuit predictions (see ROC-AUC metrics in fig. 1 caption) show that a more complete understanding and simulation of noise is needed.

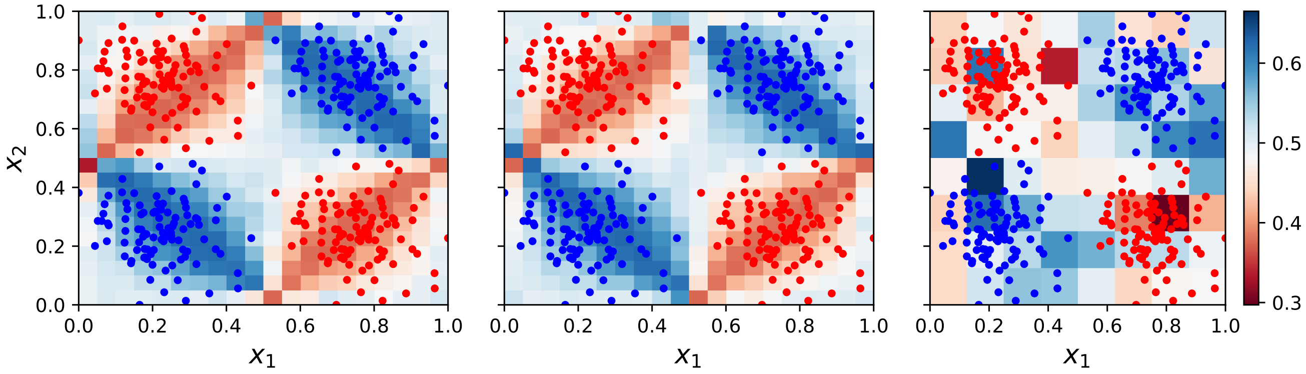


Figure 1: Predictions (background colour) of exact circuit simulation (left), noisy circuit simulation (middle) and circuit on the IBM Bogotá quantum device (right) for a XOR data set (points). The area under the the receiver operating characteristic curve was 99.93%, 99.82% and 95.83% for the predictions of the exact simulation, noisy simulation, and real quantum device, respectively. The prediction circuit consists of three qubits.

[1] We have also demonstrated this on simulated qudit-based quantum devices [13].

[2] Labels are encoded into the n_y qubits, for instance, with a one-hot encoding.

References

[1] Fabio A González, Vladimir Vargas-Calderón, and Herbert Vinck-Posada. Classification with quantum measurements. *Journal of the Physical Society of Japan*, 90(4):044002, 2021.

[2] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4): 045002, 2019.

[3] Juan Carrasquilla. Machine learning for quantum matter. *Advances in Physics: X*, 5(1):1797528, 2020.

[4] Maria Schuld and Francesco Petruccione. *Supervised Learning with Quantum Computers*. Quantum Science and Technology. Springer International Publishing, Cham, 2018. ISBN 978-3-319-96423-2. doi: 10.1007/978-3-319-96424-9. URL `http://link.springer.com/10.1007/978-3-319-96424-9`.

[5] Diego Ristè, Marcus P Da Silva, Colm A Ryan, Andrew W Cross, Antonio D Córcoles, John A Smolin, Jay M Gambetta, Jerry M Chow, and Blake R Johnson. Demonstration of quantum advantage in machine learning. *npj Quantum Information*, 3(1):1–5, 2017.

[6] Brian Coyle, Daniel Mills, Vincent Danos, and Elham Kashefi. The born supremacy: quantum advantage and training of an ising born machine. *npj Quantum Information*, 6(1):1–11, 2020.

[7] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Information-theoretic bounds on quantum advantage in machine learning. *Physical Review Letters*, 126(19):190505, 2021.

[8] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Physical review letters*, 113(13):130503, 2014.

[9] Maria Schuld, Alex Bocharov, Krysta M. Svore, and Nathan Wiebe. Circuit-centric quantum classifiers. *Phys. Rev. A*, 101:032308, Mar 2020. doi: 10.1103/PhysRevA.101.032308. URL `https://link.aps.org/doi/10.1103/PhysRevA.101.032308`.

[10] Amira Abbas, Maria Schuld, and Francesco Petruccione. On quantum ensembles of quantum classifiers. *Quantum Machine Intelligence*, 2(1):1–8, 2020.

[11] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.

[12] Fabio A. González, Alejandro Gallego, Santiago Toledo-Cortés, and Vladimir Vargas-Calderón. Learning with density matrices and random features. *CoRR*, abs/2102.04394, 2021. URL `https://arxiv.org/abs/2102.04394`.

[13] Diego H Useche, Andres Giraldo-Carvajal, Hernan M Zuluaga-Bucheli, Jose A Jaramillo-Villegas, and Fabio A González. Quantum measurement classification with qudits. *arXiv preprint arXiv:2107.09781*, 2021.

[14] Yunchao Liu, Srinivasan Arunachalam, and Kristan Temme. A rigorous and robust quantum speed-up in supervised machine learning. *Nature Physics*, pages 1–5, 2021.

[15] Vivek V Shende, Stephen S Bullock, and Igor L Markov. Synthesis of quantum-logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(6):1000–1010, 2006.

Effects of Different Encodings and Distance Functions on Quantum Instance-based Classifiers

Alessandro Berti, Anna Bernasconi, Gianna M. Del Corso, and Riccardo Guidotti

Dipartimento di Informatica, Università di Pisa, Italy

Abstract

We illustrate and compare alternatives for the quantum nearest neighbor classifier focusing on data preparation and performance. We discuss the differences in the classification process depending on data encoding, storage, and distance functions. The results show that the quantum nearest neighbor approach compares well with the classic version.

Extended Abstract

Quantum Machine Learning (QML) summarizes approaches that use synergies between machine learning and quantum computing (QC) [1–3]. Among different QML approaches, we focus on those using QC to process classical datasets [4]. More in detail, the dataset under analysis consists of *classical* records, such as images or relational data, which are then used as input for the *quantum* algorithms. In this setting, it is required a classical-quantum "interface" which is realized through ad-hoc data transformations procedures. These procedures are designed according to the task that we want to solve with QML and they can be different depending on the structure of the quantum circuits of the QML algorithm employed. One of the contributions of this work is to study and quantify how different data preparation procedures may affect the performance of QML algorithms.

A central task in QML is to understand whether there are advantages in the quantum counterpart of classical ML algorithms [5]. Several ML algorithms have been already "translated" in different ways. Examples are the K-Means clustering algorithm [2], Principle Component Analysis [6], Support Vector Machine [7], K-Nearest Neighbor [8], Neural Network [9], etc. We focus here on the Quantum K-Nearest Neighbor (QKNN) algorithm. The theoretical advantage of QKNN with respect to KNN is that QKNN can calculate the distances between the test instance and the records in the training set at the same time. In the literature, we can find different versions of QKNN [8,10–16] implementing different distance functions and requiring different data encoding. However, it is not simple to compare these results to understand which are the pros and cons of the different approaches.

We describe the quantum circuits implementing two different versions of QKNN that adopt different distance functions: Euclidean [10] and Hamming distances [15]. We delineate the circuits responsible for the calculus of the distances and those responsible for storing the data.

Since different quantum distance functions require different quantum data storage, encoding, and pre-processing, we analyze all these aspects in detail by also examining the complexity of the quantum circuits. In particular, we compare the QKNN based on amplitude encoding through FF-QRAM [17] and the QKNN based on basis encoding with storage based on the branching technique described in [18]. In addition to the performance of a quantum classifier, it is also important to take into account the number of qubits required to implement different QKNNs. Hence, we empirically experiment the different QKNN methods by varying the number of training records and the number of features adopted to assess how sensitive the classification procedures are in terms of accuracy.

Our analysis illustrates the theoretical and empirical differences on various datasets and compares them with a classic KNN algorithm. In particular, we ran experiments on three open-source datasets: iris, cancer, and mnist. Experiments for QKNN were implemented in Qiskit and run both on a quantum simulator[1] and (when possible) on the quantum computer offered by IBM Quantum Experience[2]. The link to the source code will be made available in the full paper. When possible, we experimented with the complete number of features. However, due to limited computational resources, sometimes we needed to train the QKNN algorithms on a preprocessed dataset with a reduced number of features, applying a Principal Component Analysis (PCA) [19]. For the binary encoding we adopted two different strategies to encode inputs: *Recursive Minimal Entropy Partitioning* (RMEP) method [20] and *Locality-Sensitive Hashing* (LSH) [21]. The results show that, with appropriate data encoding and training strategies, QKNN is comparable or, in some circumstances, even better than the classic KNN. The experiments highlight that one of the greater challenges in the usage of QKNN lies in the data preparation and its encoding.

Several future research directions are possible. First, we would like to study the impact of running QKNN with different values of the number k of nearest neighbors, in both amplitude and basis encoding. Second, we want to test QKNN with additional datasets. Finally, we would like to perform a similar analysis on other QML algorithms such as SVM, or Neural Networks.

References

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information (10th Anniversary edition).* Cambridge University Press, 2016.

[2] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum algorithms for supervised and unsupervised machine learning," *arXiv preprint arXiv:1307.0411*, 2013.

[3] M. Schuld, I. Sinayskiy, and F. Petruccione, "An introduction to quantum machine learning," *Contemporary Physics*, vol. 56, no. 2, pp. 172–185, 2015.

[4] M. Schuld and F. Petruccione, *Supervised learning with quantum computers.* Springer, 2018.

[5] Z. Abohashima, M. Elhoseny, E. H. Houssein, and W. M. Mohamed, "Classification with quantum machine learning: A survey," *CoRR*, vol. abs/2006.12270, 2020.

[1] Experiments were run on Linux CentOS, Intel(R) Xeon(R) Gold 6132 CPU @ 2.60GHz with an 376 GB RAM

[2] IBM Quantum: `https://quantum-computing.ibm.com/`

[6] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum principal component analysis," *Nature Physics*, vol. 10, no. 9, pp. 631–633, 2014.

[7] P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum support vector machine for big data classification," *Physical review letters*, vol. 113, no. 13, p. 130503, 2014.

[8] M. Schuld, I. Sinayskiy, and F. Petruccione, "Quantum computing for pattern classification," in *PRICAI*, ser. LNCS, vol. 8862. Springer, 2014, pp. 208–220.

[9] ——, "Simulating a perceptron on a quantum computer," *Physics Letters A*, vol. 379, no. 7, pp. 660–663, 2015.

[10] M. Schuld, M. Fingerhuth, and F. Petruccione, "Implementing a distance-based classifier with a quantum interference circuit," *EPL (Europhysics Let.)*, vol. 119, no. 6, p. 60002, 2017.

[11] Y. Ruan, X. Xue, H. Liu, J. Tan, and X. Li, "Quantum algorithm for k-nearest neighbors classification based on the metric of hamming distance," *International Journal of Theoretical Physics*, vol. 56, no. 11, pp. 3496–3507, 2017.

[12] N. Wiebe, A. Kapoor, and K. M. Svore, "Quantum nearest-neighbor algorithms for machine learning," *Quantum information and computation*, vol. 15, 2018.

[13] Y. Dang, N. Jiang, H. Hu, Z. Ji, and W. Zhang, "Image classification based on quantum k-nearest-neighbor algorithm," *Quantum Inf. Proc.*, vol. 17, no. 9, p. 239, 2018.

[14] Afham, A. Basheer, S. K. Goyal *et al.*, "Quantum k-nearest neighbor machine learning algorithm," *arXiv preprint arXiv:2003.09187*, 2020.

[15] J. Li, S. Lin, Y. Kai, and G. Guo, "Quantum k-nearest neighbor classification algorithm based on hamming distance," *arXiv preprint arXiv:2103.04253*, 2021.

[16] R. Mengoni, M. Incudini, and A. D. Pierro, "Facial expression recognition on a quantum computer," *Quantum Mach. Intell.*, vol. 3, no. 1, pp. 1–11, 2021.

[17] D. K. Park, F. Petruccione, and J.-K. K. Rhee, "Circuit-based quantum random access memory for classical data," *Scientific reports*, vol. 9, no. 1, pp. 1–8, 2019.

[18] D. Ventura and T. R. Martinez, "Quantum associative memory," *Inf. Sci.*, vol. 124, no. 1-4, pp. 273–296, 2000.

[19] P. Tan, M. S. Steinbach, and V. Kumar, *Introduction to Data Mining.* Addison-Wesley, 2005.

[20] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and unsupervised discretization of continuous features," in *ICML*, 1995, pp. 194–202.

[21] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of Massive Datasets, 2nd Ed.* Cambridge University Press, 2014.

A quantum image edge detector for the NISQ era

Alexander Geng, Ali Moghiseh, Claudia Redenbach, Katja Schladitz

Abstract—The goal of edge detection is to find ridges in the gray value relief of an image. That is, we are looking for image locations where the gray value intensity changes suddenly. Edges are among the most important components to understand and segment an image. Humans detect edges routinely visually. In digital image processing, edge detection is a standard task, too. In industrial applications, edge detection is used e.g., for extracting the structure of objects, features, or regions within an image. Thereby, changes in material properties like surface defects can be detected.

In classical image processing, a standard way to highlight the edges is the application of the image gradient. This requires the pixel-wise computation of the partial gray value derivatives which is done by convolving the image with a filter mask weighting a chosen discrete neighborhood. In the filtered image, high gray values indicate a gray value change in the original, whereas low gray values indicate homogeneous neighborhoods without changes and edges. Various methods for edge detection have been suggested, for instance the Prewitt, Sobel or Laplace filters or Canny's edge detector [1].

In contrast to classical methods, where each pixel must be considered, quantum edge detection algorithms promise exponential speedup. Exploiting a real quantum computer, we can also benefit from exponentially lower memory usage in terms of number of qubits compared to the number of bits needed to represent an image in the classical way. Several approaches for quantum edge detection have been proposed. However, most of them are only formulated in theory or for a quantum computer simulator [2-6]. For example, in QSobel [6] - a quantum version of the well-known classical Sobel filter, some steps like the COPY operation or the quantum black box for calculating the gradients of all pixels can currently not be implemented. To fill these gaps is topic of current research. The Quantum Hadamard Edge Detection algorithm was suggested as a more efficient alternative [7]. Implementations for a state vector simulator for an 8 x 8 pixel gray value image and for a 2 x 2 pixel image on a real quantum computer are provided in the Qiskit textbook [8]. Larger image sizes are however only discussed briefly.

Many quantum algorithms known so far are promising when tested on quantum computer simulators but are limited by high error rates, small number of qubits, and low coherence times when applied on real quantum computers. This can lead to outcomes too noisy to be interpretable.

Here, we introduce a method motivated by classical filtering and making use of Tacchino's quantum machine learning algorithm [9] and its extension to gray value images [10]. We use a quantum information-based cost function to compare a binary-valued (black and white) image patch of a test image with a binary filter mask. The cost function is based on the scalar product of the encoded quantum states of the two images. To stabilize the results, we average over repeated measurements. We perform this calculation for two filter masks highlighting vertical and horizontal edges, combine their results, and finally obtain the edges in the test image by a simple threshold.

For testing our approach on IBM's superconducting quantum computer 'ibmq_lagos', we use the 30 x 30 binary test image shown in Figure 1. The results are clearly interpretable and correct even without applying any error correction or mitigation techniques to reduce noise. This demonstrates the capabilities of our method in the current Noisy Intermediate Scale Quantum era.

Our algorithm is not limited to the two filter masks. Like in classical filtering, we can add other filter masks and combine the results. For example, we can construct that way a diagonal filter which emphasizes diagonal edges more clearly.

A. Geng is with the Fraunhofer Institute for Industrial Mathematics ITWM, 67663 Kaiserslautern (e-mail: alexander.geng@itwm.fraunhofer.de).

A. Moghiseh is with the Fraunhofer Institute for Industrial Mathematics ITWM, 67663 Kaiserslautern (e-mail: ali.moghiseh@itwm.fraunhofer.de).

C. Redenbach is with the University of Kaiserslautern, 67663 Kaiserslautern (e-mail: redenbach@mathematik.uni-kl.de).

K. Schladitz is with the Fraunhofer Institute for Industrial Mathematics ITWM, 67663 Kaiserslautern (e-mail: katja.schladitz@itwm.fraunhofer.de).

Moreover, our algorithm can be easily extended to gray value images, too. We only need to change the input values for the image and the filter masks, whereas the algorithm itself stays the same. Thus, our method is not limited to detect edges as other methods like [6, 7]. For example, we can adapt the algorithm to enhance, denoise, or blur the image.

To summarize, we implement an edge detector for larger images on a real quantum computer. To our knowledge, this has not been done before. Our algorithmic idea is based on quantum machine learning. The general setting and variability of our method for solving also other tasks without much additional effort is a clear advantage over pure edge detection methods.

Keywords— Quantum image processing, quantum edge detection, quantum artificial neurons, IBM Quantum Experience

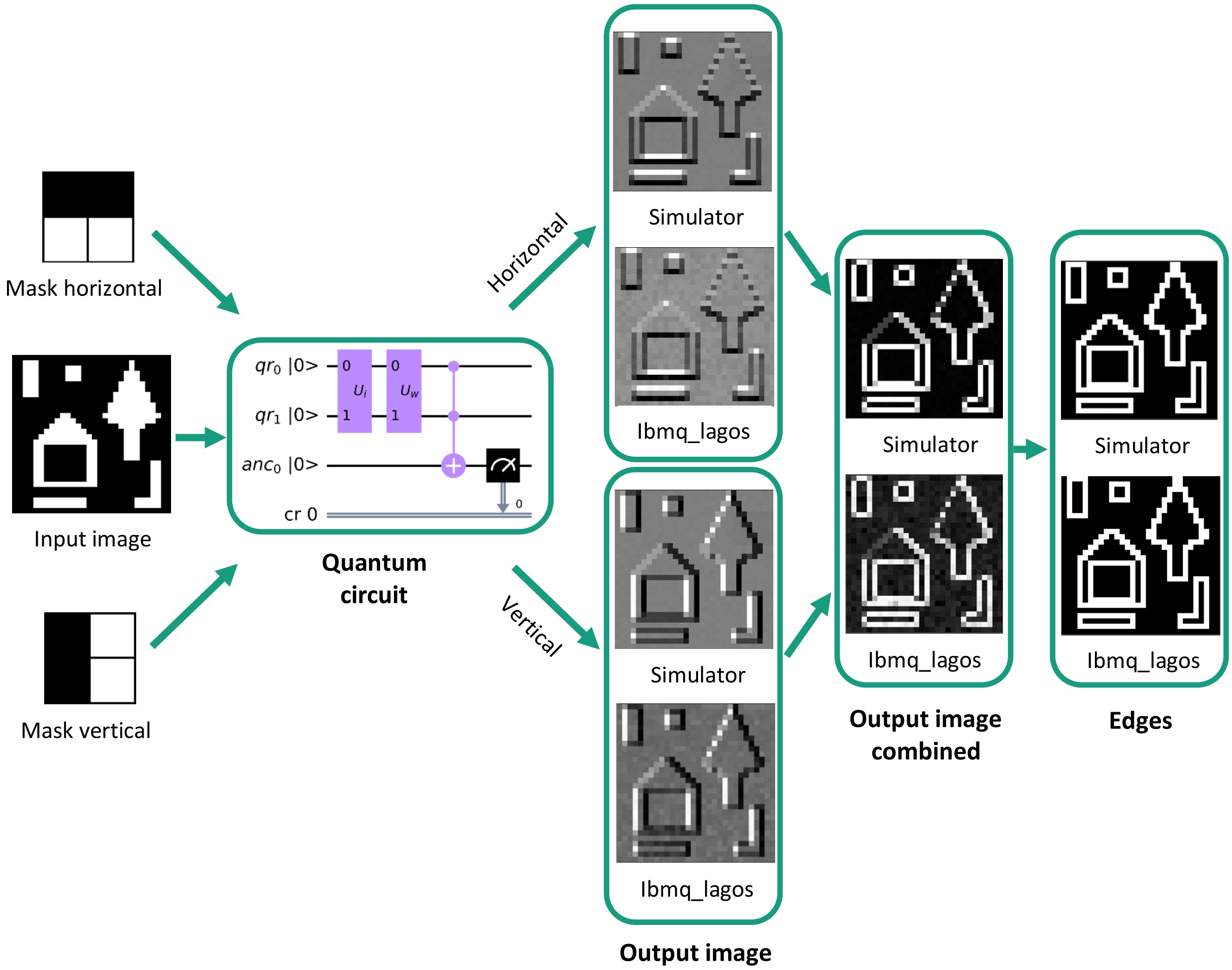


Figure 1: Input image (30 x 30), horizontal and vertical masks (2 x 2), quantum circuit, resulting heat maps for horizontal and vertical masks with mirroring as edge treatment, combination of the two heat maps, and edge detected images using a threshold. For the heat maps, IBM's `qasm_simulator` and backend `ibmq_lagos` are used.

Short abstract

Edge detection is a routine task in digital image processing. Here, we present a quantum edge detection algorithm based on a quantum artificial neuron [9] in two orthogonal directions. This machine learning approach can be easily extended to solve other tasks besides edge detection by adapting the filter masks accordingly.

[1] Gonzalez, Rafael C., and Richard E. Woods. *Digital Image Processing*. Pearson, 2018.
[2] Mastriani, Mario. "Quantum edge detection for image segmentation in optical environments." *arXiv preprint arXiv:1409.2918* (2014).
[3] Fan, Ping, Ri-Gui Zhou, Wen Wen Hu, and NaiHuan Jing. "Quantum image edge extraction based on Laplacian operator and zero-cross method." *Quantum Information Processing* 18.1 (2019): 1-23.
[4] Widiyanto, Slamet, Dini Sundani, Yuli Karyanti, and Dini Tri Wardani. "Edge detection based on quantum canny enhancement for medical imaging." *IOP Conference Series: Materials Science and Engineering*. Vol. 536. No. 1. IOP Publishing, 2019.
[5] Ma, Yulin, Hongyang Ma, and Pengcheng Chu. "c." *IEEE Access* 8 (2020): 210277-210285.
[6] Zhang, Yi, Kai Lu, and YingHui Gao. "QSobel: a novel quantum image edge extraction algorithm." *Science China Information Sciences* 58.1 (2015): 1-13.
[7] Yao, Xi-Wei, et al. "Quantum image processing and its application to edge detection: theory and experiment." *Physical Review X* 7.3 (2017): 031041.
[8] Asfaw, Abraham, et al. "Learn quantum computation using Qiskit (2021)." http://community.qiskit.org/textbook. Accessed 30 August 2021.
[9] Tacchino, Francesco, Chiara Macchiavello, Dario Gerace, and Daniele Bajoni. "An artificial neuron implemented on an actual quantum processor." *npj Quantum Information* 5.1 (2019): 1-8.
[10] Mangini, Stefano, Francesco Tacchino, Dario Gerace, Chiara Macchiavello, and Daniele Bajoni. "Quantum computing model of an artificial neuron with continuously valued input data." Machine Learning: Science and Technology 1.4 (2020): 045008.

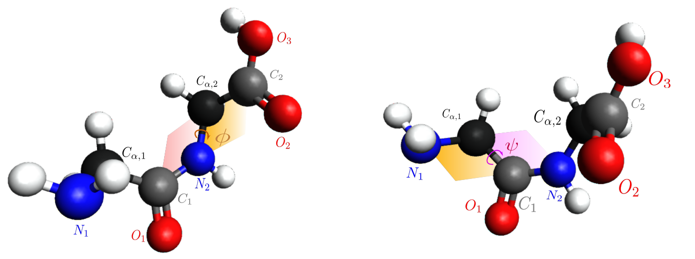
QFold: Quantum Walks and Deep Learning to Solve Protein Folding

P. A. M. Casares,[@†] Roberto Campos,[*†] M. A. Martin-Delgado[†]

[@]pabloamo@ucm.es, [*]robecamp@ucm.es, [†]Universidad Complutense Madrid

Protein folding

Proteins are complex biomolecules, made up of one or several chains of amino acids, and with a large variety of functions in organisms. To discover its function we need to discover its spatial folding, encoded as a sequence of torsion angles and difficult to see experimentally.



Structure of the algorithm

The main steps of the proposed algorithm are:

1. First, starting from the aminoacid sequence, we use a Deep Learning solution, in this case Minifold [2] in substitution of AlphaFold [3].
2. Then, perform a quantum Metropolis algorithm similar to the proposed in [4] to speedup finding the ground state of the system.

Quantum Metropolis Algorithm

In the second part of the algorithm we implement a quantum metropolis algorithm to find the state with the least energy from the guess given by a Deep Learning solution.

Szegedy's quantum walk [1] is defined on a bipartite graph. Given the acceptance probabilities $\mathcal{W}_{ij} = T_{ij}A_{ij}$, A_{ij} defined in (5), for the transition from state i to state j, one defines the unitary

$$U\left|j\right\rangle\left|0\right\rangle := \left|j\right\rangle\sum_{i\in\Omega}\sqrt{\mathcal{W}_{ji}}\left|i\right\rangle = \left|j\right\rangle\left|p_j\right\rangle. \tag{1}$$

Taking

$$R_0 := \mathbf{1} - 2\Pi_0 = \mathbf{1} - 2(\mathbf{1}\otimes\left|0\right\rangle\left\langle 0\right|) \tag{2}$$

the reflection over the state $|0\rangle$ in the second subspace, and S the swap gate that swaps both subspaces, we define the quantum walk step as

$$W := U^\dagger S U R_0 U^\dagger S U R_0. \tag{3}$$

The quantum Metropolis algorithm that we employ [4] uses a small modification of the Szegedy quantum walk, substituting the bipartite graph by a coin. That is, we will have 3 quantum registers: $|\cdot\rangle_S$ indicating the current state of the system, $|\cdot\rangle_M$ that indexes the possible moves one may take, and $|\cdot\rangle_C$ the coin register. We may also have ancilla registers $|\cdot\rangle_A$. The quantum walk step is then

$$\tilde{W} = RV^\dagger B^\dagger F B V. \tag{4}$$

Here V prepares a superposition over all possible steps one may take in register $|\cdot\rangle_M$, B rotates the coin qubit $|\cdot\rangle_C$ to have amplitude of $|1\rangle_C$ corresponding to the acceptance probability indicated by

$$A_{ij} = \min\left(1, e^{-\beta(E_j - E_i)}\right), \tag{5}$$

F changes the $|\cdot\rangle_S$ register to the new configuration (conditioned on the value of $|\cdot\rangle_M$ and $|\cdot\rangle_C = |1\rangle_C$), and R is a reflection over the state $|0\rangle_{MCA}$. We will use the following metropolis algorithm

$$|\psi(L)\rangle := \tilde{W}_L ... \tilde{W}_1 |\pi_0\rangle, \tag{6}$$

where $t = 1, ..., L$ also defines an annealing schedule

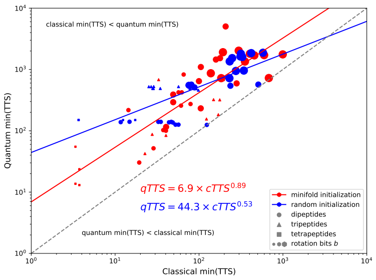
Figure of merit

A natural metric to use in this context is then the Total Time to Solution (TTS) [4] defined as the average expected time it would take the procedure to find the solution if we can repeat the procedure in case of failure:

$$TTS(t) := t\frac{\log(1-\delta)}{\log(1-p(t))}. \tag{7}$$

Simulation results

Our aim is to study whether using a **heuristic** quantum Metropolis Algorithm can give a quantum advantage in this problem compared with its classical counterpart.

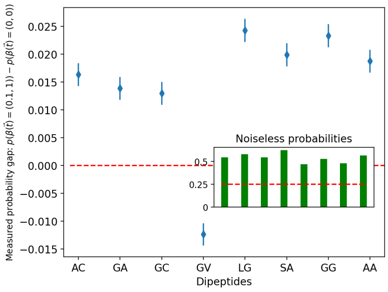


Fit exponents			
Schedule	Schedule definition	Random initialization	Minifold initialization
Fixed β	$\beta(t) = \beta$	0.53	0.89
Logarithmic	$\beta(t) = \beta(1)\log(te) = \beta(1)\log(t) + \beta(1)$	0.29	0.88
Linear	$\beta(t) = \beta(1)t$	0.34	0.86
Exponential	$\beta(t) = \beta(1)\alpha^{-t+1}$	0.37	1.00
Geometric	$\beta(t) = \beta(1)\exp(\alpha(t-1)^{1/N})$	0.29	0.85

Table 1: Table of scaling exponents for different annealing schedules and initialization options. For fixed β, the value heuristically chosen was $\beta = 1000$, while the initial β value in each of the schedules is $\beta(1) = 50$.

Qiskit experiments

We also run the smallest possible version of our algorithm in IBMQ Casablanca Due to the high level of noise our aim is to check whether there is a difference in probability in the experiment when we use $\beta = 0$, where all states should be equiprobable, and higher β values, where the probability of the target state is higher. Our experiment aims to see this difference of probability in practice.



References

[1] M. Szegedy, "Quantum speed-up of markov chain based algorithms," in *45th Annual IEEE Symposium on Foundations of Computer Science*, pp. 32–41, IEEE, 2004.

[2] E. Alcaide, "Minifold: a deeplearning-based mini protein folding engine." https://github.com/EricAlcaide/MiniFold/, 2019.

[3] A. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Zidek, A. Nelson, A. Bridgland, *et al.*, "Improved protein structure prediction using potentials from deep learning," *Nature*, 2020.

[4] J. Lemieux, B. Heim, D. Poulin, K. Svore, and M. Troyer, "Efficient Quantum Walk Circuits for Metropolis-Hastings Algorithm," *Quantum*, vol. 4, p. 287, June 2020.

Binary classification of the MNIST dataset using quantum models

Alberto Maldonado Romo[1], Jesús Yalja Montiel Pérez[1]

[1] *Centro de Investigación en Computación, Instituto Politécnico Nacional, Mexico*

Contact e-mail: alberto.maldo1312@gmail.com, yalja@ipn.mx

Abstract We perform binary classification of the MNIST dataset using two models: quantum k-nearest neighbors, mapping by angles that are into a QRAM, Grover's algorithm, and the SWAP-Test; Quantum Neural Network, mapping by amplitude and variational quantum circuits. Were developed in Qiskit and PennyLane, and a noise model of 5 qubits.

Keywords Quantum Machine Learning, Quantum classification, Quantum software, Quantum Neural Network, Quantum Random Access Memory, Benchmark.

Introduction Classical data must undergo preprocessing to be mapped to qubits. In this work, we perform classical-to-quantum data processing by angle mapping to the quantum k-nearest neighbor (quantum k-nn) algorithm and amplitude mapping to a Quantum Neural Network (QNN) [1] [2]. Different work schemes are performed for each model. The quantum k-nn implementation is based on Grover's algorithm and QRAM [3] [4], applying the SWAP test to decide similarity. As for the QNN, the Variational Quantum Circuit (VQC) or ansatz[5] [6] [7] is used to find from the expected value $\langle Z\rangle$ of the measurement to obtain the correct class, relying on a classical optimizer, Nesterov Momentum, to update the angles of our ansatz.

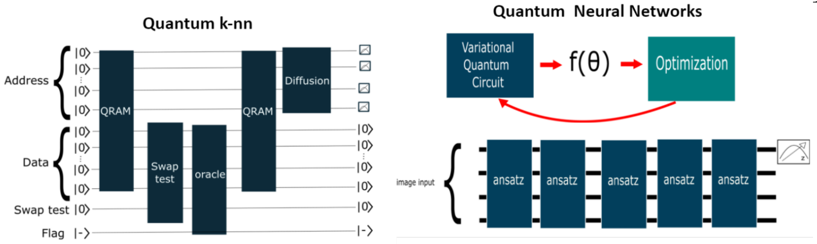


Figure 1: Schemes for generating Quantum k-nn and Quantum Neural Networks.

Methods For the quantum k-nn, consider the following mapping method to convert an image to a qubit representation. Note the equality of the tensor product

$$V_1 \otimes \ldots \otimes V_n = W_{2^n}, \tag{1}$$

where V_1 to V_n are vectors of size 1×2 and W_{2^n} is a vector of size 2^n, that contains a vector state of size $p \times q = n$, where p is the width of the image, q the height of the image, and n the number of qubits. The QRAM has the feature of accessing multiple memory cells through an address superposition. Consider the address register a, where a is equal to 2^k and k is the number of qubits. The address superposition is

$$\sum_j \psi_j |j\rangle_a \rightarrow \sum_j \psi_j |j\rangle_a \, |D_j\rangle_d \,, \tag{2}$$

where D_j is the content of the jth memory cell [3]. This structure can be represented as a binary tree depending on the state $|0\rangle$ or $|1\rangle$ or multiple nodes are selected, with a space complexity of O(log(n)) [8]. The QRAM oracle finds the distance between two instances, for instance by computing the inner product (or dot product) which is defined as $|a - b| = |a||b| - a \cdot b$, where $|a\rangle$ and $|b\rangle$ are two quantum states. It is necessary to use the SWAP test [2], which can be used to estimate the fidelity of two pure states, i.e. $|\langle a \mid b\rangle|^2$. The case of the QNN [9] is based on amplitude mapping. To go from images to a state vector, the following equation is considered

$$V_{\text{normalize}} = \frac{V}{\sqrt{\sum_i V_i^2}} \tag{3}$$

where the vector V has 2^n elements. We mapped $V_{normalize}$ using the method developed by [10]. The ansatz for each layer in the neural network is composed of Ry and Rz rotations, in such a configuration as to span all possible coordinates of each qubit. The measured qubit in this binary classifier is post-selected to equal the $|q_0\rangle$ state [11] [12].

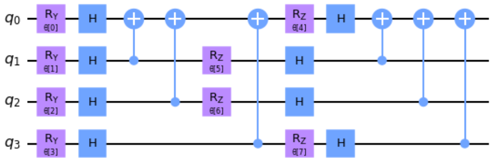


Figure 2: Ansatz or layer for the QNN.

Results Experiment were performed with two MNIST classes, 3 and 6, for classification; and resizing from 28 × 28 to 4 × 4 in order to be able to work with 4 qubits per image [13]. The case of the quantum k-nn has a function from the number of qubits for each training set to compare with the test set $2^{log_2 m} + 10$, where m is the number of instance. In the same way, it was realised that the more gates used in 1-nn the lower the performance, where it is shown that the best performance was with 64 qubits instances and in case of 256 the worst. The case of the Quantum Neural Networks, the k-fold cross validation method was used, with k = 3; 15 epoch; two training sets 700 and 7000 with 4,10,20 layers respectively of the same layer and using noise simulation by the real "ibmq_lima" computer without error mitigation. The Quantum k-nn model requires numerous qubits —at least 12— in order to have a training set of 2 instances. This means that operating with more than 2048 instances of data is not reliable, as the task size is beyond that supported by present-day simulators. QNN works with 4 qubits, and with large amounts of data in the simulator and a noise model that is equivalent to the computer ibmq_lima. Showed that 4 layers with 700 images gives 78% accuracy, and an error less of 1% using 24 CNOTs gates. Increasing the layers to 20 that means 120 CNOTs, has 50% of accuracy in the noise model. With 10 layers, being 60 CNOTs and 7,000 images gives 91% of accuracy in the noise model. Exist a relation between CNOTs gates and the size of the train set. From epoch 6 to epoch 15, we observed that increasing epoch resulted in our model overtraining, lowering performance by up to 10%. Therefore, our benchmarking found that the best performance was obtained using VQC, where with this number of CNOTs gates and size of the training set, we succeeded in the classification task with an average value of 91% accuracy.

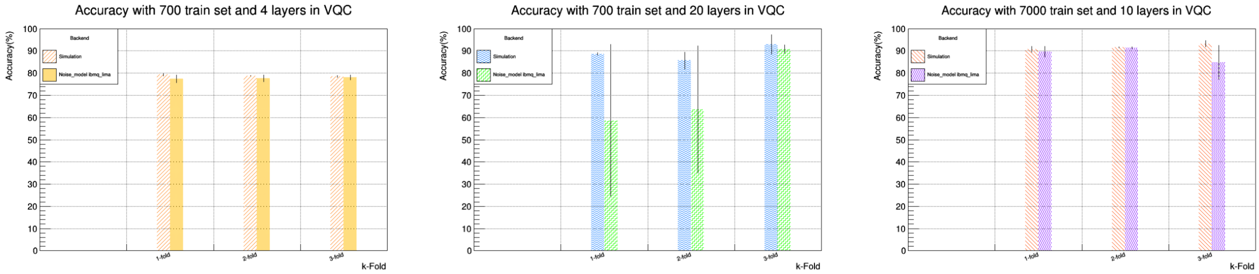


Figure 3: There are 3 experiments left, centre and right, from each an mean value was taken for the three folds, being 78.88%, 89.12% and 91.88%, and is used to obtain the relative error.

References

[1] N. Wiebe, A. Kapoor, and K. Svore, "Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning," *Quantum Information and Computation*, vol. 15, pp. 318–358, Mar. 2015.

[2] Y. Ruan, X. Xue, H. Liu, J. Tan, and X. Li, "Quantum algorithm for k-nearest neighbors classification based on the metric of hamming distance," *International Journal of Theoretical Physics*, vol. 56, Nov. 2017. DOI: 10.1007/s10773-017-3514-4.

[3] V. Giovannetti, S. Lloyd, and L. Maccone, "Quantum random access memory," *Physical review letters*, vol. 100, p. 160 501, Apr. 2008. DOI: 10.1103/PhysRevLett.100.160501.

[4] B. Toghi and D. Grover, "Mnist dataset classification utilizing k-nn classifier with modified sliding window metric," Sep. 2018.

[5] P. Katyayan and N. Joshi, "Quantum computing: Start your journey with qiskit! — open source for you (december 2020)," Dec. 2020. DOI: 10.13140/RG.2.2.32232.08961.

[6] E. Farhi and H. Neven, "Classification with quantum neural networks on near term processors," Feb. 2018.

[7] E. Grossi and M. Buscema, "Introduction to artificial neural networks," *European journal of gastroenterology hepatology*, vol. 19, pp. 1046–54, Jan. 2008. DOI: 10.1097/MEG.0b013e3282f198a0.

[8] C. Hann, G. Lee, S. Girvin, and L. Jiang, "Resilience of quantum random access memory to generic noise," *PRX Quantum*, vol. 2, Apr. 2021. DOI: 10.1103/PRXQuantum.2.020311.

[9] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, "Parameterized quantum circuits as machine learning models," *Quantum Science and Technology*, vol. 4, Oct. 2019. DOI: 10.1088/2058-9565/ab4eb5.

[10] M. Möttönen, J. Vartiainen, V. Bergholm, and M. Salomaa, "Transformation of quantum states using uniformly controlled rotations," *Quantum Information Computation*, vol. 5, pp. 467–473, Sep. 2005. DOI: 10.26421/QIC5.6-5.

[11] M. Schuld and F. Petruccione, "Supervised learning with quantum computers," *Springer*, 2018. DOI: 10.1007/978-3-319-96424-9.

[12] B. LaRose Ryan Coyle, "Robust data encodings for quantum classifiers," 2020. DOI: 10.1103/PhysRevA.102.032420,arXiv:2003.01695.

[13] A. Mutiara, M. Amir, R. Refianti, and Y. Sutanto, "Handwritten numeric image classification with quantum neural network using quantum computer circuit simulator," *International Journal of Advanced Computer Science and Applications*, vol. 11, Jan. 2020. DOI: 10.14569/IJACSA.2020.0111040.

The dilemma of quantum neural networks

Yang Qian   Xinbiao Wang   Yuxuan Du [*]   Xingyao Wu   Dacheng Tao

Abstract

Through systematic numerical experiments, we empirically observe that current QNNs fail to provide any benefit over classical learning models on real-world tasks. First, QNNs suffer from the severely limited effective model capacity. Second, the trainability of QNNs is insensitive to regularization techniques, which sharply contrasts with the classical scenario. Arxiv

1 Introduction

Driven by the promising empirical achievements of QML and the significance of understanding the power of QNNs, initial studies have been conducted to explore the trainability and the generalization ability of QNNs [1, 2, 3, 4, 5, 6, 7] by leveraging varied model complexity measures developed in statistical learning theory [8] and advanced tools in optimization theory [9]. Notably, the obtained results transmitted both positive and negative signals. To be more concrete, theoretical evidence validated that QNNs can outperform DNNs for specific learning tasks, i.e., quantum synthetic data classification [10] and discrete logarithm problem [6]. However, Ref. [11] revealed the barren plateaus' issue of QNNs, which challenges the applicability of QNNs on large-scale problems. Considering that an ambitious aim of QNNs is providing computational advantages over DNNs on real-world tasks, it is important to answer: '*Are current QNNs sufficient to solve certain real-world problems with potential advantages?*' If the response is negative, it is necessary to figure out '*how is the gap between QNNs and DNNs?*'

Problem setup. We inherit the tradition in DNNs to understand the trainability and generalization of QNNs [12]. Particularly, the explicit form of the measure of the *generalization error bound* is

$$\hat{\mathcal{R}}_S(\hat{\boldsymbol{\theta}}) - \mathcal{R}(\hat{\boldsymbol{\theta}}) := \frac{1}{n}\sum_{i=1}^{n} C\left(h(\hat{\boldsymbol{\theta}}, \boldsymbol{x}^{(i)}), \boldsymbol{y}^{(i)}\right) - \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}\left(C(h(\hat{\boldsymbol{\theta}}, \boldsymbol{x}), \boldsymbol{y})\right),$$

where $S = \{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})\}_{i=1}^{n}$ denotes the given training dataset sampled from the domain $\mathcal{X} \times \mathcal{Y}$, $h(\hat{\boldsymbol{\theta}}, \cdot) \in \mathcal{H}$ refers to the hypothesis inferred by QNN with $\mathcal{H}$ being the hypothesis space and $\hat{\boldsymbol{\theta}}$ being the trained parameters, $C : \mathcal{H} \times (\mathcal{X} \times \mathcal{Y}) \to \mathbb{R}^{+}$ is the designated loss function, and $\hat{\mathcal{R}}_S(\hat{\boldsymbol{\theta}})$ (or $\mathcal{R}(\hat{\boldsymbol{\theta}})$) represents the empirical (or expected) risk [13]. The generalization error bound in Eqn. (1) concerns when and how minimizing $\hat{\mathcal{R}}_S(\hat{\boldsymbol{\theta}})$ is a sensible approach to minimizing $\mathcal{R}(\hat{\boldsymbol{\theta}})$. A low error bound suggests that the unearthed rule $h(\hat{\boldsymbol{\theta}})$ from the dataset S can well generalize to the unseen data sampled from the same domain. Note that since the probability distribution behind data domain is generally inaccessible, the term $\mathcal{R}(\hat{\boldsymbol{\theta}})$ is intractable. A generic strategy is employing the test dataset $\tilde{S} \sim \mathcal{X} \times \mathcal{Y}$ to estimate this term, i.e., $\mathcal{R}(\hat{\boldsymbol{\theta}}) \approx \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}} \ell(h(\hat{\boldsymbol{\theta}}, \tilde{\boldsymbol{x}}^{(i)}), \tilde{\boldsymbol{y}}^{(i)})$ with $(\tilde{\boldsymbol{x}}^{(i)}), \tilde{\boldsymbol{y}}^{(i)}) \in \tilde{S}$.

The *trainability* concerns the convergence rate of the trained parameters of QNN towards the optimal parameters. Considering that the loss landscape of QNNs is generally non-convex and non-concave, which implies the computational hardness of seeking optimal $\boldsymbol{\theta}$, an alternative way to examine the trainability of QNN is analyzing its convergence rate, i.e.,

$$\mathcal{J}(\boldsymbol{\theta}) = \mathbb{E}[\|\nabla_{\boldsymbol{\theta}} \hat{\mathcal{R}}_S(\boldsymbol{\theta})\|], \tag{1}$$

where the expectation is taken over the randomness from the sample error and gate noise [4]. In other words, the metric $\mathcal{J}(\boldsymbol{\theta})$ evaluates how far the trainable parameters of QNN are away from the stationary point $\|\nabla_{\boldsymbol{\theta}} \hat{\mathcal{R}}_S(\boldsymbol{\theta})\| = 0$.

[*] Corresponding author, duyuxuan123@gmail.com

2 Results

In this section, we consider three typical quantum neural networks including quantum naive neural network (QNNN) [10], quantum embedding neural network (QENN) [14], and quantum convolutional neural network (QCNN) [15], and experimentally compare their generalization ability and trainability with classical neural networks. i.e., the multi-layer perceptron (MLP) and convolutional neural network (CNN) on a quantum synthetic dataset and two real-world datasets, i.e., the Wine dataset [16] and MNIST handwritten digit dataset [17]. Note that models involved in the comparison are designed to own similar number of trainable parameters.

For analysis of generalization ability, randomization experiments are firstly conducted to measure the effective model capacity, which directly determines models' generalization. The results reveal that DNNs overwhelm QNNs in fitting random labels, regardless of whether the training data is quantum or classical. Based on this observation, the generalization errors are evaluated on the datasets with true labels, shown in the left panel of Fig. 1. It is implicated that the learning performance of current QNNs is no better than DNNs on real-world datasets and the limited model capacity is further reduced by imperfection of NISQ machines. The right panel in Fig. 1 describes the effects of typical regularization techniques on the trainability of quantum models. The results indicate that the widely used regularization techniques in classical deep learning plays a different role in quantum machine learning. Some regularization strategies such as weight decay fail to enhance the trainability of QNNs.

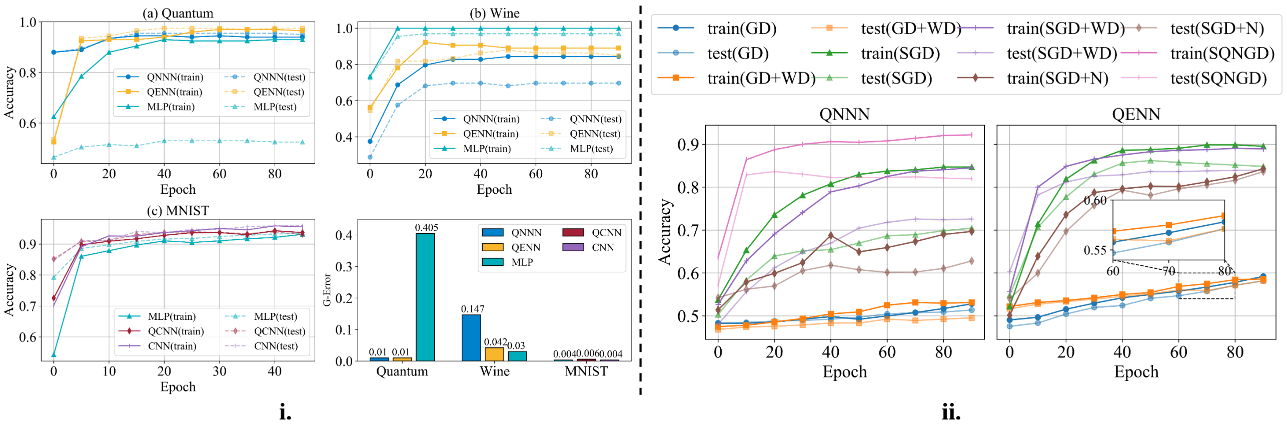


Figure 1: **Generalization and trainability of quantum neural networks and classical neural networks**. *left panel*: generalization performance on quantum data and classical data with true labels. G-Error represents the generalization error. (a), (b) and (c) show the accuracy of various models changing with training epochs, when training on quantum synthetic data, the Wine data and MNIST respectively. The bar chart inserted into each figure represents the generalization error of each model. *right panel*: effects of regularizations on the trainability of quantum model on Wine dataset. The labels 'GD', 'SGD', 'SQNGD', 'WD', and 'N' refer to the gradient descent optimizer, stochastic gradient descent optimizer, the stochastic quantum natural gradient descent optimizer, the weight decay, and execution of experiments on NISQ chips, respectively. SGD plays a significant role in accelerating convergence and achieving higher accuracy, while others the optimization process instead of boosting performance.

3 Summary

In this study, we proceed systematic numerical experiments to understand the generalization ability and trainability of typical QNNs in the view of statistical learning theory. The achieved results exhibited that current QNNs struggle a poor effective model capacity. This observation well explains why current QNNs can attain computational advantages on quantum synthetic data classification tasks and discrete logarithm problems, while they fail to compete with DNNs in tackling real-world learning tasks. Moreover, our study illustrate that the regularization techniques, which greatly contribute to the success of DNNs, have limited effects towards the trainability of QNNs. In addition, our study exhibits that quantum system noise suppresses the learnability of QNNs, which echoes with the theoretical study [5]. Last, to alleviate the dilemma of current QNNs, we discuss several prospective directions such as designing over-parameterized QNNs without barren plateaus and developing effective error mitigation techniques.

References

[1] Amira Abbas, David Sutter, Christa Zoufal, Aurélien Lucchi, Alessio Figalli, and Stefan Woerner. The power of quantum neural networks. *arXiv preprint arXiv:2011.00027*, 2020.

[2] Leonardo Banchi, Jason Pereira, and Stefano Pirandola. Generalization in quantum machine learning: a quantum information perspective. *arXiv preprint arXiv:2102.08991*, 2021.

[3] Kaifeng Bu, Dax Enshan Koh, Lu Li, Qingxian Luo, and Yaobo Zhang. On the statistical complexity of quantum circuits. *arXiv preprint arXiv:2101.06154*, 2021.

[4] Yuxuan Du, Min-Hsiu Hsieh, Tongliang Liu, Shan You, and Dacheng Tao. On the learnability of quantum neural networks. *arXiv preprint arXiv:2007.12369*, 2020.

[5] Yuxuan Du, Zhuozhuo Tu, Xiao Yuan, and Dacheng Tao. An efficient measure for the expressivity of variational quantum algorithms. *arXiv preprint arXiv:2104.09961*, 2021.

[6] Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R McClean. Power of data in quantum machine learning. *arXiv preprint arXiv:2011.01938*, 2020.

[7] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Information-theoretic bounds on quantum advantage in machine learning. *arXiv preprint arXiv:2101.02464*, 2021.

[8] Vladimir Vapnik. Principles of risk minimization for learning theory. In *Advances in neural information processing systems*, pages 831–838, 1992.

[9] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[10] Vojtěch Havlíček, Antonio D Córcoles, Kristan Temme, Aram W Harrow, Abhinav Kandala, Jerry M Chow, and Jay M Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209, 2019.

[11] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature communications*, 9(1):1–6, 2018.

[12] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. In *Advances in neural information processing systems*, pages 6158–6169, 2019.

[13] Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. Generalization in deep learning. *arXiv preprint arXiv:1710.05468*, 2017.

[14] Seth Lloyd, Maria Schuld, Aroosa Ijaz, Josh Izaac, and Nathan Killoran. Quantum embeddings for machine learning. *arXiv preprint arXiv:2001.03622*, 2020.

[15] Maxwell Henderson, Samriddhi Shakya, Shashindra Pradhan, and Tristan Cook. Quanvolutional neural networks: powering image recognition with quantum circuits. *Quantum Machine Intelligence*, 2(1):1–9, 2020.

[16] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[17] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Accelerating variational quantum algorithms with multiple quantum processors

Yuxuan Du [*] Yang Qian Dacheng Tao

Abstract

We devise an efficient distributed optimization scheme, called QUDIO, to improve the runtime efficiency of variational quantum algorithms (VQAs). Numerical simulation demonstrates that QUDIO can surprisingly achieve a superlinear runtime speedup with respect to the number of local nodes, while theoretical analyses prove the convergence of QUDIO. Arxiv

1 Introduction

Variational quantum algorithms (VQAs) have the potential of utilizing near-term quantum machines to gain certain computational advantages over classical methods. Nevertheless, modern VQAs suffer from cumbersome computational overhead, hampered by the tradition of employing a solitary quantum processor to handle large-volume data. To conquer the above issues, here we devise an efficient `QUantum` `DIstributed` `Optimization` scheme (abbreviated as QUDIO). An attractive property of QUDIO is *adequately* utilizing the accessible quantum resources to accelerate VQAs, owing to its compatibility. Namely, the deployed quantum processors are allowed to be any type of quantum hardware such as linear optical, ion-trap, and superconducting quantum chips. Such a compatibility contributes to apply a wide class of VQAs to manipulate varied large-scale computational problems and seek potential quantum advantages by unifying quantum powers in a maximum extent. The pseudocode of QUDIO is presented in Fig. 1

```
Input: The initialized parameters $\boldsymbol{\theta}^{(0)} \in [0, 2\pi)^{d_Q}$, the employed loss function $\mathcal{L}$, the given dataset/Hamilton, the hyper-parameters $\{Q, \eta, W, T\}$
The central server partitions the given problem into $Q$ parts and allocates them to $Q$ local nodes
for $t = 0, \cdots, T-1$ do
    for Quantum processor $\mathcal{Q}_i, \forall i \in [Q]$ in parallel do
        $\boldsymbol{\theta}_i^{(t,0)} = \boldsymbol{\theta}^{(t)}$
        for $w = 0, \cdots, W-1$ do
            Compute the estimate gradients $g_i^{(t,w)}$
            Update $\boldsymbol{\theta}_i^{(t,w+1)} = \boldsymbol{\theta}_i^{(t,w)} - \eta g_i^{(t,w)}$
        end for
    end for
    Synchronize $\boldsymbol{\theta}^{(t+1)} = \frac{1}{Q}\sum_{i=1}^{Q} \boldsymbol{\theta}_i^{(t,W)}$
end for
Output: $\boldsymbol{\theta}^{(T)}$
```

Figure 1: **The Pseudocode of QUDIO**. The codes highlighted by the yellow and pink shadows refer to execute them on the local nodes and the central server, respectively.

[*] Corresponding author, duyuxuan123@gmail.com

2 Results

We carry out numerical simulations to exhibit how QUDIO accelerates QNNs when dealing with a standard binary classification task with a large size of training examples. As shown in Fig. 2, for both the ideal and NISQ cases, QUDIO gains the speedup when increasing the number of local nodes Q. To be more convincing, we also perform numerical simulations to validate the effectiveness of QUDIO to accelerate conventional VQEs, as shown in Fig. 3.

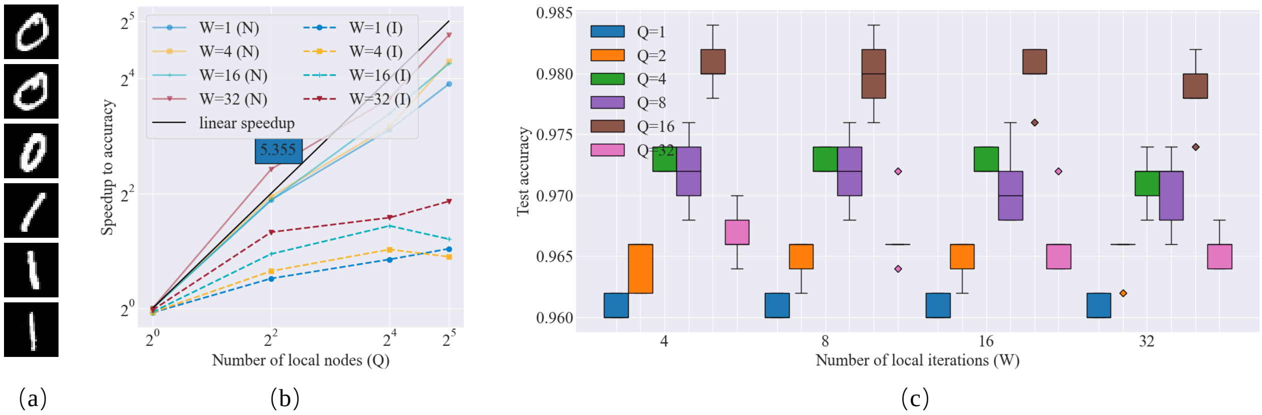


Figure 2: **Simulation results of QUDIO towards hand-written digits image classification.** (a) A visualization of some training examples sampled from the MNIST dataset. (b) Scaling behavior of QUDIO in clock-time for increasing number of local nodes Q for varied number of local steps W. The labels '$W = a$ (I)' and '$W = a$ (N)' refer that the total number of local iterations is $W = a$ under the ideal and NISQ scenarios respectively. The hyper-parameters settings for the NISQ case are $p = 10^{-5}$ and $K = 100$. (c) A box plot that illustrates the achieved test accuracy of QUDIO with varied W and Q in the NISQ scenario, where the hyper-parameters settings are same with those described in (b).

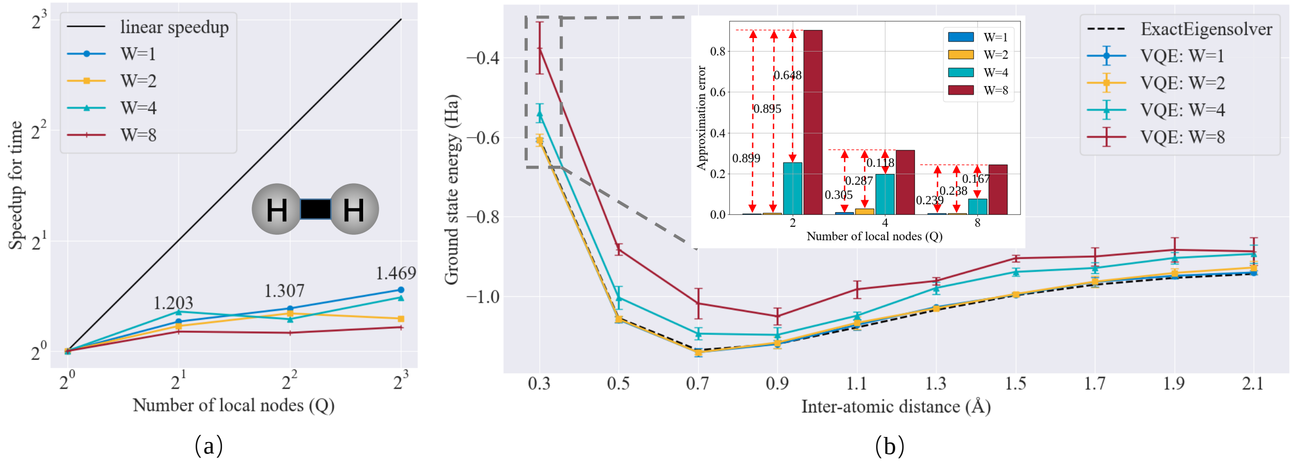


Figure 3: **Simulation results of QUDIO towards the ground state energy estimation of hydrogen molecule.** (a) The speedup ratio with respect to different number of local nodes Q and local iterations W. (b) The potential energy surface estimated by QUDIO. The black dotted line represents the exact ground state energy. The inner plot compares the error between the ground truth and the estimated results of QUDIO with the case of 0.3 inter-atomic distance.

3 Summary

In this study, we devise QUDIO to accelerate VQAs with multiple quantum processors. We also provide theoretical analysis about how the system noise and the number of measurements influence the convergence of QUDIO. An attractive feature is that in the ideal setting, QUDIO obeys the asymptotic convergence rate with conventional QNNs, which ensures its runtime speedup with respect to the increased number of local nodes. The achieved numerical simulation results confirm the effectiveness of our proposal. Particularly, in the NISQ scenario, QUDIO can achieve superline speedups in the measure of time-to-accuracy.

LHC event generation with a style quantum GAN

Carlos Bravo-Prieto,[1,2] Julien Baglio,[3] Marco Cè,[3] Anthony Francis,[4,3]
Dorota M. Grabowska,[3] and Stefano Carrazza[5,3,1]

[1] *Quantum Research Centre, Technology Innovation Institute, Abu Dhabi, UAE*
[2] *Departament de Física Quàntica i Astrofísica and Institut de Ciències del Cosmos (ICCUB), Universitat de Barcelona, Barcelona, Spain.*
[3] *Theoretical Physics Department, CERN, CH-1211 Geneva 23, Switzerland.*
[4] *Institute of Physics, National Yang Ming Chiao Tung University, Hsinchu 30010, Taiwan.*
[5] *TIF Lab, Dipartimento di Fisica, Università degli Studi di Milano and INFN Sezione di Milano, Milan, Italy.*

During the last decade we have witnessed an impressively fast development of quantum computing and today's quantum devices are typically at the so-called Noisy Intermediate-Scale Quantum (NISQ) [1] stage. NISQ devices suffer from errors due to decoherence, noisy gates and errors in the read-out measurements. Nevertheless, even at this stage quantum technologies may provide useful tools for a broad range of applications.

In particular, there has been progress with quantum algorithms designed to address problems in high energy physics (HEP), see e.g. [2]. Some of these are characterized as Quantum Machine Learning (QML) applications, based on variational and non-variational approaches.

Motivated by this outlook, we investigate the possibility of using quantum computing for the generation of Monte Carlo (MC) events through quantum generative adversarial circuits. As such we are focussing on applications in HEP and the physics being researched at the Large Hadron Collider (LHC) at CERN in Geneva, Switzerland. Here the key goal is to extract physical observables and to compare them to the corresponding theoretical predictions from the Standard Model (SM) of particle physics. This comparison step requires complex numerical simulations where millions of MC events are needed and the computational effort is immense. Quantum generative models could provide a significant acceleration of MC event generation.

With this in mind, in this contribution we describe practicable variational quantum circuit models for the generation of MC events in the described HEP context. In detail we investigate and identify the most suitable ansatz for the parametrization of a quantum generative network and through simulation on classical hardware we show that our style-qGAN generators are suitable MC event generators [3].

We propose, implement and test a reconstruction method for evaluating the qGAN model measurements on current IBM-Q and IonQ quantum devices and find promising results. Further development of hardware architectures with lower gate-error tolerances, as well as error mitigation techniques, are still required to obtain more stable results and the presented qGAN model should be considered as a proof-of-concept. Nevertheless, we believe that the approach presented here will inspire new applications in- and outside of HEP that may benefit from quantum computing.

[1] J. Preskill, Quantum computing in the nisq era and beyond, Quantum **2**, 79 (2018).

[2] A. Pérez-Salinas, J. Cruz-Martinez, A. A. Alhajri, and S. Carrazza, Determining the proton content with a quantum computer, Phys. Rev. D **103**, 034027 (2021), arXiv:2011.13934 [hep-ph].

[3] C. Bravo-Prieto, J. Baglio, M. Cè, A. Francis, D. M. Grabowska, and S. Carrazza, Style-based quantum generative adversarial networks for Monte Carlo events, (2021), arXiv:2110.06933 [quant-ph].

Quantum-inspired heuristic solvers for large-scale linear systems

Oliver Knitter[1], James Stokes[2], Shravan Veerapaneni[1,2]

Much of the motivation behind quantum algorithm research is due to the proven existence of quantum algorithms exhibiting exponential speedups over their classical counterparts, including algorithms for large-scale linear algebra [5]. Furthermore, plausible complexity-theoretic assumptions strongly suggest [9] that quantum computers are capable of preparing quantum states whose output probability distributions are hard to sample classically. Despite this promise, quantum algorithm design suffers from a few caveats.

Firstly, exponential acceleration requires access to *fault-tolerant* quantum hardware, necessitating a prohibitively large number of qubits to be effective. Even under the assumption of polylogarithmic overhead in the asymptotic limit of problem size, the requisite resources for solving linear systems of practical utility vastly exceeds the near-term capabilities of noisy intermediate-scale quantum (NISQ) devices. Secondly, the conjectured hardness results [9] for classical sampling pertain to probability distributions with no known practical utility, absent any further guidance about which quantum states should be targeted for practical problems.

Overcoming these obstacles has led to a new research direction called *variational quantum algorithms* (VQAs), in which the key idea is to encode a computational problem as an optimization problem for an unknown quantum state. More specifically, the probabilistic nature of quantum states dictates that the solution of the computational problem be somehow encoded into the output probabilities of the optimal quantum state, which may be estimated through Monte Carlo sampling. Modeling quantum states classically requires overcoming a curse of dimensionality, since the number of dimensions of a multi-qubit state space grows exponentially with the number of qubits. The potential computational advantage of VQAs stems from their use of a special purpose Quantum Processing Unit (QPU) to perform state preparation. A CPU performs gradient-based updates in tandem with the QPU in order to identify the optimal quantum state. Variational algorithms have already been designed with the goal of solving hard combinatorial optimization [3] and for ground state preparation in quantum chemistry [7]; more recently, VQAs have emerged for solving practically useful high-dimensional linear algebra problems, such as matrix-vector products [12], solutions of linear systems [2] and even singular value decomposition (SVD) [1, 11].

Nonetheless, the existence of a quantum computational advantage—as well as its underlying theoretical justification—has yet to be demonstrated for VQAs. To gain insight about the potential for quantum speedup, we turn to the deep relationship between VQAs and variational quantum Monte Carlo (VQMC), which in the past few years has undergone significant progress in expanding its capabilities to problems beyond its traditional purview. In direct analogy to VQAs, the VQMC overcomes the curse of dimensionality for a restricted subset of linear algebra problems by performing alternating steps of Monte Carlo sampling from a parametrized quantum state, followed by gradient-based optimization. However, unlike VQAs—whose computational advantage hinges on the conjectural difficulty of sampling from a parametrized quantum circuit— the VQMC gains its power by modeling the quantum state using multi-layer parametrized neural networks. The key idea underlying the recent success of VQMC is the exploitation of flexible neural networks as many-body trial wavefunctions, which has made it possible to leverage the enormous success of machine learning (ML) in solving a

[1] Department of Mathematics, University of Michigan, Ann Arbor, MI 48109

[2] Flatiron Institute, Simons Foundation, New York, NY 10010

variety of high-dimensional learning tasks. An example showcasing the successful fusion of techniques from VQMC and ML was Google DeepMind's `FermiNet` [8], whose success relied heavily on ML notions including equivariant function approximation and natural-gradient optimization. The close parallels between ideas developed in very different fields has already provided many opportunities for technology transfer between the fields of VQAs, VQMC and ML, including the discovery of a VQMC-inspired quantum natural gradient optimization algorithm for VQAs [10] and an expanding list of VQA-inspired approaches to classical combinatorial optimization [4, 13, 6].

Although VQMC realizations of linear algebra problems have so far been limited to ground state preparation and Schrodinger evolution, there is no fundamental obstruction to tackling other linear algebra problems of technological interest. In order to better understand the existence and origin of VQA quantum computational advantage, we address the above gap by pursuing VQMC approaches to linear algebra in exponentially high dimensions. Using a recently discovered Rayleigh quotient reformulation, we introduce a VQMC-based solver for the linear system $Ax = b$, where A is some gigantic row-sparse matrix, which we call the Variational Neural Linear Solver (VNLS). In addition to providing a toolkit for performing high-dimensional linear algebra, which is of intrinsic interest, the VNLS provides a quantum-inspired classical benchmark for assessing the quantum computational advantage of Variational Quantum Linear Solver (VQLS) [2], which is a proposed VQA for solving sparse high-dimensional linear systems.

By choosing to focus on VQMC, the question of quantum computational advantage can be isolated from the inherent advantage of Monte Carlo for overcoming the curse of dimensionality. In addition to further understanding the potential utility of VQAs, the VNLS comprises a new scalable algorithm for solving high-dimensional linear systems in the short term. Moreover, it hopefully represents the first example of a new paradigm for efficiently solving a variety of large-scale linear algebra problems. On a longer timeframe, the insights gained by studying VQMC realizations of NISQ algorithms on classical hardware will provide the basis for quantum algorithm technology transfer as quantum hardware matures.

References

[1] Carlos Bravo-Prieto, Diego García-Martín, and José I Latorre. Quantum singular value decomposer. *Physical Review A*, 101(6):062310, 2020.

[2] Carlos Bravo-Prieto, Ryan LaRose, Marco Cerezo, Yigit Subasi, Lukasz Cincio, and Patrick J Coles. Variational quantum linear solver. *arXiv preprint arXiv:1909.05820*, 2019.

[3] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[4] Joseph Gomes, Keri A McKiernan, Peter Eastman, and Vijay S Pande. Classical quantum optimization with neural network quantum states. *arXiv preprint arXiv:1910.10675*, 2019.

[5] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.

[6] Mohamed Hibat-Allah, Estelle M Inack, Roeland Wiersema, Roger G Melko, and Juan Carrasquilla. Variational neural annealing. *arXiv preprint arXiv:2101.10154*, 2021.

[7] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5:4213, 2014.

[8] David Pfau, James S Spencer, Alexander GDG Matthews, and W Matthew C Foulkes. Ab initio solution of the many-electron schrödinger equation with deep neural networks. *Physical Review Research*, 2(3):033429, 2020.

[9] Dan Shepherd and Michael J Bremner. Temporally unstructured quantum computation. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 465(2105):1413–1439, 2009.

[10] James Stokes, Josh Izaac, Nathan Killoran, and Giuseppe Carleo. Quantum natural gradient. *Quantum*, 4:269, 2020.

[11] Xin Wang, Zhixin Song, and Youle Wang. Variational quantum singular value decomposition. *arXiv preprint arXiv:2006.02336*, 2020.

[12] Xiaosi Xu, Jinzhao Sun, Suguru Endo, Ying Li, Simon C Benjamin, and Xiao Yuan. Variational algorithms for linear algebra. *arXiv preprint arXiv:1909.03898*, 2019.

[13] Tianchen Zhao, Giuseppe Carleo, James Stokes, and Shravan Veerapaneni. Natural evolution strategies and variational monte carlo. *Machine Learning: Science and Technology*, 2(2):02LT01, 2020.

Quantum secure learning with classical samples

Wooyeong Song,[1, *] Youngrong Lim,[2, *] Hyukjoon Kwon,[3] Gerardo Adesso,[4]
Marcin Wieśniak,[5, 6] Marcin Pawłowski,[6] Jaewan Kim,[2] and Jeongho Bang[7, †]

[1] Center for Quantum Information, Korea Institute of Science and Technology (KIST), Seoul 02792, Korea
[2] School of Computational Science, Korea Institute for Advanced Study, Seoul 02455, Korea
[3] QOLS, Blackett Laboratory, Imperial College London,
London SW7 2AZ, England, United kingdom
[4] School of Mathematical Science and Centre for the Mathematics
and Theoretical Physics of Quantum Non-Equilibrium Systems,
University of Nottingham, University Park,
Nottingham NG7 2RD, England, United Kingdom
[5] Institute of Theoretical Physics and Astrophysics, Faculty of Mathematics,
Physics and Informatics, University of Gdanńsk, Poland
[6] International Centre for Theory of Quantum Technologies, University of Gdanńsk, Poland
[7] Electronics and Telecommunications Research Institute, Daejeon 34129, Korea

We present a sampling protocol and a security condition that allows only legitimate learners to prepare a sample set that guarantees the successful learning. By combining our security concept with the sample bound in machine learning framework, we derive a sample upper bound that rule out adversarial learners.

I. INTRODUCTION

The hybridization of machine learning and quantum theory has been intensively studied, especially to explore the possibility of exploiting quantum learning speedups [1–4]. In parallel, the issue of security has been of considerable interest to the machine learning community. The term secure learning is usually used to indicate that the learning is allowed only for the legitimate learner, who wants to rule out adversarial learners. The main objective of these adversaries is to acquire ability to become equals of the legitimate learner or to render the learning of the legitimate learner counterproductive. In this context, one of the open issues is how to define a secure learning condition for detecting and preventing these adversaries. While this problem has been widely studied in classical learning [5, 6], only a few quantum-mechanical studies have been conducted so far [7–10].

II. MAIN AND RESULTS

In this work, we first design a protocol for secure sampling that runs between two legitimate learning parties. These legitimate learning parties are consist of a legitimate learner, Alice, who has learning input ($\mathbf{x}$) and Bob who has an oracle which receives input $\mathbf{x}$ and outputs its corresponding label ($c(\mathbf{x})$) to build a set of learning sample $\{(\mathbf{x}, c(\mathbf{x}))\}$. The secure sampling protocol provide a way to detect the existence of an adversarial learner (Eve) who contaminates and/or eavesdrops the sampling process. Then, we suggest a secure learning condition in the form of learning sample bound which is motivated by probably approximately correct (PAC) learning framework in machine learning. Together with the secure learning condition and sampling protocol, it is assure that

[*] The first two authors contributed equally to this work
[†] Electronic address: jbang@etri.re.kr

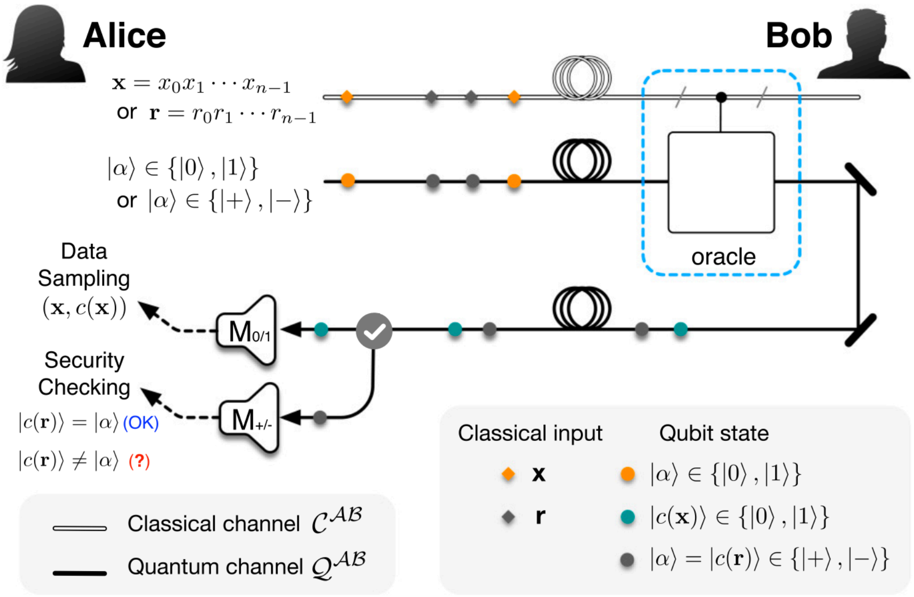

Alice
Bob
$\mathbf{x} = x_0 x_1 \cdots x_{n-1}$
or $\mathbf{r} = r_0 r_1 \cdots r_{n-1}$
$|\alpha\rangle \in \{|0\rangle, |1\rangle\}$
or $|\alpha\rangle \in \{|+\rangle, |-\rangle\}$
oracle
Data Sampling
$(\mathbf{x}, c(\mathbf{x}))$
$M_{0/1}$
Security Checking
$M_{+/-}$
$|c(\mathbf{r})\rangle = |\alpha\rangle$ (OK)
$|c(\mathbf{r})\rangle \neq |\alpha\rangle$ (?)
Classical channel $\mathcal{C}^{\mathcal{AB}}$
Quantum channel $\mathcal{Q}^{\mathcal{AB}}$
Classical input
Qubit state
◆ x
◆ r
● $|\alpha\rangle \in \{|0\rangle, |1\rangle\}$
● $|c(\mathbf{x})\rangle \in \{|0\rangle, |1\rangle\}$
● $|\alpha\rangle = |c(\mathbf{r})\rangle \in \{|+\rangle, |-\rangle\}$

FIG. 1: Schematic of our sampling protocol. Alice has facilities for the preparation of inputs, ($\mathbf{x}$, $|\alpha\rangle \in \{|0\rangle, |1\rangle\}$) or ($\mathbf{r}$, $|\alpha\rangle \in \{|+\rangle, |-\rangle\}$). Alice can also perform a single-qubit measurement to identify the returning qubit. Bob owns the oracle. Here, we consider a classical-quantum hybrid architecture with a classical input ($\mathbf{x}$ or $\mathbf{r}$) and an ancillary qubit state ($|\alpha\rangle$). The oracle does not reveal its structure.

the legitimate learner can (**1**) complete a learning with a targeted performance and (**2**) prevent adversarial learner to become a learner with same ability [11].

III. REMARKS

We present a concept of secure learning that safeguards against any malicious manipulation of learning samples. This allows us to establish the link between learning sample complexity and the condition for machine learning security. Our approach is appealing because the security condition is defined solely by the size of sample set.

[1] P. Rebentrost, M. Mohseni, and S. Lloyd, Phys. Rev. Lett. **113**, 130503 (2014).
[2] S. Lloyd, M. Mohseni, and P. Rebentrost, Nat. Phys. **10**, 631 (2014).
[3] M. Schuld, I. Sinayskiy, and F. Petruccione, Phys. Rev. A **94**, 022342 (2016).
[4] I. Kerenidis and A. Prakash, arXiv:1603.08675
[5] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, Mach. Learn. **81**, 121 (2010).
[6] B. Nelson, B. I. P. Rubinstein, L. Huang, A. D. Josheph, S. J. Lee, S. Rao, and J. D. Tygar, J. Mach. Learn. Res. **13**, 1293 (2012).
[7] J. Bang, S. W. Lee and H. Jeong, Quantum Inf. Proc. **14**, 3933 (2015).
[8] Y. B. Sheng and L. Zhou, Sci. Bull. **62** 1025 (2017).
[9] N. Wiebe and R. S. S. Kumar, New J. Phys. **20**, 123019 (2018).
[10] N. Liu and P. Rebentrost, Phys. Rev. A **97**, 042315 (2018).
[11] W. Song, Y. Lim, H. Kwon, G. Adesso, M. Wieśniak, M. Pawłowski, J. Kim, and J. Bang, Phys. Rev. A **103**, 042409 (2021).

Quantum diffusion map for nonlinear dimensionality reduction

Apimuk Sornsaeng[*] Ninnat Dangniam[†] Pantita Palittapongarnpim[‡] Thiparat Chotibut[§]

Chula Intelligent and Complex Systems, Department of Physics,
Faculty of Science, Chulalongkorn University, Bangkok, Thailand, 10330

Abstract

Diffusion map is a class of dimensionality reduction algorithms inspired by random walks on networks. We propose a quantum algorithm that takes N classical data points as an input and constructs the classical diffusion map in expected time $N^2 \text{polylog}\, N$ compared to $O(N^3)$ for the classical algorithm.

1 Overview of the work

Dimensionality reduction is a class of unsupervised machine learning that offers an automatic identification of statistical structures hidden in a high-dimensional dataset. Diffusion map (DM) [1–3] is one of the *nonlinear* dimensionality reduction algorithms that exploits the properties of random walkers to navigate the proximity structures of data points embedded in networks or graphs. This algorithm has an advantage that the new diffusion coordinate preserves the notion of proximity between data points [1–3]. DM has been applied to a wide range of unsupervised machine learning tasks from the identification of quantum phase transitions in many-body systems [4–9] to data visualization in bioinformatics [10, 11]. DM techniques require the eigen-decomposition of a matrix whose size is $N \times N$, where N is the number of data points. Therefore, the computational cost grows with the number of data points as $O(N^3)$ which is prohibitive for a large data sample.

Recently, several quantum algorithms for unsupervised learning have been proposed, such as quantum principal component analysis (qPCA) [12] and quantum spectral clustering [13]. These algorithms are able to achieve a computational speedup in part due to the assumption of the qRAM data structure and the use of quantum phase estimation in performing the eigen-decomposition. With a quantum computational speedup for dimensionality reduction in mind, we propose a quantum algorithm for unsupervised manifold learning called *quantum diffusion map* (qDM). Our qDM algorithm coherently prepares all necessary components for constructing DM in time $O(\text{polylog}\, N)$ [14]. The complete algorithm that constructs the classical diffusion map in an end-to-end manner has an expected runtime of roughly $N^2 \text{polylog}\, N$ [14], as opposed to $O(N^3)$ in classical DM. The quadratic runtime N^2 is due to the final readout using tomography.

The next section provides the detail of classical diffusion map and the last section proposes the method for constructing quantum diffusion map.

2 Classical diffusion map

In DM, the input dataset defines the all-connected graph such that the data points $\{\boldsymbol{x}^{(i)}\}$ are nodes and weighted edges $\{W_{ij}\}$ of the graph are the Gaussian kernel

$$W_{ij} \equiv K_\sigma\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right) = \exp\left(-\frac{\left\|\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}\right\|_2^2}{2\sigma}\right), \tag{1}$$

[*] apimuk25@hotmail.com

[†] ninnatdn@gmail.com

[‡] panpalitta@gmail.com

[§] thiparat.c@gmail.com

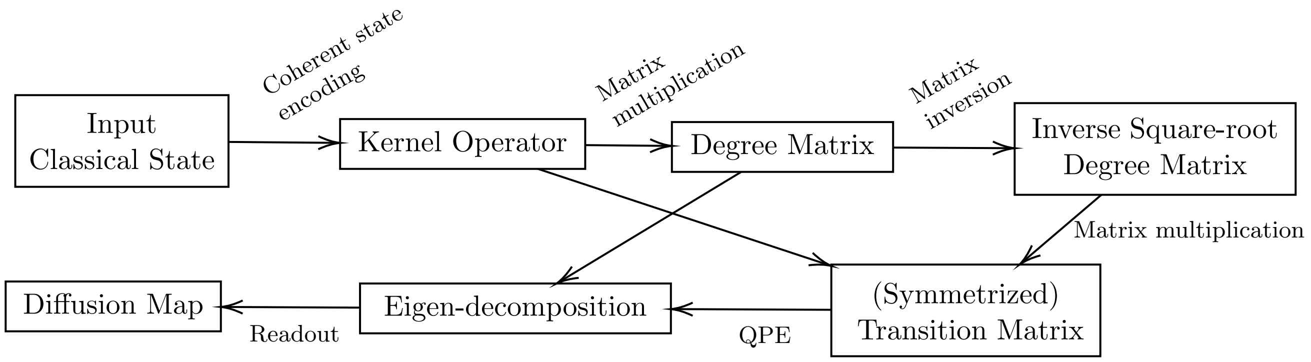


Figure 1: The detailed roadmap for our quantum diffusion map algorithm [14], which also enables an efficient construction of the transition matrix for generating random walk on graphs as a byproduct.

with the adjustable parameter σ to set the scale of data neighborhood. DM then assigns a discrete-time random walk on the graph given by the normalized weight, called the Markov transition probability from vertex i to j: $P_{ij} = W_{ij} / \sum_j W_{ij}$. The Markov transition probabilities can be expressed in matrix form as a product $P = D^{-1}K$, where the diagonal matrix D contains the degree of each node of the graph.

The *diffusion map* of a datum $\boldsymbol{x}^{(i)}$ is defined by the eigen-decomposition of the transition matrix:

$$\boldsymbol{\phi}_{t,m}\left(\boldsymbol{x}^{(i)}\right) = \left(\lambda_1^t v_1\left(\boldsymbol{x}^{(i)}\right), \cdots, \lambda_m^t v_m\left(\boldsymbol{x}^{(i)}\right)\right)^T, \tag{2}$$

where v_i is the *right* eigenvector of P with eigenvalue λ_i, $v_i\left(\boldsymbol{x}^{(j)}\right)$ denotes the j^{th} coordinate of v_i, and the eigenvalues are ordered such that $\lambda_0 = 1 > \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{N-1} \geq 0$. DM can be constructed classically in time $O(N^3)$ due to eigendecomposition of P, which is prohibitive for large-scale dataset.

3 Quantum diffusion map roadmap

Due to DM's prohibitive time complexity for large-scale data, we propose a quantum counterpart of DM called *quantum diffusion map* (qDM) which we summarize below and in Fig. 1.

1. First, with the assumption of qRAM, we construct the kernel matrix as a density matrix in time $O(\log N)$ [15]. In this encoding, each component of the kernel matrix is the inner product of quantum states that encode the classical data, which we choose to be *canonical coherent states* to obtain the Gaussian kernel. Note that this encoding *exactly* computes the Gaussian kernel (1) as opposed to that in [16] which approximately computes (more general) nonlinear kernels.

2. The Markov transition matrix is then constructed from the kernel matrix using the *quantum matrix algebra toolbox* (QMAT) [17], and *quantum matrix inversion* [18, 19], both of which require *density matrix exponentiation* [12] of the kernel matrix and relevant matrices. This key step allows us to construct the degree matrix of the graph by coherently performing row-summation of the kernel matrix, which distinguishes our algorithm from other graph-based quantum algorithms for dimensionality reduction such as [13, 16].

3. Finally, the eigen-decomposition of the transition matrix is carried out using *quantum phase estimation* (QPE).

Our algorithm performs the eigen-decomposition of the transition matrix in time $O(\text{polylog}\, N)$ [14], as opposed to $O(N^3)$ for the classical algorithm. This exponential speedup could be beneficial in achieving a quantum speedup other random walk-based algorithms. To construct the diffusion map classically, tomography is used to recover the eigenpairs of the transition matrix. The recovery process is nondeterministic and has an expected runtime of N^2 [14], giving the overall expected time complexity of $N^2\text{polylog}\, N$.

References

[1] R. R. Coifman and S. Lafon, “Diffusion maps,” *Appl. Comput. Harmon. Anal.*, vol. 21, no. 1, pp. 5–30, 2006.

[2] S. Lafon, *Diffusion maps and geometric harmonics.* PhD thesis, Yale University, 2004.

[3] B. Nadler, S. Lafon, I. Kevrekidis, and R. Coifman, “Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators,” in *Adv. Neural Inf. Process. Syst.* (Y. Weiss, B. Schölkopf, and J. Platt, eds.), vol. 18, MIT Press, 2006.

[4] J. F. Rodriguez-Nieva and M. S. Scheurer, “Identifying topological order through unsupervised machine learning,” *Nat. Phys.*, vol. 15, pp. 790–795, Aug 2019.

[5] Y. Long, J. Ren, and H. Chen, “Unsupervised manifold clustering of topological phononics,” *Phys. Rev. Lett.*, vol. 124, p. 185501, May 2020.

[6] A. Lidiak and Z. Gong, “Unsupervised machine learning of quantum phase transitions using diffusion maps,” *Phys. Rev. Lett.*, vol. 125, p. 225701, Nov 2020.

[7] A. Kerr, G. Jose, C. Riggert, and K. Mullen, “Automatic learning of topological phase boundaries,” *Phys. Rev. E*, vol. 103, p. 023310, Feb 2021.

[8] Y. Che, C. Gneiting, T. Liu, and F. Nori, “Topological quantum phase transitions retrieved through unsupervised machine learning,” *Phys. Rev. B*, vol. 102, p. 134213, Oct 2020.

[9] J. Wang, W. Zhang, T. Hua, and T.-C. Wei, “Unsupervised learning of topological phase transitions using the Calinski-Harabaz index,” *Phys. Rev. Research*, vol. 3, p. 013074, Jan 2021.

[10] K. R. Moon, D. van Dijk, Z. Wang, S. Gigante, D. B. Burkhardt, W. S. Chen, K. Yim, A. van den Elzen, M. J. Hirn, R. R. Coifman, N. B. Ivanova, G. Wolf, and S. Krishnaswamy, “Visualizing structure and transitions in high-dimensional biological data,” *Nat. Biotechnol.*, vol. 37, no. 12, pp. 1482–1492, 2019.

[11] C. L. Ecale Zhou, S. Malfatti, J. Kimbrel, C. Philipson, K. McNair, T. Hamilton, R. Edwards, and B. Souza, “multiPhATE: bioinformatics pipeline for functional annotation of phage isolates,” *Bioinformatics*, vol. 35, pp. 4402–4404, 05 2019.

[12] S. Lloyd, M. Mohseni, and P. Rebentrost, “Quantum principal component analysis,” *Nat. Phys.*, vol. 10, no. 9, pp. 631–633, 2014.

[13] I. Kerenidis and J. Landman, “Quantum spectral clustering,” *Phys. Rev. A*, vol. 103, p. 042415, Apr 2021.

[14] A. Sornsaeng, N. Dangniam, P. Palittapongarnpim, and T. Chotibut, “Quantum diffusion map for nonlinear dimensionality reduction,” *arXiv:2106.07302*, 2021.

[15] P. Rebentrost, M. Mohseni, and S. Lloyd, “Quantum support vector machine for big data classification,” *Phys. Rev. Lett.*, vol. 113, p. 130503, Sep 2014.

[16] Y. Li, R.-G. Zhou, R. Xu, W. Hu, and P. Fan, “Quantum algorithm for the nonlinear dimensionality reduction with arbitrary kernel,” *Quantum Science and Technology*, vol. 6, p. 014001, nov 2020.

[17] L. Zhao, Z. Zhao, P. Rebentrost, and J. Fitzsimons, “Compiling basic linear algebra subroutines for quantum computers,” *Quantum Mach. Intell.*, vol. 3, p. 21, jun 2021.

[18] A. W. Harrow, A. Hassidim, and S. Lloyd, “Quantum algorithm for linear systems of equations,” *Phys. Rev. Lett.*, vol. 103, no. 15, p. 150502, 2009.

[19] I. Cong and L. Duan, “Quantum discriminant analysis for dimensionality reduction and classification,” *New J. Phys.*, vol. 18, p. 073011, 2016.

On Sampling and Inference using Quantum Algorithms

S Ashutosh[†1], Deepankar Sarmah[†2], Sayantan Pramanik[⋆3] and M Girish Chandra[*4]

[†]*Indian Institute of Science Education and Research, Kolkata* [⋆]*TCS Incubation* [*]*TCS Research*

[1,2]{*sa17ms105, ds15ms006*}@*iiserkol.ac* [3,4]{*sayantan.pramanik, m.gchandra*}@*tcs.com*

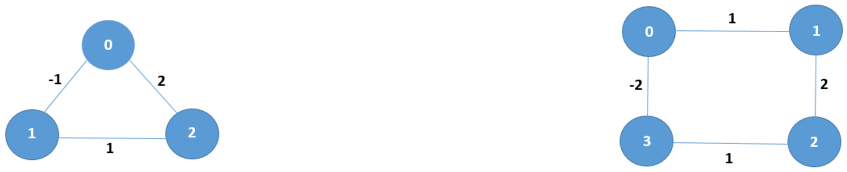
Abstract: Quantum computers are projected to handle the Gibbs sampling and the related inference on Markov networks effectively. We capture some observations obtained through extensive simulations with two popular paradigms - QA and QAOA.

1. Brief Description

In this work, we delve into the problem of Gibbs sampling and probabilistic inferencing from simple Markov networks at different temperatures using the methods of Quantum Annealing (QA) and Quantum Approximate Optimization Algorithm (QAOA). To sample from a thermal state, we allow the quantum annealer to interact with the environment and equilibrate. The samples from such a thermal state is expected to follow the Boltzmann distribution given by $P(E_i) = g(E_i)^{(e^{-E_i/kT})}/\mathcal{Z}$ where E_i represents the energy of the i^{th} state, $g(E_i)$ is the degeneracy of the energy level E_i, T is the effective temperature and $\mathcal{Z}$ represents the canonical partition function [1]. To sample from the Markov networks using QAOA, we employ the approach proposed by Verdon, et al. [2].

From the samples obtained from the above method, it was possible to construct the joint probability distribution of the variables represented by the nodes of the graph. The frequency of obtaining each sample should match the probability predicted by the equation above. This was verified by finding the Kullback-Liebler (KL) divergence of both the distributions, which indeed went to zero as the number of shots was increased. Further, we could marginalize over some of the variables and find the distribution of the remaining, find the Maximum A Posteriori (MAP) estimation for one or more variables, and also clamp over some of the variables and check the effect of such clamping over the other variables [3, 4].

2. Results and Discussion

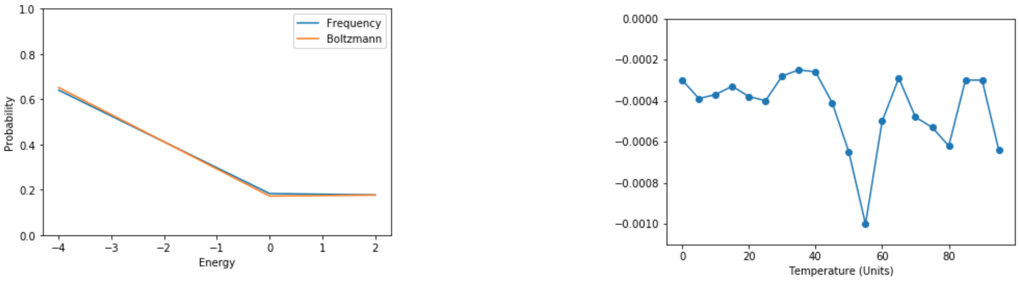


(a)

(b)

Fig. 1: Simple Markov networks for sampling and inferencing.

Few typical results and remarks from our elaborate simulation studies with different networks and clamping scenarios are recorded in this section. Figure 2a shows the frequency of the sampled states using QA at a low temperature against their energy, along with the Boltzmann distribution for comaprison. The similarity of the two distributions has been portrayed in Figure 2b through their KL divergence metric. The results of joint probability distribution, marginalization and MAP estimate are captured in Tables 1 and 2, respectively, and the marginalized distribution in Figure 3a. One of the nodes of the graph in Figure 1b was clamped to a spin state, and Figure 3b shows the effects of the clamping on the spin states of the other nodes. Here, it must be mentioned that the samples obtained using QAOA at low temperatures were markedly different from the Boltzmann distribution, while no such problem was encountered at higher temperatures; this limitation can be overcome by considering other ways of sampling. Finally, needless to say, sampling and probabilistic inference play a vital role in diverse applications. Detailed explanation and more results are available in [5].



(a)

(b)

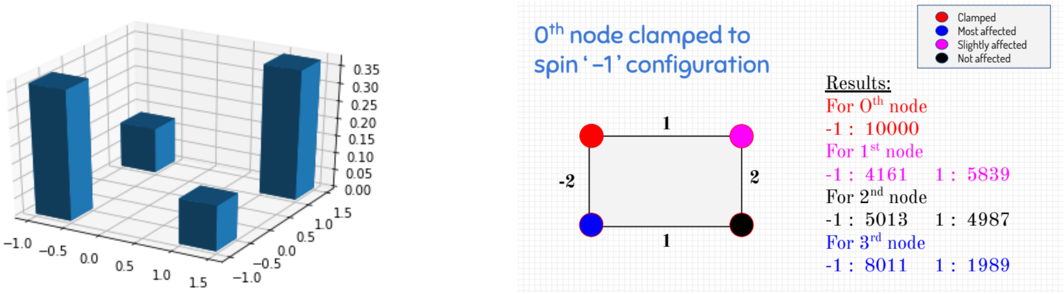
Fig. 2: (a) shows the results of sampling compared with Boltzmann distributiton, (b) shows the variation of KL divergence between frequency and Boltzmann distribution with temperature.

Table 1: Joint Probability Distribution for Figure 1a

Configuration	Probability
-1,-1,-1	0.04410419473467003
-1,-1,1	0.3258883690926387
⋮	⋮

Table 2: Results of marginalization and maximum a posteriori estimate over the last variable in Figure 1a

Configuration	Marginalized distribution	MAP estimate
-1,-1	0.3699925638273087	1
-1,1	0.13000743617269125	1
⋮	⋮	⋮



(a)

(b)

Fig. 3: (a) Distribution on marginalizing over graph in Figure 1b, effects of clamping one node on the others in (b).

References

1. D. Poulin, P. Wocjan, "Sampling from the thermal quantum Gibbs state and evaluating partition functions with a quantum computer," Phys. Rev. Lett. 103 220502 (2009).
2. G. Verdon, M. Broughton, J. Biamonte, "A quantum algorithm to train neural networks using low-depth circuits," arXiv:1712.05304v2.
3. T. Babej, "Quantum-enhanced sampling for probabilistic inference in undirected graphical models," Spring, 2018.
4. P. Wittek, "Quantum machine learning, edX course notes," University of Toronto, 2019.
5. S. Ashutosh, D. Sarmah, S. Pramanik, M.G. Chandra, "On sampling and inference using quantum algorithms," arXiv:2006.11775.

Influence of Molecular Orbital Basis Sets on the Accuracy and Efficiency of Bond Dissociation Energy by Variational Quantum Eigensolver (VQE)

Shampa Sarkar, M. R. Nirmal, M. Girish Chandra, Manoj Nambiar
TCS Research, Tata Consultancy Services Limited
shampa.sarkar@tcs.com, mr.nirmal@tcs.com, m.gchandra@tcs.com, m.nambiar@tcs.com

Abstract: Various molecular orbital basis sets have been employed to calculate potential energy surface and bond dissociation energy for H_2, LiH, using VQE. The accuracy improved with larger basis sets, subject to retaining higher order orbitals in computation.

1. Introduction

Computational Chemistry employs molecular orbital (MO) or a linear combination of atomic orbitals as basis sets, which comprise a set of algebraic functions to describe the spatial distribution of electronic wave function, as well as their spin configuration[1]. Designing of or selecting a molecular orbital basis set is a complex problem, as the flexibility of the basis set must inherently accommodate all the correlated electronic orbitals and the spatial and spin representation accurately for computation, yet be computationally efficient[2]. A larger MO basis set often provides a more accurate potential energy surface (PES) simulation of a molecule, though with significant increase in computational resources. As today's NISQ (Noisy Intermediate Scale Quantum) devices or quantum simulators are limited by qubit counts and quantum circuit depth, quantum chemistry simulations for small, diatomic molecules often employ the minimal basis sets of Gaussian orbitals, viz. STO-kG (k = 3, 6, etc.), which approximates each STO (Slater type Orbital) with k GTOs (Gaussian type Orbital) functions in the least squares. (For e.g., in STO-3G basis, each STO is represented by a linear combination of three Gaussian functions, often their exponents are constrained to be the same for all orbitals belonging to the same shell.). Such contracted GTO produce reasonable computation for an atomic orbital, however, falls on accuracy for molecular computation. More realistic basis sets with Gaussian functions comprise split-valence basis set with double zeta functions on the valence orbital, some of which are 3-21G or 6-31G or 6-31G* (also called the Pople basis set). Here, for X-YZG, X denotes the number of primitive Gaussians (G) for each core atomic orbital basis function, whereas Y and Z indicate two sets of primitive Gaussian basis functions (G) for each of the valence orbitals. Although these basis sets may promise higher accuracy, they require increasing number of qubits. For the post Hartree-Fock wavefunction based calculations, classically the most widely used basis sets are 'correlation-consistent' basis sets, termed as cc-pVNZ, where N = D, T, Q for double, triple or quadrupole zeta functions, which have even higher computational cost. Recent VQE computations also considered intrinsic atomic orbital in quantum simulation of molecules in self-consistent field manner, without incorporating electron correlation in calculation[3].

2. Quantum Simulation approach

We have used quantum simulators to perform Variational quantum eigensolver (VQE) to calculate the potential energy surface and the bond dissociation energy for small molecules, for a post Hartree-Fock Hamiltonian with the Born-Oppenheimer (BO) approximation, as a function of various molecular orbital basis sets. The basis sets span from minimal basis sets of STO-3G, STO-6G, MINAO, to larger split-valence basis sets of 3-21G, 6-31G and 6-31G*. The experiments are performed on IBM Quantum simulator in a hybrid quantum-classical computing environment, using L_BFGS_B (Limited-memory BFGS Bound) classical optimizer with maximum iteration counts as 500. We have used unitary coupled cluster single and double (UCCSD) as the trial state ansatz. Also employed is the Parity mapping to convert the electronic Hamiltonian to qubit Hamiltonian and have further used Z2 symmetry operators, to minimize the qubit counts for computation.

3. Results and Discussion

For H2, STO-3G, STO-6G and MINAO MO basis sets have been used, which required four qubits, while eight qubits were required for simulating using larger MO Pople basis sets viz., 3-21G, 6-31G and 6-31G*. With Parity mapping and subsequent two qubit reductions, the number of qubits employed for simulation of H_2 molecule have been reduced to two and six, respectively. The PES, as displayed in Fig 1(a) and the bond dissociation energy for H_2

molecule in Fig. 1(c) demonstrated higher accuracy for 6-31G* basis set, retaining all higher order orbitals for computation.

For LiH, the VQE simulation have produced the ground state energy minimum at the same inter-nuclear distance, as that of the experimental equilibrium inter-nuclear distance of 1.595 angstroms. For the comparison of the VQE results as a function of various MO basis sets, we have further frozen the core 1s electrons of Li atom and have removed the $2p_x$ and $2p_y$ molecular orbitals for Li atom, to reduce the qubit counts for computational efficiency. The number of active interacting molecular orbitals could be brought down from twelve to six for STO-3G and STO-6G bases and from twenty to fourteen for 3-21G and 6-31G bases. With Parity mapping and incorporation of Z2-symmetries, the number of qubits has further been reduced to four and twelve, respectively. The results are plotted in Figure 1(b), which showed more accurate minimum ground state energy value obtained by using 6-31G basis. However, the computed bond dissociation energies, which were calculated by subtracting the Ground State energies of 1.595 angstroms from that of 4 angstroms showed more accurate values for the STO-6G basis set. The results clearly demonstrated that the removal of the $2p_x$ and $2p_y$ orbital have reduced the accuracy of the simulated bond dissociation values. We have further performed simulation using STO-3G and STO-6G bases, without removing $2p_x$ and $2p_y$ orbitals (only with freezing the Li core 1s orbitals), which produced more accurate bond dissociation energy of 0.098 and 0.09935 Hartree respectively, in comparison with the experimental value of 0.0924[4] Hartree. These results clearly demonstrate that although larger basis sets like 3-21G, 6-31G or 6-31G* give better accuracy (as seen in the case of H_2), the results are subject to retaining higher order orbitals in the computation (as seen in LiH), which in turn require larger qubit counts and circuit depth, limited by computational efficiency of today's NISQ devices.

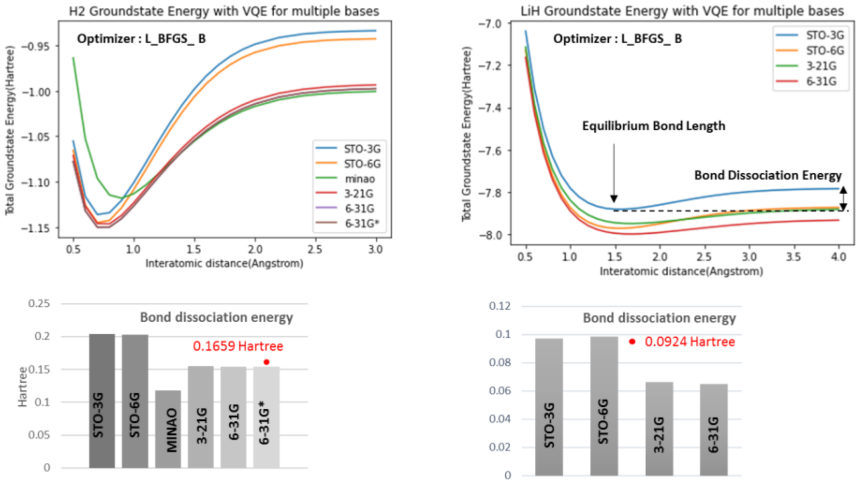


Fig. 1. PES and Bond dissociation energy of H_2 and LiH, as a function of molecular bases.

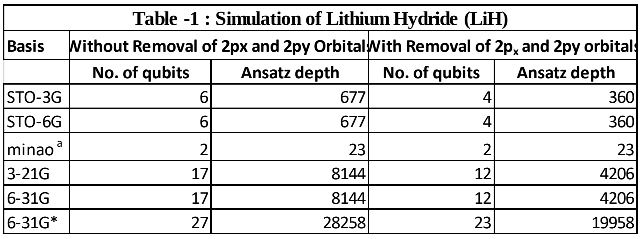


Table -1 : Simulation of Lithium Hydride (LiH)

Basis	Without Removal of 2px and 2py Orbital		With Removal of $2p_x$ and 2py orbital	
	No. of qubits	Ansatz depth	No. of qubits	Ansatz depth
STO-3G	6	677	4	360
STO-6G	6	677	4	360
minao [a]	2	23	2	23
3-21G	17	8144	12	4206
6-31G	17	8144	12	4206
6-31G*	27	28258	23	19958

4. References

[1] T. Helgaker, P. Jorgensen, J. Olsen, "Molecular Electronic Structure Theory," Wiley (2014).

[2] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, "Quantum computational chemistry," Rev. Mod. Phys. **92**, 015003 (2020).

[3] S. Barison, D. E. Galli, M. Motta, "Quantum simulation of molecular systems with intrinsic atomic orbitals", arxiv.org/pdf/2011.08137

[4] W-C Tung, M. Pavanello and L. Adamowicz, "Very accurate potential energy curve of the LiH molecule," J. Chem. Phys. **134**, 064117 (2011).