Response to Reviewers

ICLR 2022
Where Can Quantum Kernel Methods Make A Big Difference?

Reviewer 1 (Reviewer kXVf)

Comment 1: Last sentence of 2nd paragraph is unclear.
Response 1: Thanks for your suggestion. Yes, the last sentence of 2nd paragraph is not clear enough and lacks some details. What I want to express is that recently many quantum machine learning algorithms are not limited to theory and many researchers get good results through their experiments. I have listed many articles by others here, which I will not discuss in detail as they are not the subject matter of the article. I will modify the sentence as follow: For example, [1], [2], [3], [4], [5], and [6] have demonstrated the value of quantum machine learning in different machine learning tasks by specific experiments they designed, respectively.

Comment 2: 1st sentence of 3rd paragraph in unclear: kernel methods (and not the kernel method) are a family of algorithms.
Response 2: Thanks for your reminder. I agree that kernel methods are a family of algorithms. So this sentence will be modified as follow: On the other hand, the quantum kernel methods are well-known machine learning algorithms.

Comment 3: Kernel methods become (last word of p.1).
Response 3: Yes, the past tense is not needed here because the verb before is made. The sentence should be: "... made quantum kernel methods become ...".

Comment 4: We conjecture (1st contribution bullet point).
Response 4: Thanks for your advice. Yes, the word conjecture is more suitable here because it is not a method but a conjecture. The sentence should be: We conjecture that the quantum kernel function is probabilistic and classify the existing kernel functions.

Comment 5: Please cite the SVM paper when introducing them (1st par. of sec. 2).
Response 5: Thanks for the viewer's advice. When we use a professional term, we should cite it no matter how familiar we are with it. The sentence is changed to: One of the most famous methods is the support vector machine (SVM) proposed by [7].

Comment 6: Kernel trick (2nd par. sec. 2).
Response 6: Thanks. It is a typo. It should be kernel trick.

Comment 7: SVM paragraph: remove kind of.
Response 7: Thanks for the reviewer's suggestion. To make the meaning simple and clear, I will remove kind of. The sentence becomes: Support Vector Machine is a maximal margin classifier.

Comment 8: Eq. (4), a T is missing between the ϕ (also support vectors are never defined).
Response 8: Thanks for your reminder, we want to express the inner product of two functions so it should be $\phi(s_i)^T\phi(x)$ in Eq.(4). Also, we should define the support vectors since they appear in our paper. We will add a definition of the support vectors: The samples closest to the separating hyperplanes are those whose coefficients a_i are nonzero. These samples are called support vectors.

Comment 9: Top of page 4: there is a very clear theoretical answer about what kernel is a valid or not. The good performances of the sigmoid kernel (reference needed here) does not change anything.
Response 9: Thanks for reminding us of this point. I agree that there is a very clear theoretical answer about what kernel is valid or not. In our paper, we did not express the meaning clearly. What we want to express is that we don't have a theoretical standard to determine whether a kernel can be effectively used or not. Because even a kernel that is not valid such as a sigmoid kernel has a good performance as well. Also, we should add a reference for the sigmoid kernel function here since it appears in our paper. The sigmoid kernel was proposed by [8].

Comment 10: I cannot understand what authors mean by that expands the

family of kernel functions.
Response 10: Thanks for the time and efforts to review our work. What we plan to do is to classify the family of kernel functions from a probabilistic point of view, not to expand the family of kernel functions. The purpose of it is to provide a better understanding of different kernel functions. Because there is no detailed classification at present, especially when quantum kernel functions are proposed recently. What's more, some people may feel confused when they first meet the quantum kernel function, so we just want to express that the quantum kernel is similar to the classical kernel but based on a different mechanism.

Comment 11: Fig.2 seems to imply that tree and graph kernels do not satisfy Mercer's conditions: why such a separation? Also, the ... boxes are not relevant, and "Don't satisfy..." would be a better legend.
Response 11: Thanks for your advice. Fig.2 is a picture of the kernel category. There are two levels. The first level of classifying is by the data structure. For example, vector, tree, graph, and so on. The second level of classifying is two binary classifications, i.e., (1) probability/deterministic and (2) satisfy/don't satisfy Mercer's condition. In our paper, we only apply the second level of classifying to the vector kernels because the vector kernels are the kernels that we usually use.

Comment 12: The Mersenne Twister distribution could be introduced and commented before the theorem statement.
Response 12: Thank you for the helpful suggestion. The Mersenne Twister distribution should be introduced before the theorem statement since it is a terminology. We will add a introduction of Mersenne Twister distribution as follow: The Mersenne Twister is a pseudo random number generator which was first proposed by [9]. The Mersenne Twister is used as default pseudo random number generator by many software, such as Python, R, and PHP. The Mersenne Twister random distribution is a distribution that be generated by the Mersenne Twister method.

Comment 13: Eq. (5): I assume N_l, N_m instead of N_1,N_2.
Response 13: Yes, It is a typo. They should be N_l and N_m.

Comment 14: The "number of observations belonging to class X" would be more clear than the size.

Response 14: We agree with this point. "Number of observations belonging to class X" is far more clear than "size". So the corresponding sentence is modified as: "N_l and N_m are the number of observations belonging to class C_l and C_m".

Comment 15: The $\rightarrow$ vector notation is introduced once and is not consistent with the rest of the paper.
Response 15: In our paper, we use the notation $\rightarrow$ to show a mapping process. For example, $f_c : x_i \rightarrow \phi(x_i)$ shows that the mapping function f_c maps a data x_i into a higher dimensional feature space where x_i is represented by $\phi(x_i)$. Similarly, $f_q : x_i \rightarrow |\phi(x_i)\rangle$ shows that the mapping function f_q maps a data x_i into a quantum feature space where x_i is represented by $|\phi(x_i)\rangle$.

Comment 16: Expliciting the formulas in Euq.(7) is not necessary.
Response 16: Thanks for pointing out this point. Euq.(7) is tedious and we will keep the first half, which will become to: $\delta_{lm} = \frac{D(C_l,C_m)}{D(C_l)+D(C_m)}$.

Comment 17: Weird x notation.
Response 17: Thank you for the reminder. We make two changes. First, we change the x to x_i in the 8th line, 2nd paragraph, section 2. It is a typo. Then, to make consistency we modify the Equ.(1), replacing the x with x_i: $|\phi(x_i)\rangle = U(x_i)\,|0^n\rangle$. Second, to make the formula about the circuit U clearer, we replace the x with the vector $\vec{x}$ in the 4th line at top of page 3: $U(\vec{x}) = U_{\phi(\vec{x})}H^{\otimes 2}U_{\phi(\vec{x})}H^{\otimes 2}$.

Comment 18: f_c is mapping x to $\phi(x)$?!
Response 18: As we mentioned in our paper, f_c is a mapping function which maps data from a point in the original input space $\mathcal{O}$ to a higher-dimensional Hilbert feature space $\mathcal{F}_c$. We use $\phi(x)$ to represent the data x_i in such a higher-dimensional Hilbert feature space.

Comment 19: The dot product denoted with both T and .
Response 19: Thanks for your reminder, it is a typo. It should be: $\phi(x_i)^T\phi(x_j)$.

Comment 20: Note also that to compute a distance, one would use $||\phi(x) - \phi(y)||$ rather that $< \phi(x), \phi(y) >$ as claimed by the authors.
Response 20: Thanks for your note. We assume that what you mean is

$\langle\phi(x)|\phi(y)\rangle$ rather than $<\phi(x),\phi(y)>$ because there is no $<\phi(x),\phi(y)>$ appears in our paper. In our paper, we focus on the inner product. The inner product between u and v can be interpreted as projecting u onto v (or vice-versa), and then taking the product of the projected length of u ($|u|$) with the length of v ($|v|$). We can see the inner product as a measurement of similarity of two vectors u and v. Mathematically, the similarity is a distance in the data space. In quantum mechanics, we use the Dirac notation $|\cdot\rangle$ to represent a vector. The $\langle\phi(x)|\phi(y)\rangle$ is the inner product of two vector $|\phi(x)\rangle$ and $|\phi(y)\rangle$. As the inner product is a kind of measurement of similarity and the similarity is a mathematical distance, we use $\langle\phi(x)|\phi(y)\rangle$ to represent such a distance in quantum space.

Comment 21: In the quantum paragraph, I assume F_q means f_q
Response 21: Yes, it is a typo. In the 5th line of the quantum paragraph, the mapping function is f_q rather than F_q.

Comment 22: It seems to me that the definition of f_q from O to F_q is repeated. Furthermore, I cannot see any difference from the standard kernel definition
Response 22: Thanks for the time to view our work. They indeed have similar mechanisms. The f_q is a mapping function that can map a data point from the original space O to a quantum space F_q. The f_c is a mapping function that can map a data point from the original space O to a high-dimensional feature space F_c. The quantum space F_q is different from a high-dimensional feature space F_c. For the purpose of clearly showing this point and avoiding misunderstanding, we do not think it is repeated. Only the original space O is the same.

Comment 23: f_q is mapping x_i to a quantity that depends on x_j?!
Response 23: Thanks for your question. It is our mistake to make it confused. The quantity will only depend on x_i. We corrected the representation in the 9th line in quantum kernel paragraph as follow: $f_q : x_i \rightarrow |\phi(x_i)\rangle$.

Comment 24: The $|.>$ notation is very confusing for people used to kernel dot products. Even if it is the standard notation in the quantum literature, why don't we have two $|$ in the definition of f_q ?
Response 24: Yes, it is the standard notation in the quantum literature. It is called the bra–ket notation or Dirac notation. For example, a ket looks

like $|v\rangle$. Mathematically it denotes a vector, and physically it represents a state of some quantum system. To keep both the mathematical meaning and physical meaning, we use the standard notation in our paper.

Comment 25: The circuit U is never defined.
Response 25: Thanks for the reminder. A quantum kernel function are be realized by the corresponding quantum circuit. The quantum gates are the basic element for a quantum circuit, just like a logic gate for a digit circuit. Even though our main work is not to design a quantum circuit, we make some introductions in our paper. For example, in the "Quantum Kernel Method Based On Pauli Feature Map" paragraph, we introduced the mathematics of the quantum circuit behind the Z-ZZ quantum kernel method. In Fig.1(B) we showed the circuits which are printed by the IBM quantum computing platform.

Comment 26: How can the proof of Theorem 3.1 begin by "we assume" ?! If you have an extra assumption, just put it in the theorem statement.
Response 26: Thanks for this comment. Some researchers already proved that the quantum kernels can be superior to classical kernels when learning a DLP problem. Also, we know learning a Mersenne Twister distribution is a DLP problem. That's why we assume that the *Z-ZZ* feature map can effectively simulate the efficacy of the feature map proposed by [10] in the beginning. However, we cannot provide a rigorous mathematical proof of how the *Z-ZZ* feature map can effectively simulate the efficacy of the feature map proposed by [10] at present. We will continue to focus and work on this in the future.

Comment 27: The same goes for the proof of Theorem 3.2.
Response 27: Thanks for this comment. We will do more research and continue to improve it.

Comment 28: The exact statement of Theorem 3.2 should be "there exists δ_0 such that..."
Response 28: Thank you for the helpful advice. We will modify the expression of Theorem 3.2. It is going to be: In the case of a balanced number of the two classes, there exists δ_0 such that the quantum kernel method will not be better than the classical kernel method in handling classification problems when $\delta > \delta_0$. In practice, δ_0 is usually taken as 0.6.

Comment 29: The CLT does not imply the existence of a R such that the distribution is the standard Gaussian (what if the empirical mean is 1/n for instance?)
Response 29: Thanks for this comment. Our expression may confuse some readers and we will reconsider this theorem. As far as I know, although the actual distribution of the data varies, as the sample size increases, the mean of the sample will be close to the overall mean.

Comment 30: What are the "measurements" referring to?
Response 30: Thanks for your reminder. We should ass some introduction for the measurements in a quantum system. In quantum physics, a measurement is a test or manipulation of a physical system to produce a numerical result. The predictions made by quantum physics are generally probabilistic.

Comment 31: Where do σ and μ play a role?
Response 31: Thanks for this question. In the theorem, σ is the mean and the μ is the variance of the total measurements. Just for the convenience of expression.

Comment 32: Please prove "it is easy for a deterministic kernel [...] with small error".
Response 32: Thanks for this suggestion. We get this idea from empirical experience. For a binary classification problem, when the inner-class distance is large enough and the intra-class distance is small enough, the two classes can easily be separated. For example, a classical linear classifier will have a good performance in this case.

Comment 33: Theorem 3.1 only states that quantum kernels are superior than standard kernels on this example because the latter cannot learn anything. Can we prove that quantum can learn something in this case, i.e., are strictly better. Otherwise the statement is pretty vacuous.
Response 33: Thanks for this comment, and we will think about it carefully. Theorem 3.1 shows that quantum kernels are superior to classical kernels when meeting a random distribution based on Mersenne Twister Generator. To show this point, we designed an experiment, and the results are shown in Fig.4(B). The results show that the quantum kernels are almost always better than the classical kernels. As the number of data increases, the quan-

tum kernels will maintain a stable advantage over classical kernels. We think it is an interesting phenomenon. Some researchers already proved that the quantum kernels can be superior to classical kernels when learning a DLP problem. Also, we know learning a Mersenne Twister distribution is a DLP problem. That's why we assume that the *Z-ZZ* feature map can effectively simulate the efficacy of the feature map proposed by [10] in the beginning. However, we cannot provide a rigorous mathematical proof at present. We will continue to focus and work on this in the future.

Comment 34: The definition of δ is based on empirical quantities, is it suitable? Shouldn't a quantity in expectation make more sense?
Response 34: Thank you for this suggestion. In our paper we try to provide a threshold to decide which one is better to use a quantum kernel method or a classical kernel method. This threshold δ_0 is a empirical quantity that be determined through several datasets. It is not a average value or an expectation, but a threshold based on our observation. The "true" value of δ_0 is unknown, since it is a value determined by as many datasets as possible.We cannot try all the datasets to make sure of this. But, the existence of δ still make sense. At least, we know a phenomenon that the variable δ can has some influence to decide whether a quantum kernel method is better or not when compared with a classical kernel method. Based on our experience in 81 datasets, the δ_0 will take 0.6.

Comment 35: What happens for kernels defined on non-vectorial inputs? Do they have a quantum analog? It should be discussed.
Response 35: Thanks for the interesting questions, and we will think about them carefully. To start with, we think it is a very interesting topic to discuss. We list several formats of kernels in Fig 2 though we only discussed the vector kernels in our paper. Take the graph kernel as an example. The idea of graph kernel is to map a graph to some Hilbert space, and the similarity between two graphs can be obtained by the inner product operation in Hilbert space. As far as we can embed a graph into a vector format and use the inner product to represent the similarity, we can apply the quantum kernels methods to it.

Comment 36: Can we use another distance than the Euclidean one?
Response 36: Thanks for this comment. Definitely, we can. There are many ways to represent a distance, and the Euclidean distance is a popular one.

We only try the Euclidean distance to calculate the inner-class distance and the intra-class distance in our paper. However, we can also try another distance to do some experiments.

Comment 37: If we add an offset to the distance, it seems that the 0.6 threshold exhibited changes, making me dubious about such an absolute value.
Response 37: We appreciate it for pointing it out. Let me make it clear. We assume what you mean is adding some offset to the distance in the dataset. Yes, the δ of this dataset will be changed because the δ is a value related to distances. However, the threshold δ_0 will not be changed. The δ_0 is not a value that is determined by a specific dataset, but an empirical value that is determined by several datasets (In our experiment, 81 datasets). For example, assume the δ of a dataset D is 0.55 at first. Based on our theorem, the quantum kernels will be superior to the classical kernels, since the $0.55 < \delta_0 = 0.6$. Then, we add an offset to the distance, and the δ of the dataset D will be changed, for example, 0.7. Since $0.7 > \delta_0 = 0.6$, in the new case using a classical kernel will be a better choice.

Comment 38: Theorem 3.2 actually only shows that quantum kernels are worse than standard ones in a precise regime, not that they are better in the opposite scenario.
Response 38: Yes, it is. That's why we make the name "Deficiencies of the QKM". We think this work is still worthwhile. Suppose we get the δ of a specific dataset D, if the δ for the D is larger than δ_0, we have enough reason to believe that we can use classical kernels to learn this dataset. On the other hand, suppose the δ for the D is less or equal to δ_0. Even though we cannot directly say that the quantum kernels will be better, it at least provides us a choice to use quantum kernels. Whether the quantum kernels will be better depends on the data pattern. For example, as we mentioned in the paper, if we meet the Mersenne Twister random distribution, the quantum kernels will be superior.

Reviewer 2 (Reviewer DX2D)

Comment 1: The introduction of the article is well-written and reflects a good knowledge of the literature. However, the authors restrict the comparison between quantum kernel methods and classical kernel methods to the case of binary classification. Moreover, the theoretical results are presented

informally.
Response 1: Thanks for the time and efforts to review our work. Yes, we only restrict the comparison to the case of binary classification. The binary classification problem is the basic classification problem. Any multi-class classification problem can be divided into binary classification problem. There are two methods to transfer a multi-class classification problem into a binary classification problem, i.e., "one vs one" and "one vs rest". We will continue to do expand the content of our paper. What's more, the theoretical results are almost based on our experiments. The threshold δ_0 is a empirical value based on our experiments. We will try our best to make the results more formally.

Comment 2: The analysis concerns binary classification, which is one among many learning tasks to study. It is not clear how to deal with other learning tasks.
Response 2: Thanks for your constructive comments. In this paper, we only focus on a very common problem, the binary classification problem. For machine learning tasks, the two most popular tasks are classification and regression. Classification tasks include binary classification problems and multi-class classification problems. Multi-class classification problems can be transferred to binary classification problems using "one vs one" or "one vs rest" methods. As far as we know, there exist some quantum kernel methods that can deal with regression tasks, with a similar quantum mechanism. In our paper, we would like to reveal some laws to evaluate quantum kernel methods and classical kernel methods. So, we just start from a basic situation (the binary classification problem). We know there is still a lot of work to do and we will continue to work on other learning tasks.

Comment 3: The presentation of the theoretical results (Theorem 3.1 and Theorem 3.2) is poor and informal, and the proofs lack rigor. For example, the threshold 0.6 in Theorem 3.2 is mysterious and does not seem to stem from any grounded argument.
Response 3: Thanks for your comments. We will think about them carefully and reconsider the presentation. We also want to make clear about the threshold 0.6 in Theorem 3.2. In our paper, we try to provide a threshold to decide which one is better to use a quantum kernel method or a classical kernel method. This threshold δ_0 is a empirical quantity that be determined through several datasets (81 datasets in our experiment). The "true" value

of δ_0 is unknown, since it is a value determined by as many datasets as possible. We cannot try all the datasets to make sure of this. But, the existence of δ still make sense. At least, we know a phenomenon that the variable δ can has some influence to decide whether a quantum kernel method is better or not when compared with a classical kernel method. Based on our experience, the δ_0 will take 0.6. To illustrate how it works, we give an example here. Suppose we get the δ of a specific dataset D, if the δ for the D is larger than δ_0, we have enough reason to believe that we can use classical kernels to learn this dataset. On the other hand, suppose the δ for the D is less or equal to δ_0. According to theorem 3.2, even though we cannot directly say that the quantum kernels will be better, it at least provides us a choice to use quantum kernels. Whether the quantum kernels will be better depends on the data pattern. For example, as we mentioned in the paper, if we meet the Mersenne Twister random distribution, the quantum kernels will be superior.

Reviewer 3 (Reviewer cR4C)

Comment 1: The writing quality of the paper is not ideal. Sections 2 and 3 are long but not informative. The figures in experiments are hard to understand. The paper has a lot of grammar errors.
Response 1: Thanks for the time and efforts to review our work. We will reconsider the whole paper carefully and continue to improve our work. We use much space to introduce the related work in section 2 since it involves much knowledge. In section 3 we try to introduce our contributions in detail. Fig.4, Fig.5, Fig.6, and Fig.7 are mainly about the comparison of quantum kernel methods and classical kernel methods based on different datasets. The high-level variable is the dataset. We will continue to make our work more readable and correct the grammar errors.

Comment 2: The paper seems to focus on the classification problems. Quantum kernels can be applied to a much wider range of problems. The terminology "quantum kernel methods" in the title may need to be changed to quantum kernel based classification methods.
Response 2: Thanks for the helpful suggestion and we will modify the title. The title is going to be: "Where Can Quantum Kernel-Based Classification Methods Make A Big Difference?".

Comment 3: The proof of theorem 3.1 is just a summary of statements in

the literature. The proof of theorem 3.2 is just an application of CLT (and with some errors). For example, the first sentence in the proof of Theorem 3.2. "Suppose our measurement independent identical distribution ... where M is the random variable, R is the number of measurement shots." is not correct in both English and mathematics. Also, I am not sure if the proof really did the job to prove the statements in Theorem 3.2.

Response 3: Thanks for the advice. We will reconsider and continue to improve our theorems. Here we just want to make our theorems clearer. (1).Theorem 3.1 shows that quantum kernels are superior to classical kernels when meeting a random distribution based on Mersenne Twister Generator. To show this point, we designed an experiment, and the results are shown in Fig.4(B). The results show that the quantum kernels are almost always better than the classical kernels. As the number of data increases, the quantum kernels will maintain a stable advantage over classical kernels. We think it is an interesting phenomenon. Some researchers already proved that the quantum kernels can be superior to classical kernels when learning a DLP problem. Also, we know learning a Mersenne Twister distribution is a DLP problem. That's why we assume that the *Z-ZZ* feature map can effectively simulate the efficacy of the feature map proposed by [10] in the beginning. However, we cannot provide a rigorous mathematical proof at present. We will continue to focus and work on this in the future.

(2) In theorem 3.2, we try to provide a threshold to decide which one is better to use a quantum kernel method or a classical kernel method. This threshold δ_0 is a empirical quantity that be determined through several datasets (81 datasets in our experiment). The "true" value of δ_0 is unknown, since it is a value determined by as many datasets as possible. We cannot try all the datasets to make sure of this. But, the existence of δ still make sense. At least, we know a phenomenon that the variable δ can has some influence to decide whether a quantum kernel method is better or not when compared with a classical kernel method. Based on our experience, the δ_0 will take 0.6. To illustrate how it works, we give an example here. Suppose we get the δ of a specific dataset D, if the δ for the D is larger than δ_0, we have enough reason to believe that we can use classical kernels to learn this dataset. On the other hand, suppose the δ for the D is less or equal to δ_0. According to theorem 3.2, even though we cannot directly say that the quantum kernels will be better, it at least provides us a choice to use quantum kernels. Whether the quantum kernels will be better depends on the data pattern. For example, as we mentioned in the paper, if we meet the

Mersenne Twister random distribution, the quantum kernels will be superior.

Comment 4: The experiments are not convincing. Many implementation details are missing. The results, figures, and explanations are hard to understand. No replication codes are provided. My guess is the experiments are done by simulations run on the classical computer rather than real quantum computers?
Response 4: Thanks for the time and work to review our work. We will continue to improve our work. We can also apply the replication codes. The experiments are run on our local computer, but with the help of the IBM quantum platform where the bottom layer is a quantum computer.

Reviewer 4 (Reviewer 3XVL)

Comment 1: I have some difficulties with section 3: proof of theorem 3.1. The proof hinges on an assumption - "the Z-ZZ feature map can effectively simulate the efficacy of the feature map proposed by Liu et al. (2021)" - but I don't see any proof of this conjecture, at least in the mathematical sense. If this is axiomatic to the theorem then it should be presented as such in the theorem: otherwise the proof doesn't work afaict. Have I missed something here?
Response 1: We appreciate the reviewer for recognizing our work. Some researchers such as [10] already proved that the quantum kernels can be superior to classical kernels when learning a DLP problem. In our paper, we know learning a Mersenne Twister distribution is a DLP problem. That's why we assume that the *Z-ZZ* feature map can effectively simulate the efficacy of the feature map proposed by [10] in the beginning. Our experiments showed that the quantum kernel methods are superior to classical kernel methods. However, we cannot provide any proof of this conjecture at present. We will continue to focus and work on this in the future.

Comment 2: theorem 3.2: Why is δ_0 usually taken as 0.6? Is this based on the experimental results alone, or is there some intuition as to why this particular threshold is important?
Response 2: We appreciate for the time and efforts to review our work. We want to make clear about the threshold 0.6 in Theorem 3.2. In our paper, we try to provide a threshold to decide which one is better to use a quantum kernel method or a classical kernel method. This threshold δ_0 is a empirical

quantity that be determined through several datasets (81 datasets in our experiment). The "true" value of δ_0 is unknown, since it is a value determined by as many datasets as possible. We cannot try all the datasets to make sure of this. But, the existence of δ still make sense. At least, we know a phenomenon that the variable δ can has some influence to decide whether a quantum kernel method is better or not when compared with a classical kernel method. Based on our experience, the δ_0 will take 0.6. To illustrate how it works, we give an example here. Suppose we get the δ of a specific dataset D, if the δ for the D is larger than δ_0, we have enough reason to believe that we can use classical kernels to learn this dataset. On the other hand, suppose the δ for the D is less or equal to δ_0. According to theorem 3.2, even though we cannot directly say that the quantum kernels will be better, it at least provides us a choice to use quantum kernels. Whether the quantum kernels will be better depends on the data pattern. For example, as we mentioned in the paper, if we meet the Mersenne Twister random distribution, the quantum kernels will be superior.

Comment 3: However the experimental section is very thorough and quite convincing, so I am willing to overlook my misgivings regarding the theorems. Minor point: In the paragraph before equation (1), is the definition of the feature map $f_q : x_i \rightarrow \langle\phi(x_i)|\phi(x_j)\rangle$ correct? And if so, how does x_j fit here?

Response 3: Thanks for recognizing our work. We will continue to improve our work. There is a typo in the f_q. We modified it as follow: $f_q : x_i \rightarrow |\phi(x_i)\rangle$.

References

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

[2] Y. Quek, S. Fort, and H. K. Ng, "Adaptive quantum state tomography with neural networks," *npj Quantum Information*, vol. 7, no. 1, pp. 1–7, 2021.

[3] G. Verdon, T. McCourt, E. Luzhnica, V. Singh, S. Leichenauer, and J. Hidary, "Quantum graph neural networks," *arXiv preprint arXiv:1909.12264*, 2019.

[4] D. Garg, S. Ikbal, S. K. Srivastava, H. Vishwakarma, H. Karanam, and L. V. Subramaniam, "Quantum embedding of knowledge for reasoning," *Advances in Neural Information Processing Systems*, vol. 32, pp. 5594–5604, 2019.

[5] S. K. Srivastava, D. Khandelwal, D. Madan, D. Garg, H. Karanam, and L. V. Subramaniam, "Inductive quantum embedding," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[6] K. Meichanetzidis, S. Gogioso, G. De Felice, N. Chiappori, A. Toumi, and B. Coecke, "Quantum natural language processing on near-term quantum computers," *arXiv preprint arXiv:2005.04147*, 2020.

[7] C. Burges, "Data mining and knowledge discovery 2: 121," 1998.

[8] H.-T. Lin and C.-J. Lin, "A study on sigmoid kernels for svm and the training of non-psd kernels by smo-type methods," *submitted to Neural Computation*, vol. 3, no. 1-32, p. 16, 2003.

[9] M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 8, no. 1, pp. 3–30, 1998.

[10] Y. Liu, S. Arunachalam, and K. Temme, "A rigorous and robust quantum speed-up in supervised machine learning," *Nature Physics*, pp. 1–5, 2021.