

Article

Non-Parametric Semi-Supervised Learning in Many-Body Hilbert Space with Rescaled Logarithmic Fidelity

Wei-Ming Li and Shi-Ju Ran * 

Department of Physics, Capital Normal University, Beijing 100048, China; 2190602002@cnu.edu.cn

* Correspondence: sjran@cnu.edu.cn

Abstract: In quantum and quantum-inspired machine learning, a key step is to embed the data in the quantum space known as Hilbert space. Studying quantum kernel function, which defines the distances among the samples in the Hilbert space, belongs to the fundamental topics in this direction. In this work, we propose a tunable quantum-inspired kernel function (QIKF) named rescaled logarithmic fidelity (RLF) and a non-parametric algorithm for the semi-supervised learning in the quantum space. The rescaling takes advantage of the non-linearity of the kernel to tune the mutual distances of samples in the Hilbert space, and meanwhile avoids the exponentially-small fidelities between quantum many-qubit states. Being non-parametric excludes the possible effects from the variational parameters, and evidently demonstrates the properties of the kernel itself. Our results on the hand-written digits (MNIST dataset) and movie reviews (IMDb dataset) support the validity of our method, by comparing with the standard fidelity as the QIKF as well as several well-known non-parametric algorithms (naive Bayes classifiers, k -nearest neighbors, and spectral clustering). High accuracy is demonstrated, particularly for the unsupervised case with no labeled samples and the few-shot cases with small numbers of labeled samples. With the visualizations by t -stochastic neighbor embedding, our results imply that the machine learning in the Hilbert space complies with the principles of maximal coding rate reduction, where the low-dimensional data exhibit within-class compressibility, between-class discrimination, and overall diversity. The proposed QIKF and semi-supervised algorithm can be further combined with the parametric models such as tensor networks, quantum circuits, and quantum neural networks.

Keywords: quantum machine learning; semi-supervised learning; non-parametric learning; quantum kernel function

MSC: 81P16



Citation: Li, W.-M.; Ran, S.-J. Non-Parametric Semi-Supervised Learning in Many-Body Hilbert Space with Rescaled Logarithmic Fidelity. *Mathematics* **2022**, *10*, 940. <https://doi.org/10.3390/math10060940>

Academic Editor: Jan Śladowski

Received: 11 February 2022

Accepted: 11 March 2022

Published: 15 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In machine learning and other relevant fields, kernel [1,2] is defined as a special function to characterize the similarity (or distance) between any two samples after mapping them to a high-dimensional space. For the quantum and quantum-inspired machine learning (QML), an essential step to process classical data is to embed the data (e.g., images, texts, etc.) to the quantum space known as Hilbert space [3–7]. The quantum kernel function (QKF) that characterizes the distributions in the Hilbert space [3–18] is usually a critical factor for the performance of a QML scheme.

Among the existing mappings from data to the quantum state representations, a widely recognized example is known as the quantum feature map (see, e.g., [10,19–21]). It maps each feature to the state of one qubit, and each sample to an M -qubit product state (with M the number of features). Such a quantum feature map brings interpretability from the perspective of quantum probabilities, and have succeeded in the supervised [10,20,21] and unsupervised learning [19,22] algorithms as well as in the QML experiments [23]. It is unexplored to use such a quantum feature map for semi-supervised learning.

In quantum information and computation [24], fidelity serves as a fundamental quantity to characterize the similarity of two quantum states, and has been applied in tomography [25], verification [26], and detection of quantum phase transitions [27–32]. One drawback of fidelity for the many-qubit states is that it usually decreases exponentially with the number of qubits M , which is known as the “orthogonal catastrophe”. Instability or overflow of the precisions may occur for large M . One way to avoid the “orthogonal catastrophe” is to use the logarithmic fidelity as the QKF (for instance [27,33,34]). However, it is unclear for machine learning how the mutual distances of the samples or the data structure will be altered by taking logarithm on the fidelity.

Motivated by the growing needs on investigating QKFs for particularly machine learning, we here propose the rescaled logarithmic fidelity (RLF) as a tunable quantum-inspired kernel function (QIKF). To show its validity, we implement non-parametric semi-supervised learning in the Hilbert space based on RLF, which we name as RLF-NSSL. Being non-parametric, we can exclude the possible effects from the variational parameters and focus on the space and kernel. Note for the parametrized models, say neural networks, the performances are mainly determined by their architecture and parameter complexities, such as the arrangements of different types of layers and the numbers of variational parameters therein. In the RLF-NSSL, a given sample is classified by comparing the RLF’s between this sample and the clusters that are formed by labeled and pseudo-labeled samples. A strategy for pseudo-labeling is proposed. RLF-NSSL achieves better accuracy comparing with several established non-parametric methods such as naive Bayes classifiers, k -nearest neighbors, and spectral clustering. Particularly for the unsupervised or few-shot cases where the choice of kernel is crucial, the high performance of our method indicates the validity of RLF for QML.

The clusters formed by the samples with labels and pseudo-labels also define a mapping from the original space to a low-dimensional effective space. With the visualization by t-SNE [35], we show that the low-dimensional data exhibit within-class compressibility, between-class discrimination, and overall diversity. It implies that the machine learning in the Hilbert space also complies with the principles of maximal coding rate reduction (MCR²) [36,37]. We expect that these findings, including the RLF and the pseudo-labeling strategy, would generally benefit the QML using the parametric models, such as tensor networks [10,19–22,38–48], parametric quantum circuits [49–53,53–56], and quantum neural networks [57–63]. The connections to the maximal coding rate reduction shows the possibility of understanding and evaluating the QML methods from the perspective of coding rate. Our code can be publicly found on Github [64].

2. Hilbert Space and Rescaled Logarithmic Fidelity

Given a sample that we assume to be a M -component vector $\mathbf{x} = \{x_1, x_2, \dots, x_M\}$ with $0 \leq x_m \leq 1$, the feature map (see, e.g., [10,19–21]) to encode it to a M -qubit product states is written as

$$|\phi\rangle = \prod_{m=1}^M \left[\cos\left(\frac{x_m\pi}{2}\right)|0_m\rangle + \sin\left(\frac{x_m\pi}{2}\right)|1_m\rangle \right]. \quad (1)$$

Here, $|0_m\rangle$ and $|1_m\rangle$ form a set of orthonormal basis for the m -th qubit, which satisfy $\langle a_m | b_m \rangle = \delta_{ab}$.

In quantum information, the quantity to characterize the similarity between two states $|\phi^1\rangle$ and $|\phi^2\rangle$ is the fidelity f defined as the absolute value of their inner product $f(|\phi^1\rangle, |\phi^2\rangle) = |\langle \phi^1 | \phi^2 \rangle|$. As each state is normalized, we have $0 \leq f \leq 1$. With $f = 0$, the two states are orthogonal to each other and have the smallest similarity. With $f = 1$, the states satisfy $|\phi^1\rangle = e^{i\alpha}|\phi^2\rangle$ with α a universal phase factor. In this case, $|\phi^1\rangle$ and $|\phi^2\rangle$ can be deemed as a same state (meaning zero distance).

The fidelity with the feature map in Equation (1) results in a QKF to characterize the similarity between two samples \mathbf{x}^1 and \mathbf{x}^2 , which reads

$$f(\mathbf{x}^1, \mathbf{x}^2) = \prod_{m=1}^M \left| \cos \left[\frac{\pi}{2} (x_m^1 - x_m^2) \right] \right|. \quad (2)$$

In other words, the similarity between \mathbf{x}^1 and \mathbf{x}^2 is characterized by the fidelity f between the two product states obtained by implementing the feature map on these two samples.

Equation (2) shows that the fidelity is the product of M non-negative numbers $|\cos[\frac{\pi}{2}(x_m^1 - x_m^2)]| \leq 1$ (the equality holds when the corresponding feature takes the same value in the two samples, i.e., $x_m^1 = x_m^2$). Consequently, $f(\mathbf{x}^1, \mathbf{x}^2)$ decreases exponentially with the number of pixels that take different values in \mathbf{x}^1 and \mathbf{x}^2 . Taking MNIST dataset as an example, there are usually $O(10^2)$ such pixels. Then, f will be extremely small, meaning that the states from any two of the samples are almost orthogonal to each other. This is known as the “orthogonal catastrophe”, where instability or precision overflow may occur.

One way to resolve this problem is to use the logarithmic fidelity (for instance [27,33,34]).

$$\begin{aligned} F(\mathbf{x}^1, \mathbf{x}^2) &= \log_{10} f(\mathbf{x}^1, \mathbf{x}^2) \\ &= \sum_{m=1}^M \log \{ |\cos[\frac{\pi}{2}(x_m^1 - x_m^2)]| + \varepsilon \} \end{aligned} \quad (3)$$

with ε a small positive constant to avoid $\log 0$. F is a non-positive scalar that also characterizes the similarity between the given states. Though F changes monotonously with f , the mutual distances among the samples obtained by these two kernels are definitely different. For instance, we might have $\sum_n F(\mathbf{x}^1, \mathbf{x}^n) < \sum_n F(\mathbf{x}^2, \mathbf{x}^n)$ while $\sum_n f(\mathbf{x}^1, \mathbf{x}^n) > \sum_n f(\mathbf{x}^2, \mathbf{x}^n)$, due to the nonlinearity of the logarithmic function.

3. Rescaled Logarithmic Fidelity and Classification Scheme

In this work, we take advantage of the nonlinearity and define the rescaled logarithmic fidelity (RLF) as

$$\tilde{f}_\beta(\mathbf{x}^1, \mathbf{x}^2) = \beta^{F(\mathbf{x}^1, \mathbf{x}^2)} = \beta^{\log_{10} f(\mathbf{x}^1, \mathbf{x}^2)} \quad (4)$$

with β a tunable parameter that we dub as the rescaling factor. In particular, for $\beta = 10$, the RLF becomes the fidelity, i.e., $\tilde{f}_{10}(\mathbf{x}^1, \mathbf{x}^2) = f(\mathbf{x}^1, \mathbf{x}^2)$.

With certain labeled training samples $\{\mathbf{x}^l\}$ from P classes, an unlabeled sample \mathbf{y} can be classified in a supervised learning process. First, we transform \mathbf{y} to a P -dimensional effective vector $\tilde{\mathbf{y}}$, where its p -th element is the average RLF with the training samples from the p -th class

$$\tilde{y}_p = \frac{1}{N_p} \sum_{\mathbf{x}^l \in \text{class-}p} \tilde{f}_\beta(\mathbf{x}^l, \mathbf{y}), \quad (5)$$

with N_p the number of the labeled samples that belong to the p -th class. We call the labeled samples in a same class as a cluster. The clusters define a dimensionality reduction map given by Equation (5) from the original feature space to a P -dimensional space. In practice, we take $N_p = N$ as a same number for all p . The classification of \mathbf{y} is then indicated by the largest element of $\tilde{\mathbf{y}}$ as

$$c(\mathbf{y}) = \arg \max_p (\tilde{y}_p). \quad (6)$$

One can see that except certain hyper-parameters such as the rescaling factor β and the number of labeled samples, the above method contains no variational parameters, thus is dubbed as non-parametric supervised learning with RLF (RLF-NSL).

Classically, RLF can be easily calculated. Therefore, the classification algorithms based on RLF can be regarded as the quantum-inspired machine learning schemes running on classical computers. Considering to run such algorithms on the quantum platforms, the main challenge is the estimation of Equation (5) in order to obtain the similarity between a given sample and the clusters. It requires to estimate the rescaled logarithmic fidelity \tilde{f}_β and calculate the summation over the samples in the cluster. In our cases, estimating \tilde{f}_β is much easier than estimating the fidelity or implementing the full-state tomography for arbitrary states, since it is essentially the fidelity between two product states. Quantum acceleration over classical computation is unlikely in calculating such a fidelity, however, it is possible to gain quantum acceleration by parallelly computing the summations over the samples. This requires to design the corresponding quantum circuit regarding RLF, which is an open issue for the future investigations.

To demonstrate how β affects the classification accuracy, we choose the MNIST dataset [65] as an example, and randomly take $N = 10$ labeled samples from each class of the training set. The MNIST dataset contains the grey-scale images of hand-written digits, where the resolution of each image is 28×28 (meaning 784 features in each image). The images are divided into two sets with 60,000 images as the training samples and 10,000 as the testing samples. We obtain the effective vectors $\{\tilde{\mathbf{y}}\}$ of all testing samples using Equation (5), and calculate the classification using Equation (6). The testing accuracy γ is calculated as the number of the correctly classified testing samples divided by the total number. Figure 1 shows the γ when the number of the labeled samples in each class is small (few-shot learning with $N = 10$). We show the average of γ by implementing the simulations for 20 times, and the variances are illustrated by the shadowed areas. All the variances in this paper are obtained a similar way. One can see γ firstly rises and then drops by increasing β , and reaches the maximum around $1.2 < \beta < 2$. Note the β that gives the maximal γ slightly changes with different N .

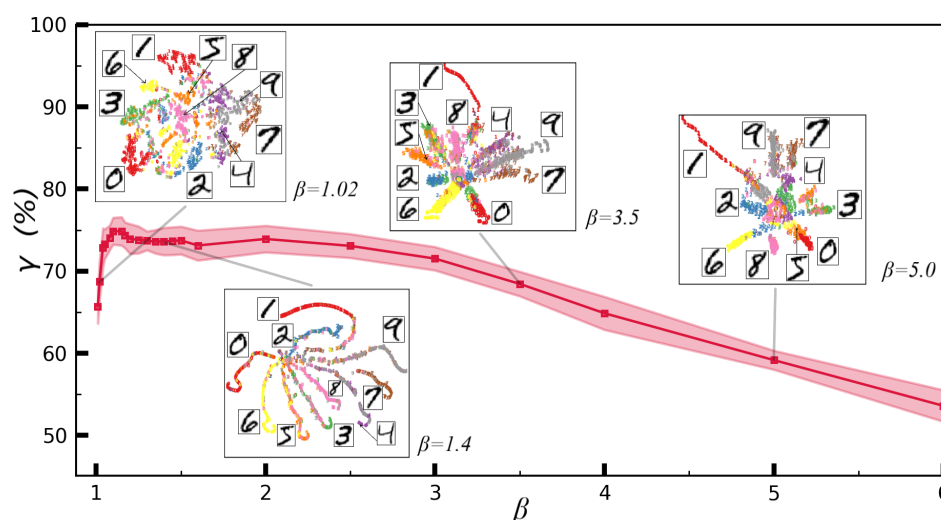


Figure 1. The β dependence of the testing accuracy γ of the testing samples for the ten classes in the MNIST dataset, where ε_t is evaluated by randomly taking $N=10$ training samples from each classes. Note the RLF becomes the standard fidelity for $\beta = 10$. We take the average of ε_t by implementing the simulations for 20 times, and the variances are indicated by the shadowed area. By t-SNE, the insets show the visualized distributions of 2000 effective vectors $\{\tilde{\mathbf{y}}\}$ (Equation (5)) that are randomly taken from the testing samples.

In the insets, we randomly take 200 testing samples from each class, and reduce the dimension of the effective vectors $\{\tilde{\mathbf{y}}\}$ from 10 to 2 by t-SNE [35], in order to visualize the distribution of the testing samples. The t-SNE is a non-linear dimensionality reduction method. It maps the given samples to a lower-dimensional space by reducing the number of features. The reduction is optimal in the sense that the mutual distances (or similarities)

among the samples in the lower-dimensional space should be close to those in the original space. By eyes, one can observe better separation for larger γ (e.g., $\beta = 1.4$) compared with those β 's giving smaller γ . More discussions are given below from the perspective of rate reduction [36,37]. We also confirm with more simulations that the fidelity (equivalently with $\beta = 10$ in the RLF) gives lower accuracy with $\gamma \simeq 50\%$. Note this accuracy is also not stable since the fidelity is exponentially small.

Figure 2 demonstrates how the testing accuracy of RLF-NSL is affected by β with different numbers of labeled samples N in each class. In all N s that vary from 6 to 240, γ firstly rapidly rises and the slowly decreases with β . Approximately for $\beta \simeq 1.3$, relatively high testing accuracy is obtained in all cases.

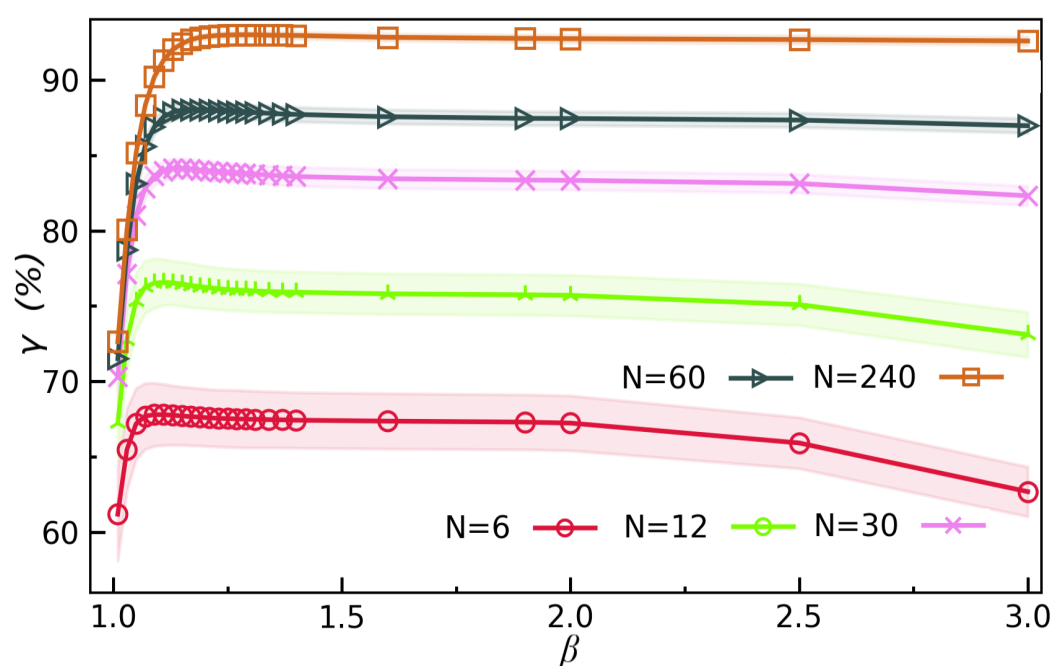


Figure 2. Testing accuracy γ of non-parametric supervised learning using rescaled logarithmic fidelity (RLF-NSL) on the MNIST dataset with different number of labeled samples N in each class.

4. Non-Parametric Semi-Supervised Learning with Pseudo-Labels

Based on RLF, we propose a non-parametric semi-supervised learning algorithm (RLF-NSSL). Different from supervised learning where sufficiently many labeled samples are required to implement the machine learning tasks, the key of semi-supervised learning is, in short, to utilize the samples whose labels are predicted by the algorithm itself. The generated labels are called pseudo-labels. The supervised learning can be considered as a special case of semi-supervised learning with zero pseudo-labels. For the unsupervised kernel-based classifications where there is no labeled samples, pseudo-labels can be useful to implement the classification tasks in a way similar to the supervised cases. The strategy of tagging the pseudo-labels is key to the prediction accuracy. Therefore, for the unsupervised (and also the few-shot cases with a small number of labeled samples), the performance should strongly rely on the choice of kernel. Here, we define P clusters, of which each contains two parts: all the N labeled training samples in this class and \tilde{N} unlabeled samples that are classified to this class. The rescaling factor is taken as the optimal β with $(N + \tilde{N})$ labeled samples in RLF-NSL. The key is how to choose the \tilde{N} samples with pseudo-labels to expand the clusters.

Our strategy is to divide the unlabeled training samples into batches for classification and pseudo-labeling. The clusters are initialized by the labeled samples. Then, for each batch of the unlabeled samples, we classify them by calculating the effective vectors given by Equation (5), where the summation is over all the samples with labels and pseudo-labels (if any). Then, we add these samples to the corresponding clusters according to their

classifications. The cluster is used to classify the testing set after all unlabeled training samples are classified.

Inevitably, the incorrect pseudo-labels would be introduced into the clusters, which may harm the classification accuracy. Therefore, we propose to update the samples in the clusters. To this aim, we define the confidence. For a sample \mathbf{y} in the p -th cluster, it is defined as

$$\eta = \frac{\tilde{y}_p}{\sum_{p'=1}^P \tilde{y}_{p'}}, \quad (7)$$

with \tilde{y}_p obtained by Equation (5). Then, in each cluster, we keep N_Δ pseudo-labels with the highest confidence. The rest pseudo-labels are removed, and the corresponding samples are thrown to the pool of the unlabeled samples, which are to be classified in the future iterations.

Figure 3 shows the testing accuracy γ of the MNIST dataset with different numbers of the labeled samples N . The accuracy of RLF-NSL (green line) already surpasses the recognized non-parametric methods including k -nearest neighbors (KNN) [66] with $k = 1$ and 10, and the naive Bayesian classifier [67]. KNN is also a kernel-based classification method. One first calculates the distances (or similarities) between the target sample and all labeled samples, and then find the k labeled samples with the smallest distances. The classification of the target sample is given by finding the class that has the largest number in these k samples. The performance of RLF-NSL significantly surpasses a baseline model, where we simply replace the RLF f_β in Equation (5) by the Euclidean kernel

$$f_E = \|\mathbf{x}^l - \mathbf{y}\|. \quad (8)$$

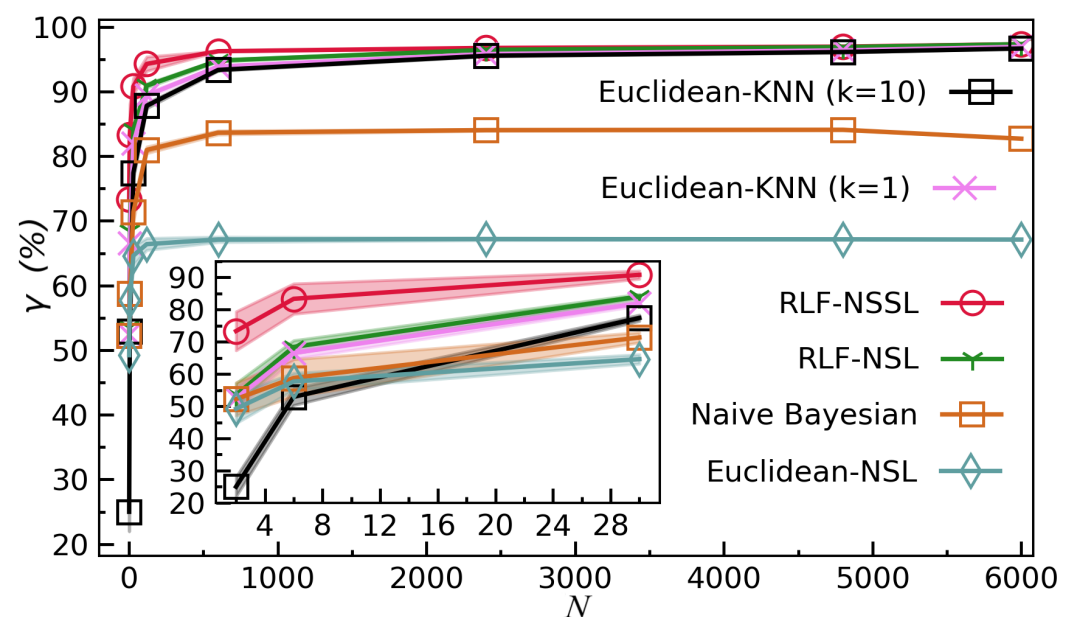


Figure 3. Testing accuracy γ of non-parametric semi-supervised and supervised learning using rescaled logarithmic fidelity (RLF-NSSL and RLF-NSL, respectively) on the MNIST dataset. Our results are compared with k -nearest neighbors with $k = 1$ and 10, naive Bayesian classifiers, and a baseline model by simply replacing RLF by the Euclidean distance. For more results of the KNN with different values of k and those of the p -norm distance with different values of p , please refer to the Appendix A.

The RLF-NSSL achieves the highest accuracy among all the presented methods. Significant improvement is observed particularly for the few-shot learning with small N , as

shown in the inset. Note for different N , we optimize β in RLF-NSL and we fix $\beta = 1.3$ in RLF-NSSL.

For the unsupervised learning, we assume there are no labeled samples from the very beginning. All samples in the clusters will be those with pseudo-labels. To start with, we randomly chose one sample to form a cluster. From all the unlabeled samples, every time we select a new sample (denoted as \tilde{x}) that satisfies two conditions:

$$\tilde{x} = \operatorname{argmin}_{\mathbf{x}} \sum_{\mathbf{x}_i \in \text{clusters}} F(\mathbf{x}, \mathbf{x}_i); \quad (9)$$

$$F(\tilde{x}, \mathbf{x}_i) < \mu \text{ for } \forall \mathbf{x}_i \in \text{clusters}. \quad (10)$$

with μ a preset small constant. Repeating the above procedure for $(P - 1)$ times, we have P clusters, of which each contains one sample. These samples have relatively small mutual similarities, thus are reasonable choices to initialize the clusters. We classify all the samples out of the clusters using the method explained in Section 3. Then, all samples will be added to the corresponding clusters according to the classifications.

The next step is to use the semi-supervised learning method introduced in Section 4 to update the samples in the clusters. Specifically, we remove the pseudo-labels for part of the samples in each cluster with the lowest confidence, and throw them to the pool of the unlabeled samples. We subsequently classify all the unlabeled samples and add them to the clusters correspondingly. Repeat the processes above until the clusters converge.

Table 1 compares the testing accuracy γ of our RLF-NSSL with other two unsupervised methods k -means [68,69] and spectral clustering [70–72]. We use the way proposed in [73] to determine the labels of the clusters. For each iteration in the RLF-NSSL to update the clusters, we remove the pseudo-labels of 35% of the samples with the lowest confidence in each cluster, which are to be re-classified in the next iteration. Our RLF-NSSL achieves the highest accuracy among these three methods. RLF-NSSL exhibits relatively high standard deviation, possibly due to the large fluctuation induced by the (nearly) random initialization of the clusters. Such fluctuation can be suppressed by incorporating a proper initialization strategy.

Table 1. The testing accuracy γ and the standard deviation (std.) on the MNIST dataset using k -means, spectral clustering, and RLF-NSSL ($N = 0$, i.e., no labeled samples). We use the way proposed in [73] to determine the labels of the clusters in the case of unsupervised learning. For the k -means, we use the randomly initialized clustering center and take 270 iteration steps. The similarity is characterized by Euclidean distance. For spectral clustering, we use the SpectralClustering function from the “sklearn” package in Python.

	<i>k</i> -Means	Spectral Clustering	RLF-NSSL ($N = 0$)
γ (%)	56.21	65.46	72.64
Std.	1.83	0.1	5.43

We compare the testing accuracy by using different kernels and classification strategies, as shown in Figure 4. We choose IMDB [74], a recognized dataset in the field of natural language processing. Each sample is a comment on a movie, and the task is to predict whether it is positive or negative. The dataset contains 50,000 samples, in which half for training and half for testing. For convenience, we limit the maximal number of the features in a sample (i.e., the maximal number of words in a comment) to $M_{\max} = 100$, and finally use 2773 training samples and 2963 testing samples. The labeled samples are randomly selected from the training samples, and the testing accuracy is evaluated by the testing samples. We test two classification strategies, which are KNN and NSL. We also compare different kernels. The Euclidean distance f_E is given by Equation (8). For the Gaussian kernel, the distance is defined by a Gaussian distribution, which satisfies

$$f_G(\mathbf{x}^1, \mathbf{x}^2) = e^{-\frac{f_E^2(\mathbf{x}^1, \mathbf{x}^2)}{2\sigma^2}}. \quad (11)$$

where σ controls the variance. For the Euclidean-NSL algorithm, we use f_E in Equation (5) to obtain the classifications. The rest parts are the same as RLF-NSL. For the Euclidean-KNN algorithm, we use f_E to obtain the k labeled samples with the smallest distances. In RLF-NSL, we flexibly adjust the rescaling factor β as the number of labeled samples varies. The RLF-NSL achieves the highest testing accuracy among these algorithms.

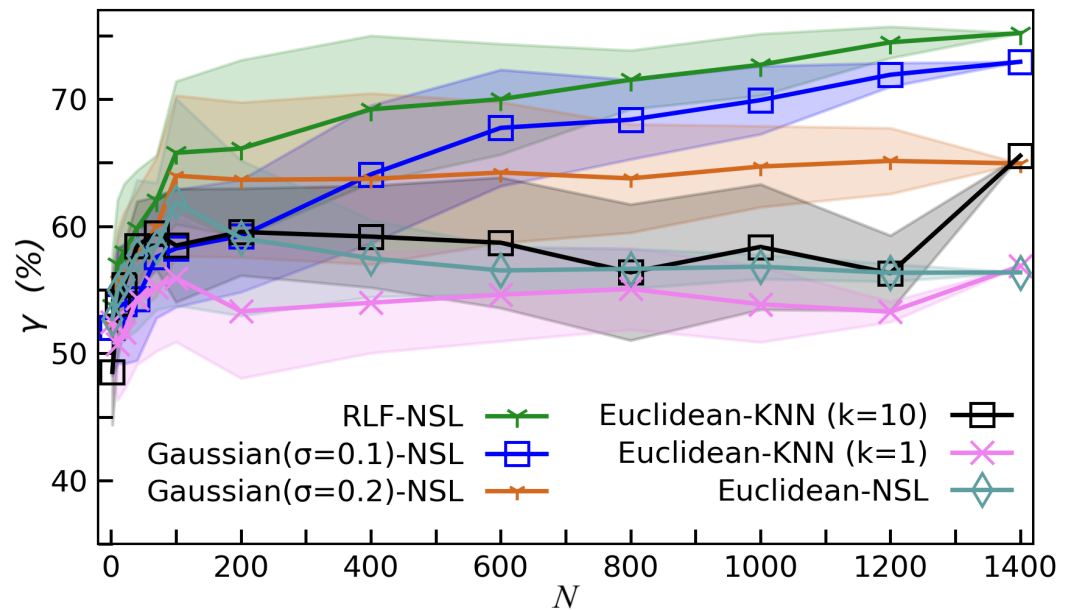


Figure 4. The testing accuracy of the RLF-NSL on the IMDB dataset comparing different kernels (Euclidean, Gaussian, and RLF) and classification strategies (KNN and NSL). The x-axis shows the number of labeled samples in each class. For more results of the KNN with different values of k and those of the Gaussian kernel with different values of σ , please refer to the Appendix A.

To demonstrate how the classification precision is improved by updating the pseudo-labeled samples in the clusters, we take the few-shot learning by RLF-NSSL as an example. At the beginning, there are $N = 6$ labeled samples in each class to define the cluster. For the zeroth epoch, the low-dimensional data and classifications are obtained by these labeled samples. In the update process, each epoch contains three sub-steps. In the first sub-step, we classify 500 samples with the highest RLF samples and add them to the corresponding clusters according to their classifications. In the second and third sub-steps, we update the clusters by replacing part of the pseudo-labeled samples. Specifically, we move 500 samples that have the lowest confidence η from each cluster to the pool of the unclassified samples. Then we calculate the classifications of all samples in the pool, and add the 500 samples with the highest RLF to the corresponding clusters.

In Figure 5a,b, we show the confidence η and classification accuracy γ_c for the samples inside the clusters. Each time when we add new samples to the clusters in the first sub-step of each epoch (see red markers), both η and γ_c decrease. By updating the clusters, we observe obvious improvement of η by replacing the less confident samples in the clusters. Slight improvement of γ_c is observed in general after the second sub-step. Even though the pseudo-labels of the samples in the clusters become less accurate as the clusters contain more and more pseudo samples, we observe monotonous increase (with insignificant fluctuations) of the testing accuracy γ as shown in Figure 5c. This is a convincing evidence on the validity of our RLF-NSSL and the pseudo-labeling strategy.

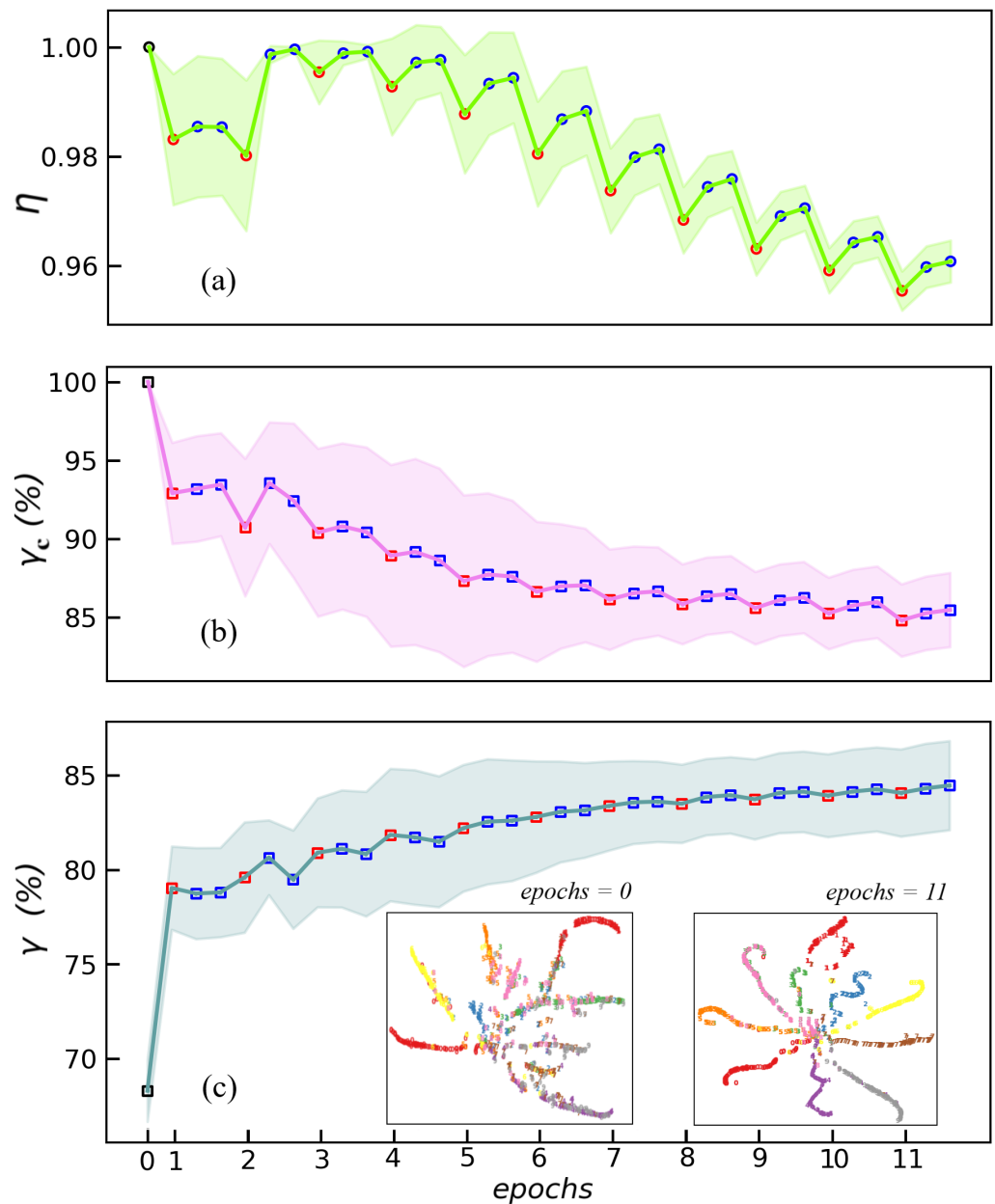


Figure 5. For the RLF-NSSL on the MNIST dataset with $N = 6$ labeled samples in each class (few-shot case), (a,b) show the confidence η and classification accuracy γ_c of the samples in the clusters, respectively, in different epochs. (c) shows the classification accuracy γ for the testing set. The insets of (c) illustrate the visualizations by applying t-SNE to the testing samples in the low-dimensional space (Equation (5)). See the details in the main text.

5. Discussion from the Perspective of Rate Reduction

In Refs. [36,37], several general principles were proposed on the continuous mapping from the original feature space to a low-dimensional space for the purposes of classification or clustering, known as the principles of maximal coding rate reduction (MCR²). Considering the classification problems, the representations should satisfy the following properties: (a) samples from the same class should belong to low-dimensional linear subspaces; (b) samples from different classes should belong to different low-dimensional subspaces and be uncorrelated; (c) the variance of features for the samples in the same class should be as large as possible as long as (b) is satisfied. These three principles are known as within-class compressibility, between-class discrimination, and overall diversity, respectively.

Our results imply that MCR^2 should also apply to the machine learning in the Hilbert space of many qubits. In our scheme, the clusters map each sample (the product state obtained by the feature map given by Equation (4)) to a P -dimensional vector. The distribution of these vectors is defined by their mutual distances based on the RLF.

The insets of Figure 5c show the visualizations of the P -dimensional vectors from the testing set at the 0th and 11th epochs. At the 0th epoch, we simply use the labeled samples to define the clusters. The testing accuracy is less than 70%. At the 11th epoch, the clusters consist of the labeled and pseudo-labeled samples. The pseudo-labeled samples are updated using the RLF-NSSL algorithm. The testing accuracy is round 85%. Comparing the two distributions in the two-dimensional space, it is obvious that at the 11th epoch, the samples in the same class tend to form a one-dimensional stream in the visualization, indicating the within-class compressibility. The samples are distributed as “radiations” from the middle toward the edges, indicating the overall diversity. Each two neighboring radial lines give a similar angle, indicating the between-class discrimination. Inspecting from these three aspects, one can see that the samples at the 11th epoch better satisfy the MCR^2 than those at the 0th epoch. Similar phenomena can be observed in the insets of Figure 1. The β giving higher testing accuracy would better obey the MCR^2 , and vice versa. These results provide preliminary evidence for the validity of MCR^2 for the machine learning in the Hilbert space.

6. Summary

With the fast development of quantum and quantum-inspired machine learning, there are the strong needs of understanding and exploring the quantum kernel functions. In this work, we propose the rescaled logarithmic fidelity (RLF) to define a tunable quantum-inspired kernel function for characterizing the similarity between many-body states. The advantages of RLF are revealed by applying it for non-parametric semi-supervised learning. Higher accuracy is achieved particularly for the unsupervised and few-shot cases, compared with the standard fidelity and several well-established non-parametric methods. From the visualization of the data in the low-dimensional effective space defined by the RLF, our results support the applicability of MCR^2 theory. Surely, more tests on different datasets should be done to verify the advantages of the RLF over the classical kernel functions. The proposed RLF and the semi-supervised learning strategies contribute to understand and improve the QML methods, including those with parametric models (such as parametrized quantum circuits [49–53,53–56] and tensor networks [10,19–22,38–48]).

Author Contributions: Conceptualization, S.-J.R.; methodology, S.-J.R.; programming, W.-M.L.; validation, W.-M.L. and S.-J.R.; formal analysis, S.-J.R.; writing—original draft preparation, W.-M.L.; writing—review and editing, W.-M.L. and S.-J.R.; visualization, W.-M.L.; supervision, S.-J.R.; project administration, S.-J.R.; funding acquisition, S.-J.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by NSFC (Grant No. 12004266 and No. 11834014), Beijing Natural Science Foundation (No. 1192005 and No. Z180013), Foundation of Beijing Education Committees (No. KM202010028013), and the key research project of Academy for Multidisciplinary Studies, Capital Normal University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors are thankful to Ding Liu and Zhanghao Zhouyin for stimulating discussions.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Supplemental Results on the MNIST and IMDB Datasets

In the appendix, we compare the accuracies by tuning the hyperparameters including k in the KNN, σ in the Gaussian distribution, and p of the p-norm. Our results show that the RLF-NSL achieves in general the highest accuracy by taking the optimal rescaling factor β .

Appendix A.1. MNIST Dataset

Tables A1 and A2 give the classification accuracies γ and the standard deviation of KNN for different values of k with the numbers of labeled samples $N = 6$ (the few-shot case) and $N = 600$. The accuracies of the RLF-NSL are also given for comparison, where we take the rescaling factor $\beta = 1.08$ and 1.35 for $N = 6$ and 600 , respectively.

Table A1. The classification accuracy γ and the standard deviation (std.) on the MNIST dataset for $N = 6$ by KNN with different k and by the RLF-NSL with $\beta = 1.08$.

k	1	2	3	4	6	8	10	12	RLF-NSL
γ (%)	66.53	60.52	60.15	60.52	56.07	54.84	52.90	50.77	68.40
std.	1.86	1.95	2.32	1.64	2.54	2.22	2.59	3.06	2.03

Table A2. The classification accuracy γ and the standard deviation (std.) on the MNIST dataset for $N = 600$ by KNN with different k and by the RLF-NSL with $\beta = 1.35$.

k	1	2	3	4	6	8	10	12	RLF-NSL
γ (%)	93.86	93.67	93.90	93.67	93.73	93.60	93.34	93.23	94.78
std.	0.14	0.11	0.05	0.11	0.17	0.15	0.16	0.12	0.16

In Figure A1, we show the classification accuracy γ by replacing the RLF kernel in the RLF-NSL algorithm with the p-norm kernel, where p varies from 0 to 20. Note the p-norm gives the sparsity for $p = 0$, and gives the maximum of the absolute values of the elements for $p \rightarrow \infty$. The horizontal dash line shows the accuracy of the RLF-NSL with $\beta = 1.08$. The shadows illustrate the standard deviation. Our results suggest that γ increases with p , but soon converges for about $p > 10$. The converged accuracy is lower than that of the RLF-NSL.

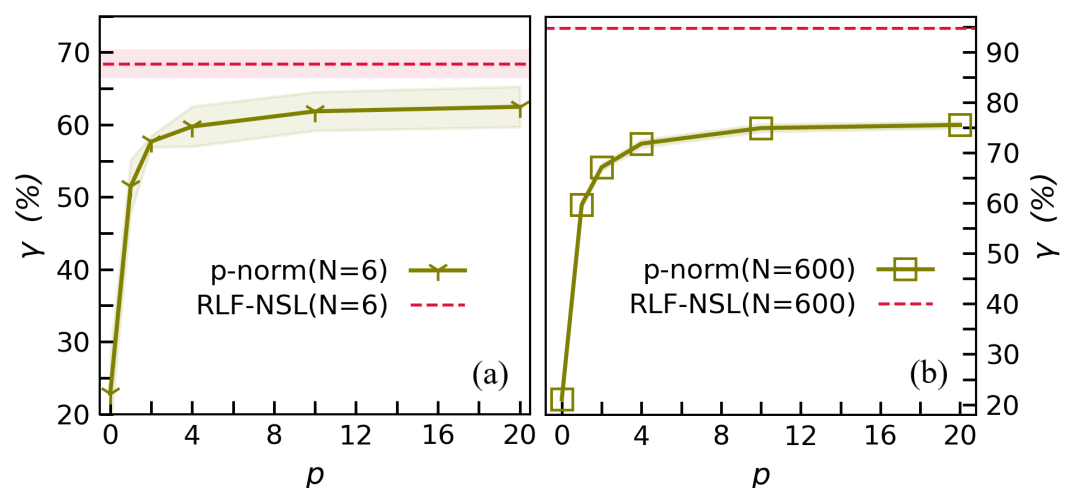


Figure A1. The classification accuracy γ on the MNIST dataset for (a) $N = 6$ and (b) $N = 600$. The solid line with symbols show the results by using p-norm as the kernel in the NSL algorithm. The horizontal dash lines show the accuracies of the RLF-NSL with $\beta = 1.08$ and 1.35 , respectively. The shadows demonstrate the standard deviation.

Appendix A.2. IMDb Dataset

Tables A3 and A4 give the classification accuracies γ and the standard deviation of KNN for different values of k with the numbers of labeled samples $N = 100$ and $N = 1200$. The accuracies of the RLF-NSL are also given for comparison, where we take the rescaling factor $\beta = 1.06$ in both tables.

Table A3. The classification accuracy γ and the standard deviation (std.) on the IMDb dataset for $N = 100$ by KNN with different k and by the RLF-NSL with $\beta = 1.06$.

k	1	2	4	6	8	10	12	14	16	RLF-NSL
γ (%)	55.92	50.47	54.69	55.60	57.09	58.72	57.93	59.67	56.69	65.78
std.	5.02	5.21	5.13	4.81	5.16	4.16	3.92	4.17	4.45	5.62

Table A4. The classification accuracy γ and the standard deviation (std.) on the IMDb dataset for $N = 1200$ by KNN with different k and by the RLF-NSL with $\beta = 1.06$.

k	1	2	4	6	8	10	12	14	16	RLF-NSL
γ (%)	50.82	48.89	55.54	56.63	57.19	56.68	58.04	56.51	51.14	74.46
std.	4.58	0.86	1.70	2.86	2.21	3.63	2.18	3.58	2.21	1.24

Tables A5 and A6 give the classification accuracies γ and the standard deviation of NSL algorithm by using the Gaussian kernel functions with σ varying from 0.05 to 0.5. The highest accuracy appears near $\sigma \simeq 0.2$ for $N = 100$, and near $\sigma \simeq 0.1$ for $N = 1200$. In both tables, we take $\beta = 1.06$ in the RLF-NSL algorithm, which obtains the highest accuracies comparing with those from the Gaussian kernel function.

Table A5. The classification accuracy γ and the standard deviation(std.) on the IMDb for $N = 100$ by the NSL with the Gaussian and RLF kernels. For the Gaussian kernel, we vary σ from 0.05 to 0.5. For the RLF-NSL, we take $\beta = 1.06$.

σ	0.05	0.1	0.15	0.2	0.25	0.5	RLF-NSL
γ (%)	53.35	58.25	60.84	63.97	61.66	61.48	65.78
std.	3.62	4.623	4.99	5.80	5.97	7.50	5.62

Table A6. The classification accuracy γ and the standard deviation(std.) on the IMDb for $N = 1200$ by the NSL with the Gaussian and RLF kernels. For the Gaussian kernel, we vary σ from 0.05 to 0.5. For the RLF-NSL, we take $\beta = 1.06$.

σ	0.05	0.1	0.15	0.2	0.25	0.5	RLF-NSL
γ (%)	61.78	71.91	70.03	65.14	62.53	58.76	74.46
std.	1.18	0.93	2.12	2.59	1.91	1.57	1.24

Figure A2 demonstrates the classification accuracies γ on the IMDb dataset for (a) $N = 100$ and (b) $N = 1200$. The solid lines with symbols show the γ obtained by the NSL algorithm using the p -norm as the kernel, where p varies from 0 to 10. Our data suggest that the value of p does not cause much difference on the accuracy. The results by the RLF-NSL algorithm with $\beta = 1.06$ are given by the horizontal dash lines. The shadows illustrate the standard deviation. In general, the RLF-NSL achieves higher accuracy than those using the p -norm kernel.

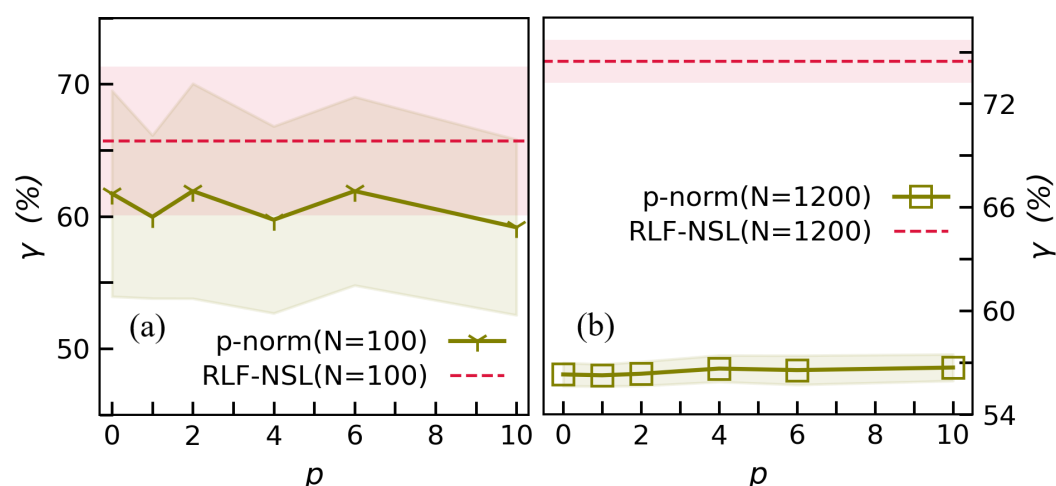


Figure A2. The classification accuracies γ on the IMDb dataset for (a) $N = 100$ and (b) $N = 1200$, obtained by the NSL with the p-norm for different values of p (solid lines with symbols) and with the RLF as the kernel (horizontal dash lines). The shadows show the standard deviation.

References

- Shawe-Taylor, J.; Cristianini, N. *Kernel Methods for Pattern Analysis*; Cambridge University Press: Cambridge, UK, 2004.
- Hofmann, T.; Schölkopf, B.; Smola, A.J. Kernel methods in machine learning. *Ann. Stat.* **2008**, *36*, 1171–1220. [\[CrossRef\]](#)
- Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum machine learning. *Nature* **2017**, *549*, 195–202. [\[CrossRef\]](#) [\[PubMed\]](#)
- Schuld, M.; Killoran, N. Quantum Machine Learning in Feature Hilbert Spaces. *Phys. Rev. Lett.* **2019**, *122*, 040504. [\[CrossRef\]](#) [\[PubMed\]](#)
- Havlíček, V.; Córcoles, A.D.; Temme, K.; Harrow, A.W.; Kandala, A.; Chow, J.M.; Gambetta, J.M. Supervised learning with quantum-enhanced feature spaces. *Nature* **2019**, *567*, 209–212. [\[CrossRef\]](#)
- Lloyd, S.; Schuld, M.; Ijaz, A.; Izaac, J.; Killoran, N. Quantum embeddings for machine learning. *arXiv* **2020**, arXiv:quant-ph/2001.03622.
- Schuld, M. Supervised quantum machine learning models are kernel methods. *arXiv* **2021**, arXiv:quant-ph/2101.11020.
- Wiebe, N.; Braun, D.; Lloyd, S. Quantum Algorithm for Data Fitting. *Phys. Rev. Lett.* **2012**, *109*, 050505. [\[CrossRef\]](#)
- Lloyd, S.; Mohseni, M.; Rebentrost, P. Quantum principal component analysis. *Nat. Phys.* **2014**, *10*, 631–633. [\[CrossRef\]](#)
- Stoudenmire, E.; Schwab, D. Supervised learning with tensor networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 4806–4814.
- Schuld, M.; Sinayskiy, I.; Petruccione, F. Prediction by linear regression on a quantum computer. *Phys. Rev. A* **2016**, *94*, 022342. [\[CrossRef\]](#)
- Benedetti, M.; Realpe-Gómez, J.; Biswas, R.; Perdomo-Ortiz, A. Quantum-Assisted Learning of Hardware-Embedded Probabilistic Graphical Models. *Phys. Rev. X* **2017**, *7*, 041052. [\[CrossRef\]](#)
- Schuld, M.; Fingerhuth, M.; Petruccione, F. Implementing a distance-based classifier with a quantum interference circuit. *EPL (Europhys. Lett.)* **2017**, *119*, 60002. [\[CrossRef\]](#)
- Kerenidis, I.; Landman, J.; Luongo, A.; Prakash, A. q-means: A quantum algorithm for unsupervised machine learning. In Proceedings of the NeurIPS 2019, Vancouver, BC, Canada, 8–14 December 2019.
- Zhao, Z.; Fitzsimons, J.K.; Fitzsimons, J.F. Quantum-assisted Gaussian process regression. *Phys. Rev. A* **2019**, *99*, 052331. [\[CrossRef\]](#)
- LaRose, R.; Coyle, B. Robust data encodings for quantum classifiers. *Phys. Rev. A* **2020**, *102*, 032420. [\[CrossRef\]](#)
- Huang, H.Y.; Broughton, M.; Mohseni, M.; Babbush, R.; Boixo, S.; Neven, H.; McClean, J.R. Power of data in quantum machine learning. *Nat. Commun.* **2021**, *12*, 2631. [\[CrossRef\]](#) [\[PubMed\]](#)
- Park, D.K.; Blank, C.; Petruccione, F. The theory of the quantum kernel-based binary classifier. *Phys. Lett. A* **2020**, *384*, 126422. [\[CrossRef\]](#)
- Han, Z.Y.; Wang, J.; Fan, H.; Wang, L.; Zhang, P. Unsupervised Generative Modeling Using Matrix Product States. *Phys. Rev. X* **2018**, *8*, 031012. [\[CrossRef\]](#)
- Liu, D.; Ran, S.J.; Wittek, P.; Peng, C.; García, R.B.; Su, G.; Lewenstein, M. Machine learning by unitary tensor network of hierarchical tree structure. *New J. Phys.* **2019**, *21*, 073059. [\[CrossRef\]](#)
- Sun, Z.Z.; Peng, C.; Liu, D.; Ran, S.J.; Su, G. Generative tensor network classification model for supervised machine learning. *Phys. Rev. B* **2020**, *101*, 075135. [\[CrossRef\]](#)
- Ran, S.J.; Sun, Z.Z.; Fei, S.M.; Su, G.; Lewenstein, M. Tensor network compressed sensing with unsupervised machine learning. *Phys. Rev. Res.* **2020**, *2*, 033293. [\[CrossRef\]](#)
- Wang, K.; Xiao, L.; Yi, W.; Ran, S.J.; Xue, P. Quantum image classifier with single photons. *arXiv* **2020**, arXiv:quant-ph/2003.08551.

24. Nielsen, M.A.; Chuang, I. Quantum computation and quantum information. *Am. J. Phys.* **2002**, *70*, 558. [[CrossRef](#)]
25. D'Ariano, G.M.; Lo Presti, P. Quantum Tomography for Measuring Experimentally the Matrix Elements of an Arbitrary Quantum Operation. *Phys. Rev. Lett.* **2001**, *86*, 4195–4198. [[CrossRef](#)] [[PubMed](#)]
26. Buhrman, H.; Špalek, R. Quantum Verification of Matrix Products. In Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm, Miami, FL, USA, 22–24 January 2006; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2006; SODA '06, pp. 880–889.
27. Zhou, H.Q.; Orús, R.; Vidal, G. Ground State Fidelity from Tensor Network Representations. *Phys. Rev. Lett.* **2008**, *100*, 080601. [[CrossRef](#)]
28. Abasto, D.F.; Hamma, A.; Zanardi, P. Fidelity analysis of topological quantum phase transitions. *Phys. Rev. A* **2008**, *78*, 010301. [[CrossRef](#)]
29. Schwandt, D.; Alet, F.; Capponi, S. Quantum Monte Carlo Simulations of Fidelity at Magnetic Quantum Phase Transitions. *Phys. Rev. Lett.* **2009**, *103*, 170501. [[CrossRef](#)]
30. Quan, H.T.; Cucchietti, F.M. Quantum fidelity and thermal phase transitions. *Phys. Rev. E* **2009**, *79*, 031101. [[CrossRef](#)]
31. Zhao, J.H.; Zhou, H.Q. Singularities in ground-state fidelity and quantum phase transitions for the Kitaev model. *Phys. Rev. B* **2009**, *80*, 014403. [[CrossRef](#)]
32. Xiong, H.N.; Ma, J.; Sun, Z.; Wang, X. Reduced-fidelity approach for quantum phase transitions in spin- $\frac{1}{2}$ dimerized Heisenberg chains. *Phys. Rev. B* **2009**, *79*, 174425. [[CrossRef](#)]
33. Ran, S.J. Encoding of matrix product states into quantum circuits of one- and two-qubit gates. *Phys. Rev. A* **2020**, *101*, 032310. [[CrossRef](#)]
34. Yang, Y.; Sun, Z.Z.; Ran, S.J.; Su, G. Visualizing quantum phases and identifying quantum phase transitions by nonlinear dimensional reduction. *Phys. Rev. B* **2021**, *103*, 075106. [[CrossRef](#)]
35. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
36. Ma, Y.; Derksen, H.; Hong, W.; Wright, J. Segmentation of Multivariate Mixed Data via Lossy Data Coding and Compression. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 1546–1562. [[CrossRef](#)] [[PubMed](#)]
37. Yu, Y.; Chan, K.H.; You, C.; Song, C.; Ma, Y. Learning Diverse and Discriminative Representations via the Principle of Maximal Coding Rate Reduction. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 9422–9434.
38. Socher, R.; Chen, D.; Manning, C.D.; Ng, A. Reasoning with Neural Tensor Networks for Knowledge Base Completion. In *Advances in Neural Information Processing Systems*; Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2013; Volume 26.
39. Cheng, S.; Chen, J.; Wang, L. Information Perspective to Probabilistic Modeling: Boltzmann Machines versus Born Machines. *Entropy* **2018**, *20*, 583. [[CrossRef](#)]
40. Chen, J.; Cheng, S.; Xie, H.; Wang, L.; Xiang, T. Equivalence of restricted Boltzmann machines and tensor network states. *Phys. Rev. B* **2018**, *97*, 085104. [[CrossRef](#)]
41. Cheng, S.; Wang, L.; Xiang, T.; Zhang, P. Tree tensor networks for generative modeling. *Phys. Rev. B* **2019**, *99*, 155131. [[CrossRef](#)]
42. Huggins, W.; Patil, P.; Mitchell, B.; Whaley, K.B.; Stoudenmire, E.M. Towards quantum machine learning with tensor networks. *Quantum Sci. Technol.* **2019**, *4*, 024001. [[CrossRef](#)]
43. Schröder, F.A.Y.N.; Turban, D.H.P.; Musser, A.J.; Hine, N.D.M.; Chin, A.W. Tensor network simulation of multi-environmental open quantum dynamics via machine learning and entanglement renormalisation. *Nat. Commun.* **2019**, *10*, 1062. [[CrossRef](#)]
44. Efthymiou, S.; Hidayi, J.; Leichenauer, S. TensorNetwork for Machine Learning. *arXiv* **2019**, arXiv:cs.LG/1906.06329.
45. Sun, Z.Z.; Ran, S.J.; Su, G. Tangent-space gradient optimization of tensor network for machine learning. *Phys. Rev. E* **2020**, *102*, 012152. [[CrossRef](#)] [[PubMed](#)]
46. Guo, C.; Modi, K.; Poletti, D. Tensor-network-based machine learning of non-Markovian quantum processes. *Phys. Rev. A* **2020**, *102*, 062414. [[CrossRef](#)]
47. Cheng, S.; Wang, L.; Zhang, P. Supervised learning with projected entangled pair states. *Phys. Rev. B* **2021**, *103*, 125117. [[CrossRef](#)]
48. Reyes, J.; Stoudenmire, E.M. A multi-scale tensor network architecture for machine learning. *Mach. Learn. Sci. Technol.* **2021**, *2*, 035036. [[CrossRef](#)]
49. Zhu, D.; Linke, N.M.; Benedetti, M.; Landsman, K.A.; Nguyen, N.H.; Alderete, C.H.; Perdomo-Ortiz, A.; Korda, N.; Garfoot, A.; Brecque, C.; et al. Training of quantum circuits on a hybrid quantum computer. *Sci. Adv.* **2019**, *5*, eaaw9918. [[CrossRef](#)] [[PubMed](#)]
50. Benedetti, M.; Garcia-Pintos, D.; Perdomo, O.; Leyton-Ortega, V.; Nam, Y.; Perdomo-Ortiz, A. A generative modeling approach for benchmarking and training shallow quantum circuits. *NPJ Quantum Inf.* **2019**, *5*, 45. [[CrossRef](#)]
51. Benedetti, M.; Lloyd, E.; Sack, S.; Fiorentini, M. Parameterized quantum circuits as machine learning models. *Quantum Sci. Technol.* **2019**, *4*, 043001. [[CrossRef](#)]
52. Chen, S.Y.C.; Yang, C.H.H.; Qi, J.; Chen, P.Y.; Ma, X.; Goan, H.S. Variational Quantum Circuits for Deep Reinforcement Learning. *IEEE Access* **2020**, *8*, 141007–141024. [[CrossRef](#)]
53. Du, Y.; Hsieh, M.H.; Liu, T.; Tao, D. Expressive power of parametrized quantum circuits. *Phys. Rev. Res.* **2020**, *2*, 033125. [[CrossRef](#)]
54. Cao, S.; Wossnig, L.; Vlastakis, B.; Leek, P.; Grant, E. Cost-function embedding and dataset encoding for machine learning with parametrized quantum circuits. *Phys. Rev. A* **2020**, *101*, 052309. [[CrossRef](#)]

55. Xin, T.; Che, L.; Xi, C.; Singh, A.; Nie, X.; Li, J.; Dong, Y.; Lu, D. Experimental Quantum Principal Component Analysis via Parametrized Quantum Circuits. *Phys. Rev. Lett.* **2021**, *126*, 110502. [[CrossRef](#)] [[PubMed](#)]
56. Cincio, L.; Rudinger, K.; Sarovar, M.; Coles, P.J. Machine Learning of Noise-Resilient Quantum Circuits. *PRX Quantum* **2021**, *2*, 010324. [[CrossRef](#)]
57. Farhi, E.; Neven, H. Classification with Quantum Neural Networks on Near Term Processors. *arXiv* **2018**, arXiv:quant-ph/1802.06002.
58. McClean, J.R.; Boixo, S.; Smelyanskiy, V.N.; Babbush, R.; Neven, H. Barren plateaus in quantum neural network training landscapes. *Nat. Commun.* **2018**, *9*, 4812. [[CrossRef](#)]
59. Cong, I.; Choi, S.; Lukin, M.D. Quantum convolutional neural networks. *Nat. Phys.* **2019**, *15*, 1273–1278. [[CrossRef](#)]
60. Killoran, N.; Bromley, T.R.; Arrazola, J.M.; Schuld, M.; Quesada, N.; Lloyd, S. Continuous-variable quantum neural networks. *Phys. Rev. Res.* **2019**, *1*, 033063. [[CrossRef](#)]
61. Mari, A.; Bromley, T.R.; Izaac, J.; Schuld, M.; Killoran, N. Transfer learning in hybrid classical-quantum neural networks. *Quantum* **2020**, *4*, 340. [[CrossRef](#)]
62. Beer, K.; Bondarenko, D.; Farrelly, T.; Osborne, T.J.; Salzmann, R.; Scheiermann, D.; Wolf, R. Training deep quantum neural networks. *Nat. Commun.* **2020**, *11*, 808. [[CrossRef](#)]
63. Shen, H.; Zhang, P.; You, Y.Z.; Zhai, H. Information Scrambling in Quantum Neural Networks. *Phys. Rev. Lett.* **2020**, *124*, 200504. [[CrossRef](#)]
64. For Those Who Are Interested in Reproducing Our Results, We Have Made Our Codes Publicly. Available online: <https://github.com/Li-Wei-Ming/rlf.git> (accessed on 16 September 2021).
65. LeCun, Y.; Cortes, C.; Burges, C.J. The MNIST Database of Handwritten Digits. Available online: <http://yann.lecun.com/exdb/mnist/> (accessed on 17 April 2019).
66. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [[CrossRef](#)]
67. Langley, P.; Iba, W.; Thompson, K. An analysis of Bayesian classifiers. *Aai* **1992**, *90*, 223–228.
68. Sarle, W.S. Algorithms for clustering data. *Technometrics* **1990**, *32*, 227–229. [[CrossRef](#)]
69. Kaufman, L.; Rousseeuw, P.J. *Finding Groups in Data: An Introduction to Cluster Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2009; Volume 344.
70. Mehrotra, K.; Mohan, C.K.; Ranka, S. *Elements of Artificial Neural Networks*; MIT Press: Cambridge, MA, USA, 1997.
71. Ng, A.Y.; Jordan, M.I.; Weiss, Y. *On Spectral Clustering: Analysis and an Algorithm*; MIT Press: Cambridge, MA, USA, 2001; pp. 849–856, NIPS’01.
72. Kamvar, K.; Sepandar, S.; Klein, K.; Dan, D.; Manning, M.; Christopher, C. *Spectral Learning*; Technical Report 2003-25; Stanford InfoLab: Stanford, CA, USA, 2003.
73. Munkres, J. Algorithms for the Assignment and Transportation Problems. *J. Soc. Ind. Appl. Math.* **1957**, *5*, 32–38. [[CrossRef](#)]
74. Maas, A.L.; Daly, R.E.; Pham, P.T.; Huang, D.; Ng, A.Y.; Potts, C. Learning Word Vectors for Sentiment Analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; Association for Computational Linguistics: Portland, OR, USA, 2011; pp. 142–150.