Quantum Machine Learning: theory, applications and implementations on the Pasqal quantum processors

Slimane Thabet
(Dated: September 20, 2021)

I. INTRODUCTION

Pasqal est une spin-off de l'Institut d'Optique Graduate School (IOGS) situé à Palaiseau. Après plus de 10 ans de recherche universitaire à l'IOGS, l'équipe de Pasqal construit la prochaine génération de processeurs quantiques pour l'informatique et la simulation quantiques en contrôlant des dizaines d'atomes de rubidium individuels avec des lasers [1, 2].

L'apprentissage automatique, ou machine learning, est vu comme une application prometteuse des ordinateurs quantiques. En effet, les qubits évoluent dans un espace de Hilbert de taille exponentielle, ce qui permettrait d'implémenter des modèles probabilistes impossibles à évaluer sur des ordinateurs classiques [3]. Trouver un avantage quantique en machine learning nécessite de faire face à plusieurs défis parmi lesquels: encoder les données dans un circuit quantique, concevoir une procédure d'optimisation adaptée au calcul quantique, assurer la robustesse au bruit des ordinateurs quantiques actuels, prouver la capacité de généralisation des modèles quantiques ainsi créés, faire en sorte que les résultats obtenus soient meilleurs que pour les modèles classiques en termes de performance, de coût, ou d'explicabilité. Tous ces sous-domaines ont connu des progrès énormes ces cinq dernières années.

Le doctorant s'efforcera de développer la théorie et les procédures expérimentales liées au quantum machine learning avec à l'esprit l'implémentation sur les processeurs quantiques de Pasqal à base d'atomes neutres. Il s'appuiera sur les récents travaux sur la compariason des modèles probabilistes quantiques et classiques dans l'analyse des données financières [4] du professeur Elham Kashefi au LIP6, le laboratoire d'informatique de la faculté des sciences et de l'ingiénierie de l'université de la Sorbonne. L'expertise initiale des équipes du LIP6 en cryptographie quantique pourrait également ouvrir la voie à l'exploration de procédures d'apprentissage quantique sécurisées.

Les tâches principales de la thèse comprennent (mais ne sont pas limitées à):

- (T1) Développer la théorie de l'apprentissage via des modèles quantiques, notamment les capacités de généralisation et l'efficacité d'entraînement.
- (T2) Mettre en place des protocoles théoriques et expérimentaux pour implémenter des algorithmes de machine learning quantique, en particulier les procédures d'entrainement.
- (T3) Appliquer les techniques développées dans les tâches précédentes à des types de données particuliers, comme les graphes, les images, ou le texte.

Une estimation du calendrier global de la thèse est disponible figure 1

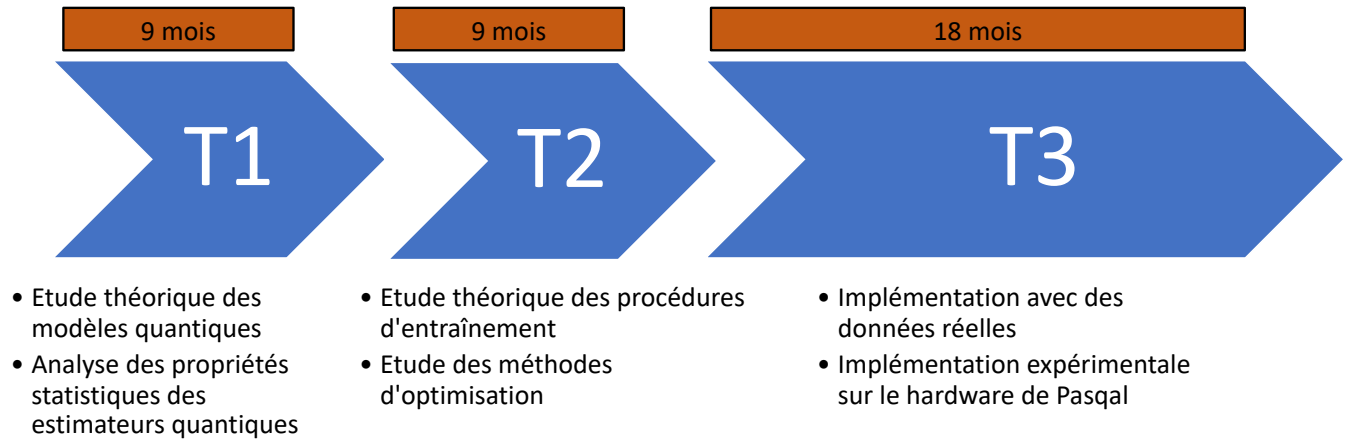

FIG. 1. Prévision de calendrier pour le déroulement de la thèse.

Dans ce qui suit, nous développons quelques points techniques liés aux tâches (T1), (T2), (T3) avec pour chaque partie une présentation des questions associées.

II. PASQAL'S HARDWARE

Pasqal's hardware is composed of Rubidium atoms trapped in optical tweezers on which lasers are applied [2]. N atoms located on $(r_1, \dots r_N)$ behave on the hamiltonian

$$H(t) = \sum_{i=1}^{N} \Omega_i(t) X_i + \sum_{i=1}^{N} \delta_i(t) Z_i + \frac{1}{2} \sum_{i<j} \frac{C_6}{|r_i - r_j|^6} n_i n_j \tag{1}$$

where Ω_i, δ_i are respectively the rabi frequency and the detuning of the driving laser fields pointing on each atoms, $n_i = (Id + Z_i)/2$.

Ω_is, δ_is and r_is are tunable parameters by the experimenter. The objective of this project can be summed up by determining how to tune those parameters in order to encode data and create parameterized quantum machine learning model.

III. (T1) THEORY

A. General ML problem

We consider two variables x and y such that there exist a latent unknown function f such that $y = f(x)$.

The problem of supervised learning is to provide an estimator $\hat{f}$ of f from a dataset of observations $(x_1, x_2, \dots, x_N)$ with targets $(y_1, y_2, \dots, y_N)$. The goal is to make a prediction $\hat{y} = \hat{f}(x)$ from a new observed data point x. If the targets are discrete, it is called a classification problem, and if the targets are continuous, it is called a regression problem.

Many ways exist to provide such an estimator, and we will develop the case of parameterized estimators. More specifically, we are interested by finding a function $\hat{f}$ parameterized by a vector of parameters $\boldsymbol{\theta} = (\theta_1, \dots, \theta_p) \in \Theta \subset \mathbb{R}^p$. We note the output of such a model $\hat{f}(x; \boldsymbol{\theta})$. Parameterized models encompass many machine learning algorithms such as linear regression, support vector machines, random forest, and neural networks.

Learning consists of finding $\tilde{\boldsymbol{\theta}}$ such that

$$\tilde{\boldsymbol{\theta}} = \arg\min_{\theta \in \Theta} \mathcal{L}(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta} \in \Theta} \frac{1}{N} \sum_{i=1}^{N} l(\hat{f}(x_i; \boldsymbol{\theta}) - y_i) \tag{2}$$

where l is a loss function.

B. Quantum models

A quantum model, or quantum neural network, is a function of x parameterized by θ that is evaluated by measurement of an observable on a quantum circuit. It can be formulated as

$$\hat{f}(x; \theta) = \langle 0|^{\otimes n} U(x; \boldsymbol{\theta})^\dagger M U(x; \boldsymbol{\theta}) |0\rangle^{\otimes n} \tag{3}$$

where U is a unitary evolution dependent both on x and θ and M is an observable, and n is the size of the register.

A specific class of quantum models is the alternate evolution models. In this case, we specify the form of U such that

$$U(x; \boldsymbol{\theta}) = \prod_{i=1}^{p} U_i(\theta_i) V(x) \tag{4}$$

where U_is and V are unitary evolutions.

Questions arise:

1. Which functions can be learned ?
2. How efficient are quantum models compared to classical ones ?

C. Elements of learning theory with quantum models

Determining how effective quantum models are is a very active area of research. In this section, we will detail some answers that have been brought recently.

1. *Fourier series*

Schuld et al. [5] provide some answers in the case of one dimensional variable. They assume a form of encoding unitary $V(x) = \exp(-ixH)$ where H is a fixed hamiltonian represented by an hermitian matrix.

f and $\hat{f}$ can be rewritten as Fourier series

$$f(x) = \sum_{\omega \in \Omega} c_\omega e^{i\omega x} \quad \text{and} \quad \hat{f}(x) = \sum_{\omega \in \hat{\Omega}} \hat{c}_\omega(\theta) e^{i\omega x} \tag{5}$$

$\hat{f}$ would then coincide with f iif their spectrum and coefficients coincide. The authors show that the spectrum of $\hat{f}$ is only determined by the eigenvalues of the encoding hamiltonian H whereas the coefficients are determined by the rest of the circuit. They conclude that quantum models are universal approximators if one is authorized to have as many parameters and repetitions of the encoding layer as necessary.

2. *Effective dimension and Fischer information*

In [6], the authors propose to use a metric called the effective dimension to evaluate the performance of statistical models. The effective dimension is linked to the Fischer information. They compare the effective dimension of quantum models to classical feed-forward neural networks.

A *statistical model* $\mathcal{M}_\Theta$ is a set of joint probability distributions between a data pair (x, y) as $p(x,\ y;\ \theta) = p(y|x;\theta)p(x)$ parameterized by a vector $\theta \in \Theta \subset [-1, 1]^d$. $\mathcal{M}_\Theta = \{p(.,.;\theta),\ \theta \in \Theta\}$.

The Fischer information matrix evaluated in θ is the matrix such that

$$F(\theta) = \mathbb{E}_{(x,y)\sim p}\big[\frac{\partial}{\partial\theta} p(x,\ y;\ \theta) \frac{\partial}{\partial\theta} p(x,\ y;\ \theta)^T\big] \in \mathbb{R}^{d\times d} \tag{6}$$

For n data samples, it can be approximated by the empirical Fischer information matrix

$$\hat{F}(\theta) = \sum_{i=1}^{n} \big[\frac{\partial}{\partial\theta} p(x_i,\ y_i;\ \theta) \frac{\partial}{\partial\theta} p(x_i,\ y_i;\ \theta)^T\big] \in \mathbb{R}^{d\times d} \tag{7}$$

The effective dimension of a statistical model $\mathcal{M}_\Theta = \{p(.,.;\theta),\ \theta \in \Theta\}$ for $\gamma \in (0, 1]$ and n data samples is defined as

$$d_{\gamma,n}(\mathcal{M}_\Theta) = 2\frac{\log(\frac{1}{V_\Theta}\int_\Theta \sqrt{\det(I_d + \frac{\gamma n}{2\pi \log n}\hat{F}(\theta)}\mathrm{d}\theta)}{\log(\frac{\gamma n}{2\pi \log n})} \tag{8}$$

where $V_\Theta := \int_\Theta d\theta$ is the volume of the parameter space and $\hat{F}(\theta) := d\frac{V_\Theta}{\int_\Theta \mathrm{tr}(F(\theta))\mathrm{d}\theta}$ is the normalized Fischer information matrix.

The authors show empirically that the spectrum of the Fischer matrix is concentrated around a few high eigenvalues for classical neural networks whereas it is more uniform for quantum neural networks. Furthermore, they show that quantum neural networks have a higher effective dimension than classical neural networks.

D. Perspectives

- What is the expressivity of quantum models compared to classical ones ?
- Are there convergence guarantees of quantum models ?
- Which inductive bias are related to quantum models ?

IV. (T2) IMPLEMENTATION

A. Theory

Modern machine learning algorithm are trained by gradient descent. At each step of the gradient descent, the new parameters $\boldsymbol{\theta}^{t+1}$ are updated by the rule $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \alpha\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}^t)$. In order to succesfully train a ML model, one should be able to efficiently compute the gradient. Deep learning practitioners have come up with the backpropagation rule that enables to compute the gradients of billions of parameters without too much time or memory consumption.

We will now explore strategies that have been found to compute the gradient of quantum circuits. [7–9]

1. *Parameter shift rule*

Let us assume a simple case, with a one dimensional parameter θ and a one dimensional variable x. We define $U(x;\theta) = \exp(-i\frac{\theta}{2}\sigma)V(x)$ where σ is a Pauli matrix. $\hat{f}$ is defined as in equation 3. Therefore the gradient is given by

$$\partial_\theta \hat{f}(x;\theta) = \frac{1}{2}\left(\hat{f}(x;\theta+\frac{\pi}{2}) - \hat{f}(x;\theta-\frac{\pi}{2})\right) \tag{9}$$

This formula is called the *parameter shift rule*. The gradient of a quantum function as defined in equation 3 can then be evaluated by two quantum circuit. This formula is exact, and it only suffers from the sampling error.

In the more general case of a multi-parameter layered evolution as in equation 4, the gradient of individual θ_is can be evaluated with the same formula by

$$\partial_{\theta_i} \hat{f}(x;\boldsymbol{\theta}) = \frac{1}{2}\left(\hat{f}(x;\boldsymbol{\theta}^{i+}) - \hat{f}(x;\boldsymbol{\theta}^{i-})\right) \tag{10}$$

where $\boldsymbol{\theta}^{i\pm}$ is equal to $\boldsymbol{\theta}$ where θ_i has been shifted by $\pm\frac{\pi}{2}$

2. *Stochastic parameter shift rule*

Sometimes the unitary evolutions are more complicated than the case described therefore. Thus, we have to use other ways to compute the gradients. Let us assume that $U(\theta) = \exp(-i(\theta G + H))$ where G and H are two non-commuting hamiltonians with G being a tensor product of Pauli matrices. We drop the dependence in x for simplification. Let us introduce

$$f_\pm(\boldsymbol{\theta}, s) = \langle\psi|U(\boldsymbol{\theta}, s)^\dagger_\pm M U(\boldsymbol{\theta}, s)_\pm|\psi\rangle \tag{11}$$

$$U_\pm(\boldsymbol{\theta}, s) = e^{is(\boldsymbol{\theta}G+H)}e^{\pm i\pi/4G}e^{i(1-s)(\boldsymbol{\theta}G+H)} \tag{12}$$

We have the following identity:

$$\partial_{\boldsymbol{\theta}} f = \int_0^1 [f_+(\boldsymbol{\theta}, s) - f_-(\boldsymbol{\theta}, s)]ds \tag{13}$$

The integral can then be estimated by Monte Carlo methods. There is still the constraint to be able to apply the unitary evolution $e^{\pm i\pi/4G}$. That is not possible on some devices like the one of Pasqal, the only gates one could apply are on the form $e^{it(bG+H)}$ where t and b are real. One can then approximate $e^{\pm i\pi/4G}$ by $e^{i\epsilon(\pi/(4\epsilon)G+H)}$ where $\epsilon \ll 1$.

B. Implementation on the Pasqal processor

Let us suppose now that our evolution is driven by the Pasqal hamiltonian defined in equation 1. We assume that it doesn't depend on time and we have $U = \exp(-iHt)$. The goal of this project would be to design a protocol inpired by the parameter-shift rules and their extensions to compute the derivatives $\partial\hat{f}/\partial\Omega_i, \partial\hat{f}/\partial\delta_i, \partial\hat{f}/\partial r_i$.

1. One local rabi pulse

As a first example, we will consider the hamiltonian defined in 1 where a single atom is excited. The driving hamiltonian then becomes

$$H = \Omega X_0 + \frac{1}{2}\sum_{i<j}\frac{C_6}{|r_i - r_j|^6}n_i n_j = \Omega X_0 + H_I \tag{14}$$

We define $U = \exp(-iH), f(\Omega) = \langle\psi|U^\dagger M U|\psi\rangle$. Computing the derivative $\partial f/\partial\Omega$ can be done with the stochastic parameter shift rule described in IV A 1 with $G = X_0$ and $H = H_I$. Since H_I can never be turned off, one have to use the approximation trick.

2. Global pulse

We now apply a global pulse to the system. That is currently the regime where the hardware works better. he driving hamiltonian is then

$$H = \Omega\sum_{i=1}^{N} X_i + \frac{1}{2}\sum_{i<j}\frac{C_6}{|r_i - r_j|^6}n_i n_j = \Omega G + H_I \tag{15}$$

The stochastic parameter shift rule can no longer be applied directly because G is not a product of Pauli matrices anymore. Instead, one need to decompose the contributions of Ω by the chain rule.

Let us define

$$H(\Omega_1, \ldots, \Omega_N) = \sum_{i=1}^{N}\Omega_i X_i + H_I \tag{16}$$

$$U = e^{-iH} \tag{17}$$

$$g(\Omega_1, \ldots, \Omega_N) = \langle\psi|U^\dagger M U|\psi\rangle \tag{18}$$

We then have $f(\Omega) = g(\Omega, \ldots \Omega)$ and

$$\frac{\partial f}{\partial\Omega} = \sum_{i=1}^{N}\frac{\partial g}{\partial\Omega_i}(\Omega, \ldots, \Omega) \tag{19}$$

Each term in the sum can be estimated by the parameter shift rule.

C. Perspectives

- Can we find parameter-shift rules for 'non-trivial' hamiltonian evolutions such as Ising or XY ?
- Can we differentiate w.r.t the initial position of the atoms in the register ?
- Are parameter shift rules efficient in practice to train quantum models ?

V. (T3) THE CASE OF GRAPH STRUCTURED DATA

In many fields such as chemistry, bioinformatics, network analysis, computer vision, natural language processing, data can naturally be modelled by graphs. However standard machine learning procedures are designed to take vectors as input. Therefore, specific algorithms for graph structured data had to be developed in order to tackle those tasks, such as graph neural networks and graph kernels [10–12]. The goal of this project would be to design a quantum version of these algorithms, to implement them on the hardware of Pasqal, and to apply them in a real case scenario.

A. Quantum graph neural networks

Quantum graph neural networks are a family of quantum neural networks introduced by [13]. Given a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, one can assign a hilbert space $\mathcal{H}_v$ for a node v and design a parameterized hamiltonian

$$\hat{\mathcal{H}}_{\mathcal{G}} = \sum_{(u,v)\in\mathcal{E}} W_{uv}\hat{O}_u^{(uv)} \otimes \hat{O}_v^{(uv)} + \sum_{v\in\mathcal{V}} V_v \hat{P}_v \tag{20}$$

where $\hat{O}_u^{(uv)}, \hat{O}_v^{(uv)}, \hat{P}_v$ are operators acting respectively on the Hilbert spaces of the nodes v, u, v. W_{uv} and V_v are trainable weights. $\hat{\mathcal{H}}_{\mathcal{G}}$ encodes the graph in its topology by design and thus contains potentially useful features about $\mathcal{G}$. The quantum state $|\psi_f\rangle = \prod_{p=1}^{P} \exp(-i\eta_p \hat{\mathcal{H}}_M)\exp(-i\theta_p\hat{\mathcal{H}}_{\mathcal{G}})\,|\psi_0\rangle$ given by P alternating evolutions of the hamiltonian $\hat{\mathcal{H}}_{\mathcal{G}}$ and a mixing hamiltonian $\hat{\mathcal{H}}_M$ such that $[\hat{\mathcal{H}}_{\mathcal{G}}, \hat{\mathcal{H}}_M] \neq 0$ parameterized by η_ps and θ_ps is a suitable ansatz for a machine learning problem on graphs. The hardware of Pasqal naturally enables the encoding of graph data. One can indeed dispose atoms in the register such that the Ising interaction between them corresponds to a desired graph. This procedure is nonetheless limited to local graphs.

B. Quantum graph kernels

A kernel is an inner product in a latent embedding space. More specifically, a graph kernel is a symmetric, positive semidefinite function defined on the space of graphs $\mathbb{G}$. Given a kernel K, there exists a map $\phi : \mathbb{G} \to \mathcal{H}$ into a Hilbert space $\mathcal{H}$ such that $K(\mathcal{G}_1,\mathcal{G}_2) = \langle\phi(\mathcal{G}_1)|\phi(\mathcal{G}_2)\rangle$ for all $\mathcal{G}_1,\mathcal{G}_2 \in \mathbb{G}$ [14]. Once the kernel is computed, it can be used in classical machine learning algorithm such as Support Vector Machine and Kernel Ridge Regression.

A quantum graph kernel is a protocol to compute a graph kernel using a quantum computer. Once the kernel is computed, it is used as an input for a kernel-based ML algorithm. A previous work has been done to design a quantum graph kernel called the Quantum Evolution kernel (QE) using a neutral atom platform. [15] The core idea of the QEK is to encode the graph data into the topology of a hamiltonian. In the case of a neutral atom platform, it would be the Ising part of the evolution. The procedure is summarized in figure 2

Given two graphs $\mathcal{G},\mathcal{G}'$, the computation of QE is performed in two steps: respectively associate probability distributions $\mathcal{P}$, $\mathcal{P}'$, and compute a distance d between those probability distributions. The final kernel is given by $k(\mathcal{G},\mathcal{G}') = \exp(-d(\mathcal{P},\mathcal{P}'))$.

We consider a system whose time-evolution is governed by a hamiltonian with the graph topology, noted $\hat{\mathcal{H}}_{\mathcal{G}}$ and we introduced another hamiltonian $\hat{\mathcal{H}}_\theta$ parameterized by θ and non-commuting with $\hat{\mathcal{H}}_{\mathcal{G}}$. The system starts in a predefined state $|\psi_0$. It then evolves according to an aternating scheme of one layer from $\hat{\mathcal{H}}_\theta$ and one layer from $\hat{\mathcal{H}}_{\mathcal{G}}$. The final state is then

$$|\psi_f\rangle = \prod_{i=1}^{p} \left(\mathrm{e}^{-\mathrm{i}\hat{\mathcal{H}}_{\theta_i}}\mathrm{e}^{-\mathrm{i}\hat{\mathcal{H}}_{\mathcal{G}}t_i}\right)\mathrm{e}^{-\mathrm{i}\hat{\mathcal{H}}_{\theta_0}}\,|\psi_0\rangle \tag{21}$$

The state is finally measured according to a diagonal observable $\hat{\mathcal{O}}$. The histogram of the different values of $\hat{\mathcal{O}}$ constitutes the probability distribution. The kernel is computed by measuring a distance between those probability distributions. The parameters θ_is and t_is are then optimized on a training set.

We performed numerical experiments on several datasets, and we compared our evolution kernel to classical graph kernels [14], the Random Walk (RW) kernel and the Graphlet Subsampling (GS) kernel.

C. Perspectives

- Can the potential advantages of quantum computing for machine learning be also found in graph based data ?
- Do graph based data have properties that make them suitable or not suitable for a potential quantum speed-up ?

Dataset	QE_1 (150)	QE_4 (2000)	QE_8 (6000)	GS	RW
IMDB-MULTI	46.8 ± 4.4	$\mathbf{48.1 \pm 4.4}$	47.7 ± 4.4	40.9 ± 3.5	45.2 ± 3.4
IMDB-BIN	69.0 ± 6.1	71.6 ± 5.7	$\mathbf{71.8 \pm 5.4}$	66.5 ± 5.9	67.8 ± 6.5
PTC_FM	62.5 ± 7.9	65.8 ± 7.9	$\mathbf{66.0 \pm 7.6}$	61.5 ± 8.9	59.4 ± 7.8
PROTEINS	73.3 ± 1.2	74.5 ± 2.6	$\mathbf{76.0 \pm 5.3}$	73.3 ± 1.2	73.3 ± 1.2
NCI1	78.1 ± 0.8	78.6 ± 3.2	$\mathbf{80.1 \pm 3.5}$	78.1 ± 0.8	78.1 ± 0.8
Fingerprint	58.6 ± 2.0	$\mathbf{60.2 \pm 3.2}$	60.1 ± 3.3	57.9 ± 3.3	59.9 ± 2.2

TABLE I. Comparison of the accuracy of the graph kernel on various data sets. The average over all cross-validation splits and the standard deviation associated is displayed. These values were obtained measuring the Ising energy after p layers with an Ising graph Hamiltonian (QE_p). Next to each model is indicated a rough approximation of the number of evaluations. The quantum kernel in the Ising setting outperforms the other kernels considered. The resources needed to evaluate the score on bigger data sets (sl e.g.IMDB-MULTI and Fingerprint) allow for less evaluations. Models with more layers could be tested in the case of Ising, which is faster to simulate than XY.

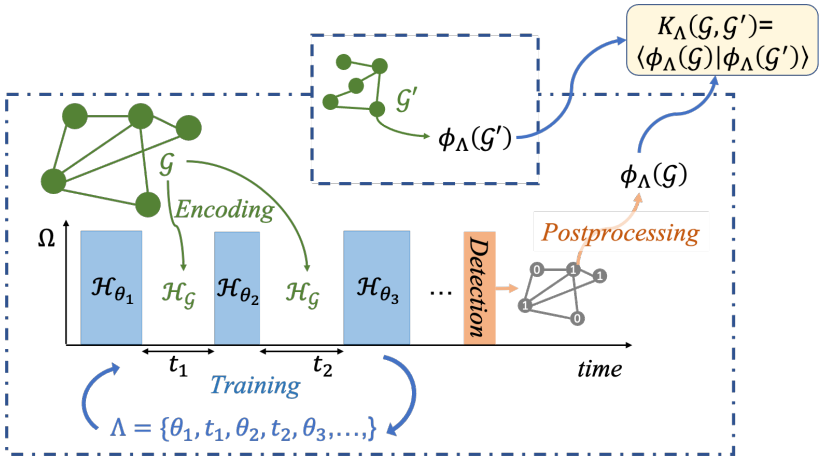

FIG. 2. Schematics of the feature map at the heart of the QE kernel. One first encodes an input graph $\mathcal{G}$ into a Hamiltonian $\hat{\mathcal{H}}_\mathcal{G}$, which is used in a parametrized sequence, alternating evolution with $\hat{\mathcal{H}}_\mathcal{G}$ and pulses with Hamiltonian $\hat{\mathcal{H}}_{\theta_i}$. An observable is measured at the end of the pulse, yielding a bitstring. From this bitstring (or a list of bistrings resulting from repeated iterations of this pulse), a probability distribution is extracted, out of which the graph kernel is computed. $\phi_\Lambda(\mathcal{G})$ is the associated feature vector in the latent space, and is never explicitly computed. The precise form of the pulse sequence is determined through training on a graph data set.

[1] Pascal Scholl, Michael Schuler, Hannah J Williams, Alexander A Eberharter, Daniel Barredo, Kai-Niklas Schymik, Vincent Lienhard, Louis-Paul Henry, Thomas C Lang, Thierry Lahaye, Andreas M Läuchli, and Antoine Browaeys. Quantum simulation of 2D antiferromagnets with hundreds of Rydberg atoms. *Nature*, 595(7866):233–238, 2021. ISSN 1476-4687. doi:10.1038/s41586-021-03585-1. URL https://doi.org/10.1038/s41586-021-03585-1.

[2] Daniel Barredo, Sylvain De Léséleuc, Vincent Lienhard, Thierry Lahaye, and Antoine Browaeys. An atom-by-atom assembler of defect-free arbitrary two-dimensional atomic arrays. *Science*, 354(6315):1021–1023, 2016.

[3] Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces. *Physical review letters*, 122(4): 040504, 2019.

[4] Brian Coyle, Maxwell Henderson, Justin Chan Jin Le, Niraj Kumar, Marco Paini, and Elham Kashefi. Quantum versus classical generative modelling in finance. *Quantum Science and Technology*, 6(2):024013, 2021.

[5] Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Physical Review A*, 103(3):032430, 2021.

[6] Amira Abbas, David Sutter, Christa Zoufal, Aurélien Lucchi, Alessio Figalli, and Stefan Woerner. The power of quantum neural networks. *Nature Computational Science*, 1(6):403–409, 2021.

[7] Leonardo Banchi and Gavin E. Crooks. Measuring analytic gradients of general quantum evolution with the stochastic parameter shift rule. *Quantum*, 5:386, Jan 2021. ISSN 2521-327X. doi:10.22331/q-2021-01-25-386. URL http://dx.doi.org/10.22331/q-2021-01-25-386.

[8] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum circuit learning. *Physical Review A*, 98 (3):032309, 2018.

[9] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on

quantum hardware. *Physical Review A*, 99(3):032331, 2019.

[10] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1263–1272. PMLR, 06–11 Aug 2017. URL http://proceedings.mlr.press/v70/gilmer17a.html.

[11] Pierre Latouche and Fabrice Rossi. Graphs in machine learning: an introduction. *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 207–218, April 2015.

[12] Giulia Muzio, Leslie O'Bray, and Karsten Borgwardt. Biological network analysis with deep learning. *Briefings in Bioinformatics*, 22(2):1515–1530, 11 2020. ISSN 1477-4054. doi:10.1093/bib/bbaa257. URL https://doi.org/10.1093/bib/bbaa257.

[13] Guillaume Verdon, Trevor McCourt, Enxhell Luzhnica, Vikash Singh, Stefan Leichenauer, and Jack Hidary. Quantum graph neural networks. *arXiv preprint arXiv:1909.12264*, 2019.

[14] Giannis Nikolentzos, Giannis Siglidis, and Michalis Vazirgiannis. Graph kernels: A survey. *arXiv preprint arXiv:1904.12218*, 2019.

[15] Louis-Paul Henry, Slimane Thabet, Constantin Dalyac, and Loïc Henriet. Quantum evolution kernel: Machine learning on graphs with programmable arrays of qubits. *arXiv preprint arXiv:2107.03247*, 2021.