QUANTUM FEATURE MAPS FOR GRAPH MACHINE LEARNING ON A NEUTRAL ATOM QUANTUM PROCESSOR

Boris Albrecht , Constantin Dalyac [*], Lucas Leclerc [†], Luis Ortiz-Gutiérrez & Slimane Thabet[*]
Equal contribution.
PASQAL, 7 rue Léonard de Vinci, 91300 Massy, France

Mauro D'Arcangelo , Julia R. K. Cline , Vincent E. Elfving , Lucas Lassablière , Henrique Silvério Bruno Ximenez , Louis-Paul Henry , Adrien Signoles , Loïc Henriet [‡]
PASQAL, 7 rue Léonard de Vinci, 91300 Massy, France

ABSTRACT

Using a quantum processor to embed and process classical data enables the generation of correlations between variables that are inefficient to represent through classical computation. A fundamental question is whether these correlations could be harnessed to enhance learning performances on real datasets. Here, we report the use of a neutral atom quantum processor comprising up to 32 qubits to implement machine learning tasks on graph-structured data. To that end, we introduce a quantum feature map to encode the information about graphs in the parameters of a tunable Hamiltonian acting on an array of qubits. Using this tool, we first show that interactions in the quantum system can be used to distinguish non-isomorphic graphs that are locally equivalent. We then realize a toxicity screening experiment, consisting of a binary classification protocol on a biochemistry dataset comprising 286 molecules of sizes ranging from 2 to 32 nodes, and obtain results which are comparable to the implementation of the best classical kernels on the same dataset. Using techniques to compare the geometry of the feature spaces associated with kernel methods, we then show evidence that the quantum feature map perceives data in an original way, which is hard to replicate using classical kernels.

INTRODUCTION

Representing data in the form of graphs is ubiquitous in many domains of sciences. Using the exponentially large Hilbert space accessible to a quantum computer in order to generate graph embeddings is an appealing idea, with many proposals and theoretical studies over the past few years Schuld & Killoran (2019); Havlíček et al. (2019); Schuld et al. (2020); Kishi et al. (2021). With the recent advances in geometric quantum machine learning Larocca et al. (2022); Skolik et al. (2022), works have shown how graph-structured data could be encoded into quantum states and manipulated for classification, clustering or regression tasks. The Quantum Evolution Kernel (QEK) approach was introduced in 2021 Henry et al. (2021), which is based on evolving a quantum register over alternating layers of (graph-encoding) Hamiltonians, and training for classification tasks. Here, we specifically focus on a QEK-type quantum feature map Henry et al. (2021) for graph-structured data that we experimentally investigate for various learning tasks on a 32-qubit neutral atom Quantum Processing Unit (QPU).

[*] Also affiliated at LIP6, CNRS, Sorbonne Université, 4 Place Jussieu, 75005 Paris, France

[†] Also affiliated at Université Paris-Saclay, Institut d'Optique Graduate School,CNRS, Laboratoire Charles Fabry, 91127 Palaiseau, France

[‡] loic@pasqal.com

1 Quantum feature map for graph-structured data

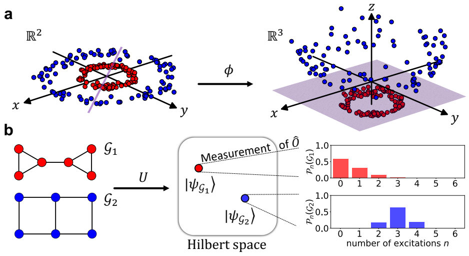
a
$\mathbb{R}^2$
x
y
ϕ
$\mathbb{R}^3$
z
x
y
b
$\mathcal{G}_1$
$\mathcal{G}_2$
U
Measurement of $\hat{O}$
$|\psi_{\mathcal{G}_1}\rangle$
$|\psi_{\mathcal{G}_2}\rangle$
Hilbert space
$\mathcal{P}_n(\mathcal{G}_1)$
$\mathcal{P}_n(\mathcal{G}_2)$
1.0
0.5
0.0
0 1 2 3 4 5 6
number of excitations n

Figure 1: **a**. We seek to find a binary classifier enabling the separation of the two data classes by a hyperplane (purple). After a transformation $\phi((x, y)) = (x, y, x^2 + y^2)$ mapping the data point from $\mathbb{R}^2$ to $\mathbb{R}^3$, the transformed data points become linearly separable. **b**. Illustration of the hardware-implemented quantum feature map $U(\mathcal{G}; t)$ and subsequent measurements for graph-structured data. A parameterized quantum unitary U is applied to atomic registers arranged under the form of UD graphs. Experimentally measured observable distributions are then used for learning tasks.

In many classical machine learning methods, one seeks to map input data into a different space called the feature space using a transform called the feature map, making it easier to work with. In Quantum Machine Learning Schuld & Killoran (2019); Havlíček et al. (2019), the embedding is usually done on a quantum feature space which is a Hilbert space associated with a set of qubits. Such an embedding is usually built from the dynamics of a quantum system depending on the input data as well as external variational parameters.

In this paper, we use a neutral-atom based QPU made of single ^{87}Rb atoms trapped in arrays of optical tweezers Barredo et al. (2018); Nogrette et al. (2014); Browaeys & Lahaye (2020); Henriet et al. (2020); Morgado & Whitlock (2021). The ability to modify the spatial arrangement of the atoms allow us to create a quantum system whose Hamiltonian reflects the topology of various graphs. More details are given in the appendix A. We will restrict ourselves to a set of graphs called Unit Disk (UD) graphs, for which two nodes in the plane are connected by an edge if the distance between them is smaller than a given threshold Starting from a UD graph $\mathcal{G}$ reproduced in the array of tweezers with qubits all starting in $|0\rangle$, we apply a parameterized laser pulse onto the atoms in order to generate a wavefunction $|\psi_{\mathcal{G}}\rangle$ of the form $|\psi_{\mathcal{G}}\rangle = U(\mathcal{G}; t)\,|0\rangle^{\otimes|\mathcal{G}|}$, where we define the time-evolution operator $U(\mathcal{G}; t) = \mathcal{T}\left[\exp\left(-i/\hbar\int_{s=0}^{t}\hat{\mathcal{H}}_{\mathcal{G}}(s)ds\right)\right]$ to be our *quantum feature map* unitary for graph-structured data with $\hat{\mathcal{H}}_{\mathcal{G}}$ the hamiltonian of the system dependent on $\mathcal{G}$

The graph quantum feature map already shows interesting properties when associated with single-body observables $\langle\hat{O}_{j=1,\ldots,|\mathcal{G}|}\rangle$. The measured values are not only affected by local graph properties such as node degrees, but also by more global ones such as the presence of cycles. This enrichment provided by the quantum dynamics contrasts with the locality of node representations in many classical graph machine learning. This key feature comes from the fact that the quantum dynamics of a given spin model (*e.g.* an Ising model) will be significantly influenced by the complete structure of the graph. We illustrate experimentally this behavior for two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ that are non-isomorphic but locally identical. In these graphs, nodes can be separated into two equivalence classes according to their neighborhood: border nodes B have one degree-3 neighbor and one degree-2 neighbor, while center nodes C have two degree-2 neighbors and one degree-3 neighbor (see Fig. 2**a**). We will see that the presence of interactions will enable us to discriminate between $\mathcal{G}_1$ and $\mathcal{G}_2$ by comparing the dynamics of local observables on border and center nodes.

By looking at $\hat{O} = \sum_{i=1}^{6}\hat{n}_i$, we can quantify the difference in the dynamics between the two graphs. We first compute the histogram $\mathcal{P}_i$ of number of excitations observed in each shot on graph $\mathcal{G}_i$. The difference between those graphs is then estimated via the Jensen-Shannon divergence JS of their respective histograms Bai & Hancock (2013), which is defined as $JS(\mathcal{P}_1, \mathcal{P}_2) = H\left(\frac{\mathcal{P}_1+\mathcal{P}_2}{2}\right) - \frac{H(\mathcal{P}_1)+H(\mathcal{P}_2)}{2}$. Here $H(\mathcal{P}) = -\sum_k p_k \log p_k$ is the Shannon entropy of $\mathcal{P} = (p_1, \ldots p_{|\mathcal{G}|})$.

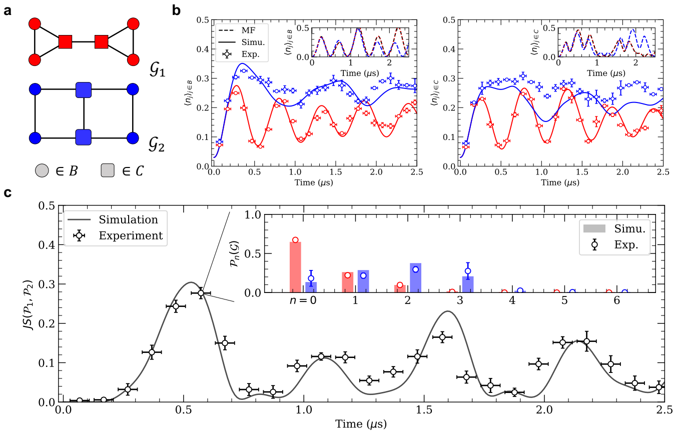

Figure 2: **a**. $\mathcal{G}_1$ (red) and $\mathcal{G}_2$ (blue) are two different graphs with identical local structure. Based on their neighborhood, the nodes either belong to the *border* B (circle) or to the *center* C (square). **b**. We plot the evolution of the mean occupation $\langle n_i \rangle$ of the two regions B (left) and C (right) for both graphs $\mathcal{G}_1$ (red) and $\mathcal{G}_2$ (blue). The dots represent the experimental results while the full curves show noisy simulation results. Horizontal error bars account for the sequence-trigger uncertainty ($\approx$ 40ns) while the vertical ones account for the sampling noise. The insets show the corresponding mean field dynamics (dashed) with only NN (black) or full (colored) interactions. **c**. The evolution of the Jensen-Shannon divergence obtained experimentally (dot) is compared to the noisy simulation (plain). At each point in time, $JS(\mathcal{P}_1, \mathcal{P}_2)$ is computed using the excitation distributions $\mathcal{P}_{1/2} = \{\mathcal{P}_n(\mathcal{G}_{1/2})\}_{n=0\ldots6}$ obtained either numerically (bar) or experimentally (dot). The inset depicts $\mathcal{P}_{1/2}$ obtained at $t \approx 0.57$ μs which yields the maximum value $JS_{max} \approx 0.28$ reached.

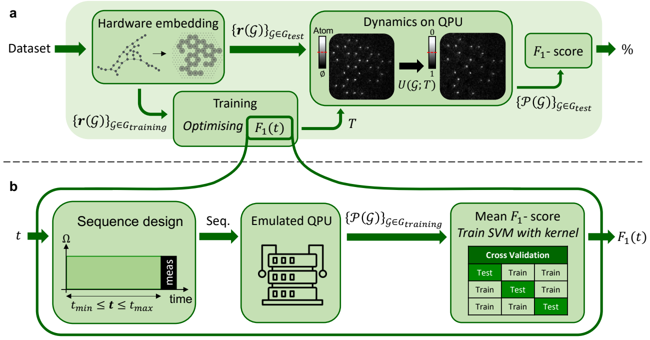
a
Dataset
Hardware embedding
$\{\mathbf{r}(\mathcal{G})\}_{\mathcal{G}\in G_{test}}$
Dynamics on QPU
Atom
$U(\mathcal{G};T)$
F_1 - score
%
$\{\mathcal{P}(\mathcal{G})\}_{\mathcal{G}\in G_{test}}$
$\{\mathbf{r}(\mathcal{G})\}_{\mathcal{G}\in G_{training}}$
Training
Optimising $F_1(t)$
T
b
t
Sequence design
Ω
meas
$t_{min} \leq t \leq t_{max}$
time
Seq.
Emulated QPU
$\{\mathcal{P}(\mathcal{G})\}_{\mathcal{G}\in G_{training}}$
Mean F_1 - score
Train SVM with kernel
Cross Validation
Test
Train
Train
Train
Test
Train
Train
Train
Test
$F_1(t)$

Figure 3: **a**. A dataset of graphs $\mathcal{G}$ is mapped onto atomic registers $\mathbf{r}(\mathcal{G})$, and split between a training and a test set $\mathcal{G}_{training}, \mathcal{G}_{test}$. We use the training set to determine numerically the optimal pulse sequence to be applied on the hardware using a grid search algorithm for optimizing $F_1(t)$ (see **b**). This training phase outputs the optimal parameter T used to design the laser-pulse sequence applied experimentally on each register of the test set. The resulting dynamics performed on the QPU generates $U(\mathcal{G};T)$, driving the system from $|0\rangle^{\otimes \mathcal{G}}$ to $|\psi_{\mathcal{G}}\rangle$. F_1 is then derived from the measured probability distributions $\{\mathcal{P}(\mathcal{G})\}_{\mathcal{G}\in G_{test}}$. **b.** The optimization of the score function F_1 during the training includes several steps. The input t, taken from the parameter space $[t_{min}, t_{max}]$ defines a laser sequence with Ω and δ fixed parameters followed by a measurement. The dynamics of the system is emulated and enables us to compute the probability distributions associated to this particular value of t for the whole training part of the dataset. Finally, $F_1(t)$ is obtained by fitting the SVM with the kernel constructed from those probability distributions.

2 Binary classification task

We now use the graph distance metric introduced in the previous section to tackle a binary classification task on a dataset of chemical compounds called PTC-FM (Predictive Toxicity Challenge on Female Mice) Helma et al. (2001); Siglidis et al. (2018). We use this metric as a graph kernel, and we feed this kernel to a Support Vector Machine (SVM) algorithm. The pipeline is described figure 3. More details can be found in the appendix B.After a training of our model, we experimentally obtain an F_1-score of $60.4 \pm 5.1\%$. For comparison purposes, we examine the performances of other kernels on this dataset: the Graphlet Sampling (GS), Random Walk (RW), Shortest Path (SP) and SVM-ϑ kernels, the size kernel, all these kernels being described in detail in Appendix H. The F_1-scores reached by the various kernels are collected in Table 1. Obtained scores range from $49.8 \pm 6.0\%$ up to $58.2 \pm 5.5\%$. Those results show that the Quantum Evolution Kernel is competitive with standard classical kernels on this dataset. We briefly describe the rest of our analysis and experiments, which are detailed in the appendix B. Since the size of the graph is an important feature of the dataset, we also construct a version of the kernel insensitive to it, called size-compensated (B.1). Although it achieves a poor score, we show that it is competitive compared to other kernels with respect to a metric called geometric difference (B.4, I) and introduced by Huang et al. (2021). We finally construct a synthetic dataset for which QEK supersedes all other classical kernels with the same procedure (B.5).

Kernel	F_1-score (%)	Kernel	F_1-score (%)
QEK	60.4 ± 5.1	Graphlet Sampling	56.9 ± 5.0
QEK (size-compensated)	45.1 ± 3.7	Random Walk	55.1 ± 6.9
SVM-ϑ	58.2 ± 5.5	Shortest Path	49.8 ± 6.0
Size	56.7 ± 5.6		

Table 1: F_1-score reached experimentally on the PTC-FM dataset by QEK (± std. on the splits) and other classical kernels. The values are the average over a 5-fold cross-validation repeated 10 times.

Conclusion

In this paper, we reported the implementation of a quantum feature map for graph-structured data on a neutral atom quantum processor with up to 32 qubits. We experimentally showed that this embedding was not only sensitive to local graph properties but was also able to probe more global structures such as cycles. This property offers a promising way to expand the capabilities of standard GNN architectures, which have been shown to have the same expressiveness as the Weisfeiler-Lehman (WL) Isomorphism test in terms of distinguishing non-isomorphic graphs Morris et al. (2019); Xu et al. (2019). For example, a standard GNN architecture will treat $\mathcal{G}_1$ and $\mathcal{G}_2$ shown in Fig. 2**a** in the same way, as they have the same local structure. Some properties of quantum-enhanced version of GNNs have been explored in Thabet et al. (2022).

We then used the quantum graph feature map for a toxicity screening procedure on a standard biochemistry dataset comprising 286 graphs of sizes ranging from 2 to 32 nodes. This procedure achieved a F_1-score of $60.4 \pm 5.1\%$, on par with the best classical kernels. We intentionally did not include GNNs in the benchmark, as they belong to another distinct family of models. Beyond this pure performance assessment, we showcased the potential advantage of using a quantum feature map through the computation of geometric differences with respect to said classical kernels, which are metrics evaluating the degree of similarity between the kernels' feature spaces, and by artificially constructing a dataset for which the quantum procedure exhibited superior performances.

This proof-of-concept illustrates the potential of quantum-enhanced methods for graph machine learning tasks. Our study paves the way for the incorporation of quantum-enhanced algorithms with standard ML solutions, aiming at constructing better tools for graph data analysis and prediction. Further work on more diverse datasets will be required to assess the viability of the approach compared to powerful state-of-the-art GNN architectures (Gilmer et al., 2017; Ying et al., 2021; Rampášek et al., 2022; Kreuzer et al., 2021). Additionally, our results showcase the power and versatility of neutral atom QPUs, with their ability to change the register geometry from run to run. Going forward, the implementation of similar methods on non-local graphs could be envisaged by embedding them into three-dimensional registers Dalyac & Henriet (2022a) or moving the qubits throughout the course of the computation Bluvstein et al. (2021).

ACKNOWLEDGMENTS

We thank Jacob Bamberger, Lucas Beguin, Antoine Browaeys, Luc Couturier, Romain Fouilland, Thierry Lahaye, Hsin-Yuan Huang, Christophe Jurczak, and Georges-Olivier Reymond for fruitful discussions.

REFERENCES

Lu Bai and Edwin R Hancock. Graph kernels from the jensen-shannon divergence. *Journal of mathematical imaging and vision*, 47(1):60–69, 2013.

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, November 2016. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aah3778. URL https://science.sciencemag.org/content/354/6315/1021.

Daniel Barredo, Vincent Lienhard, Sylvain de Léséleuc, Thierry Lahaye, and Antoine Browaeys. Synthetic three-dimensional atomic structures assembled atom by atom. *Nature*, 561(7721):79–82, September 2018. ISSN 1476-4687. doi: 10.1038/s41586-018-0450-2. URL https://www.nature.com/articles/s41586-018-0450-2.

L. Bé guin, A. Vernier, R. Chicireanu, T. Lahaye, and A. Browaeys. Direct measurement of the van der waals interaction between two rydberg atoms. *Physical Review Letters*, 110(26), jun 2013. doi: 10.1103/physrevlett.110.263201. URL https://doi.org/10.1103%2Fphysrevlett.110.263201.

Dolev Bluvstein, Harry Levine, Giulia Semeghini, Tout T. Wang, Sepehr Ebadi, Marcin Kalinowski, Alexander Keesling, Nishad Maskara, Hannes Pichler, Markus Greiner, Vladan Vuletic, and Mikhail D. Lukin. A quantum processor based on coherent transport of entangled atom arrays. *arXiv e-prints*, art. arXiv:2112.03923, December 2021.

Antoine Browaeys and Thierry Lahaye. Many-body physics with individually controlled rydberg atoms. *Nature Physics*, 16(2):132–142, Feb 2020. ISSN 1745-2481. doi: 10.1038/s41567-019-0733-z. URL https://doi.org/10.1038/s41567-019-0733-z.

Andrew Byun, Minhyuk Kim, and Jaewook Ahn. Finding the maximum independent sets of platonic graphs using rydberg atoms. *PRX Quantum*, 3:030305, Jul 2022. doi: 10.1103/PRXQuantum.3.030305. URL https://link.aps.org/doi/10.1103/PRXQuantum.3.030305.

MacKay David J C. *An Example Inference Task: Clustering*. Cambridge University press, 2004.

Constantin Dalyac and Loic Henriet. Embedding the mis problem for non-local graphs with bounded degree using 3d arrays of atoms, 2022a. URL https://arxiv.org/abs/2209.05164.

Constantin Dalyac and Loic Henriet. Embedding the mis problem for non-local graphs with bounded degree using 3d arrays of atoms, 2022b. URL https://arxiv.org/abs/2209.05164.

Constantin Dalyac, Loic Henriet, Emmanuel Jeandel, Wolfgang Lechner, Simon Perdrix, Marc Porcheron, and Margarita Veshchezerova. Qualifying quantum approaches for hard industrial optimization problems. a case study in the field of smart-charging of electric vehicles. *EPJ Quantum Technology*, 8(1):12, May 2021. ISSN 2196-0763. doi: 10.1140/epjqt/s40507-021-00100-3. URL https://doi.org/10.1140/epjqt/s40507-021-00100-3.

Sylvain de Lé séleuc, Daniel Barredo, Vincent Lienhard, Antoine Browaeys, and Thierry Lahaye. Analysis of imperfections in the coherent optical excitation of single atoms to rydberg states. *Physical Review A*, 97(5), may 2018. doi: 10.1103/physreva.97.053803. URL https://doi.org/10.1103%2Fphysreva.97.053803.

E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, Dec 1959. ISSN 0945-3245. doi: 10.1007/BF01386390. URL https://doi.org/10.1007/BF01386390.

S. Ebadi, A. Keesling, M. Cain, T. T. Wang, H. Levine, D. Bluvstein, G. Semeghini, A. Omran, J.-G. Liu, R. Samajdar, X.-Z. Luo, B. Nash, X. Gao, B. Barak, E. Farhi, S. Sachdev, N. Gemelke, L. Zhou, S. Choi, H. Pichler, S.-T. Wang, M. Greiner, V. Vuletić, and M. D. Lukin. Quantum optimization of maximum independent set using rydberg atom arrays. *Science*, 376(6598): 1209–1215, 2022. doi: 10.1126/science.abo6587. URL `https://www.science.org/doi/abs/10.1126/science.abo6587`.

Robert W. Floyd. Algorithm 97: Shortest path. *Commun. ACM*, 5(6):345, jun 1962. ISSN 0001-0782. doi: 10.1145/367766.368168. URL `https://doi.org/10.1145/367766.368168`.

Thomas M. J. Fruchterman and Edward M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991. doi: https://doi.org/10.1002/spe.4380211102. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.4380211102`.

Alpha Gaëtan, Yevhen Miroshnychenko, Tatjana Wilk, Amodsen Chotia, Matthieu Viteau, Daniel Comparat, Pierre Pillet, Antoine Browaeys, and Philippe Grangier. Observation of collective excitation of two individual atoms in the rydberg blockade regime. *Nature Physics*, 5(2):115–118, Feb 2009. ISSN 1745-2481. doi: 10.1038/nphys1183. URL `https://doi.org/10.1038/nphys1183`.

Thomas Gärtner, Peter Flach, and Stefan Wrobel. On graph kernels: Hardness results and efficient alternatives. In Bernhard Schölkopf and Manfred K. Warmuth (eds.), *Learning Theory and Kernel Machines*, pp. 129–143, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. ISBN 978-3-540-45167-9.

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 (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/gilmer17a.html`.

Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, 2019. doi: 10.1038/s41586-019-0980-2. URL `https://doi.org/10.1038/s41586-019-0980-2`.

Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, Mar 2019. ISSN 1476-4687. doi: 10.1038/s41586-019-0980-2. URL `https://doi.org/10.1038/s41586-019-0980-2`.

C. Helma, R. D. King, S. Kramer, and A. Srinivasan. The Predictive Toxicology Challenge 2000–2001 . *Bioinformatics*, 17(1):107–108, 01 2001. ISSN 1367-4803. doi: 10.1093/bioinformatics/17.1.107. URL `https://doi.org/10.1093/bioinformatics/17.1.107`.

Loïc Henriet. Robustness to spontaneous emission of a variational quantum algorithm. , 101(1): 012335, January 2020. doi: 10.1103/PhysRevA.101.012335.

Loïc Henriet, Lucas Beguin, Adrien Signoles, Thierry Lahaye, Antoine Browaeys, Georges-Olivier Reymond, and Christophe Jurczak. Quantum computing with neutral atoms. *Quantum*, 4:327, September 2020. ISSN 2521-327X. doi: 10.22331/q-2020-09-21-327. URL `https://doi.org/10.22331/q-2020-09-21-327`.

Louis-Paul Henry, Slimane Thabet, Constantin Dalyac, and Loïc Henriet. Quantum evolution kernel: Machine learning on graphs with programmable arrays of qubits. *Phys. Rev. A*, 104:032416, Sep 2021. doi: 10.1103/PhysRevA.104.032416. URL `https://link.aps.org/doi/10.1103/PhysRevA.104.032416`.

Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R. McClean. Power of data in quantum machine learning. *Nature Communications*, 12(1):2631, May 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-22539-9. URL `https://doi.org/10.1038/s41467-021-22539-9`.

D. Jaksch, J. I. Cirac, P. Zoller, S. L. Rolston, R. Cô té, and M. D. Lukin. Fast quantum gates for neutral atoms. *Physical Review Letters*, 85(10):2208–2211, sep 2000. doi: 10.1103/physrevlett.85.2208. URL `https://doi.org/10.1103%2Fphysrevlett.85.2208`.

Minhyuk Kim, Kangheun Kim, Jaeyong Hwang, Eun-Gook Moon, and Jaewook Ahn. Rydberg quantum wires for maximum independent set problems. *Nature Physics*, 18(7):755–759, Jul 2022. ISSN 1745-2481. doi: 10.1038/s41567-022-01629-5. URL `https://doi.org/10.1038/s41567-022-01629-5`.

Kaito Kishi, Takahiko Satoh, Rudy Raymond, Naoki Yamamoto, and Yasubumi Sakakibara. Graph kernels encoding features of all subgraphs by quantum superposition, 2021.

Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems*, 34:21618–21629, 2021.

Niels M. Kriege, Fredrik D. Johansson, and Christopher Morris. A survey on graph kernels. *Applied Network Science*, 5(6), 2020. doi: 10.1007/s41109-019-0195-3. URL `https://doi.org/10.1007/s41109-019-0195-3`.

Martín Larocca, Frédéric Sauvage, Faris M. Sbahi, Guillaume Verdon, Patrick J. Coles, and M. Cerezo. Group-invariant quantum machine learning. *PRX Quantum*, 3:030341, Sep 2022. doi: 10.1103/PRXQuantum.3.030341. URL `https://link.aps.org/doi/10.1103/PRXQuantum.3.030341`.

Sylvain de Leseleuc de kerouara. *Quantum simulation of spin models with assembled arrays of Rydberg atoms*. Theses, Université Paris Saclay (COmUE), December 2018. URL `https://pastel.archives-ouvertes.fr/tel-02088297`.

L. Lovasz. On the shannon capacity of a graph. *IEEE Transactions on Information Theory*, 25(1): 1–7, 1979. doi: 10.1109/TIT.1979.1055985.

Sohir Maskey, Yunseok Lee, Ron Levie, and Gitta Kutyniok. Stability and generalization capabilities of message passing graph neural networks. *CoRR*, abs/2202.00645, 2022. URL `https://arxiv.org/abs/2202.00645`.

Charles A. Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *Journal of Machine Learning Research*, 7(95):2651–2667, 2006. URL `http://jmlr.org/papers/v7/micchelli06a.html`.

M. Morgado and S. Whitlock. Quantum simulation and computing with Rydberg-interacting qubits. *AVS Quantum Science*, 3(2):023501, June 2021. doi: 10.1116/5.0036562.

Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):4602–4609, Jul. 2019. doi: 10.1609/aaai.v33i01.33014602. URL `https://ojs.aaai.org/index.php/AAAI/article/view/4384`.

Minh-Thi Nguyen, Jin-Guo Liu, Jonathan Wurtz, Mikhail D. Lukin, Sheng-Tao Wang, and Hannes Pichler. Quantum optimization with arbitrary connectivity using Rydberg atom arrays. *arXiv e-prints*, art. arXiv:2209.03965, September 2022.

F. Nogrette, H. Labuhn, S. Ravets, D. Barredo, L. Béguin, A. Vernier, T. Lahaye, and A. Browaeys. Single-atom trapping in holographic 2d arrays of microtraps with arbitrary geometries. *Phys. Rev. X*, 4:021034, May 2014. doi: 10.1103/PhysRevX.4.021034. URL `https://link.aps.org/doi/10.1103/PhysRevX.4.021034`.

Hannes Pichler, Sheng-Tao Wang, Leo Zhou, Soonwon Choi, and Mikhail D. Lukin. Quantum Optimization for Maximum Independent Set Using Rydberg Atom Arrays. *arXiv e-prints*, art. arXiv:1808.10816, August 2018.

Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. *arXiv preprint arXiv:2205.12454*, 2022.

Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces. *Phys. Rev. Lett.*, 122:040504, Feb 2019. doi: 10.1103/PhysRevLett.122.040504. URL https://link.aps.org/doi/10.1103/PhysRevLett.122.040504.

Maria Schuld, Kamil Brádler, Robert Israel, Daiqin Su, and Brajesh Gupt. Measuring the similarity of graphs with a gaussian boson sampler. *Physical Review A*, 101(3):032314, 2020.

Jun SHAO and Dongsheng TU. *The Jackknife and Bootstrap*. SPRINGER, 1995.

N. Shervashidze, SVN. Vishwanathan, TH. Petri, K. Mehlhorn, and KM. Borgwardt. Efficient graphlet kernels for large graph comparison. In *JMLR Workshop and Conference Proceedings Volume 5: AISTATS 2009*, pp. 488–495, Cambridge, MA, USA, April 2009. Max-Planck-Gesellschaft, MIT Press.

N. Šibalić, J.D. Pritchard, C.S. Adams, and K.J. Weatherill. ARC: An open-source library for calculating properties of alkali rydberg atoms. *Computer Physics Communications*, 220:319–331, November 2017. doi: 10.1016/j.cpc.2017.06.015. URL https://doi.org/10.1016/j.cpc.2017.06.015.

Giannis Siglidis, Giannis Nikolentzos, Stratis Limnios, Christos Giatsidis, Konstantinos Skianis, and Michalis Vazirgiannis. Grakel: A graph kernel library in python, 2018. URL https://arxiv.org/abs/1806.02193.

Andrea Skolik, Michele Cattelan, Sheir Yarkoni, Thomas Bäck, and Vedran Dunjko. Equivariant quantum circuits for learning on weighted graphs. 2022. doi: 10.48550/ARXIV.2205.06109. URL https://arxiv.org/abs/2205.06109.

Slimane Thabet, Romain Fouilland, and Loic Henriet. Extending Graph Transformers with Quantum Computed Aggregation. *arXiv e-prints*, art. arXiv:2210.10610, October 2022.

S.V.N. Vishwanathan, Nicol N. Schraudolph, Risi Kondor, and Karsten M. Borgwardt. Graph kernels. *Journal of Machine Learning Research*, 11(40):1201–1242, 2010. URL http://jmlr.org/papers/v11/vishwanathan10a.html.

Jonathan Wurtz, Pedro L. S. Lopes, Nathan Gemelke, Alexander Keesling, and Shengtao Wang. Industry applications of neutral-atom quantum computing solving independent set problems. *arXiv e-prints*, art. arXiv:2205.08500, May 2022.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34:28877–28888, 2021.

A Quantum feature map for graph-structured data

In this paper, we use a neutral-atom based QPU made of single ^{87}Rb atoms trapped in arrays of optical tweezers Barredo et al. (2018); Nogrette et al. (2014); Browaeys & Lahaye (2020); Henriet et al. (2020); Morgado & Whitlock (2021). The qubits are encoded into the ground state $|0\rangle = \left|5S_{1/2}, F=2, m_F=2\right\rangle$ and a Rydberg state $|1\rangle = \left|60S_{1/2}, m_J=1/2\right\rangle$.

When promoted to Rydberg states, the atoms behave as large electric dipoles and thus experience dipole-dipole interactions which boils down, for the chosen Rydberg level, to Van der Waals terms only. The dynamics of a set of N qubits at positions $\{\mathbf{r}_i\}_{i=1\ldots N}$ is thus governed by the following Hamiltonian:

$$\hat{\mathcal{H}} = \hbar \sum_{i=1}^{N} \left(\frac{\Omega}{2} \hat{\sigma}_i^x - \delta \, \hat{n}_i \right) + \sum_{i<j} \frac{C_6}{|\mathbf{r}_i - \mathbf{r}_j|^6} \hat{n}_i \hat{n}_j \tag{1}$$

where $\hat{\sigma}_i^\alpha$ are Pauli matrices and $\hat{n}_i = (1 + \hat{\sigma}_i^z)/2$. $|\mathbf{r}_i - \mathbf{r}_j|$ is the distance between qubits i and j, and $C_6/h \simeq 138$ GHz$\cdot\mu$m^6 for the Rydberg state considered Bé guin et al. (2013); Šibalić et al. (2017). Controlling both intensities and frequencies of each laser field, we can effectively drive the qubit register uniformly with time-dependent tunable Rabi frequency Ω and detuning δ.

Key to our study is the programmability of the qubit register's geometry. In neutral atom processors, one can modify the spatial arrangement of qubits Barredo et al. (2016; 2018) and reproduce the geometrical shape of various graphs with atoms in tweezers. We will restrict ourselves to a set of graphs called Unit Disk (UD) graphs, for which two nodes in the plane are connected by an edge if the distance between them is smaller than a given threshold. UD graphs are intimately related to Rydberg physics through the mechanism of Rydberg blockade Jaksch et al. (2000); Gaëtan et al. (2009); Browaeys & Lahaye (2020), where an atom excited to a Rydberg state prevents other neighboring atoms to be excited within a certain blockade radius. For an atomic register reproducing a UD graph $\mathcal{G}$, the $1/r^6$ power law of the van der Waals interactions effectively restricts, in a good approximation, the summation in the third term of Eq. equation 1 to pairs of indices (i, j) sharing an edge in $\mathcal{G}$. The topology of the interaction term in Eq. equation 1 then becomes the one of the graph under consideration, giving rise to a graph-dependent Hamiltonian $\hat{\mathcal{H}}_\mathcal{G}$. This property has notably been harnessed for solving combinatorial graph optimization problems Pichler et al. (2018); Henriet (2020); Dalyac et al. (2021); Nguyen et al. (2022); Ebadi et al. (2022); Kim et al. (2022); Byun et al. (2022); Wurtz et al. (2022); Dalyac & Henriet (2022b).

Starting from a UD graph $\mathcal{G}$ reproduced in the array of tweezers with qubits all starting in $|0\rangle$, we apply a parameterized laser pulse onto the atoms in order to generate a wavefunction $|\psi_\mathcal{G}\rangle$ of the form

$$|\psi_\mathcal{G}\rangle = U(\mathcal{G}; t) \, |0\rangle^{\otimes|\mathcal{G}|}, \tag{2}$$

where we define the time-evolution operator $U(\mathcal{G}; t) = \mathcal{T}\left[\exp\Big(-i/\hbar \int_{s=0}^{t} \hat{\mathcal{H}}_\mathcal{G}(s) ds\Big)\right]$ to be our *quantum feature map* unitary for graph-structured data. Throughout this paper, we will restrict ourselves to laser pulses with constant detuning δ and Rabi frequency Ω, with an adjustable duration t. Depending on the task at hand, we consider various observables $\hat{O}$ to evaluate on $|\psi_\mathcal{G}\rangle$. Measurements of a site-dependent (respectively global) observable give rise to a probability distribution $\mathcal{P}$ which is node (graph) specific and can be used for various machine learning tasks at the node (graph) level. In the following, we show theoretically and experimentally that the graph quantum feature map already shows interesting properties when associated with local or global observables built from single-body expectation values $\langle \hat{O}_{j=1,\ldots,|\mathcal{G}|} \rangle$.

B Binary classification task

B.1 QEK and size-compensated QEK

In order to realize the classification task on this dataset, we need to turn the quantum graph embedding introduced in Sec. 1 into a kernel K, the Quantum Evolution Kernel by following the approach originally proposed in Ref. Henry et al. (2021). We compute the distributions $\mathcal{P}$ of the total number of Rydberg excitations $\sum_j \hat{n}_j$ measured in the final state on graph $\mathcal{G}$ and we use again the Jensen-Shannon divergence from section **??**sec:embeddings to build a kernel out of those distributions.

$$K(\mathcal{G}, \mathcal{G}') = \exp\left[-JS(\mathcal{P}, \mathcal{P}')\right]. \tag{3}$$

This kernel is well-defined, *i.e.* the kernel matrix is always positive definite Bai & Hancock (2013).

Going forward, we modify the QEK procedure in order to make the kernel insensitive to size. To that end, we compare the measurement distributions obtained for different graph sizes using a convolution operation. Let us consider two graphs $\mathcal{G}_i$ and $\mathcal{G}_j$ of N_i and $N_j = N_i + \Delta N > N_i$ nodes

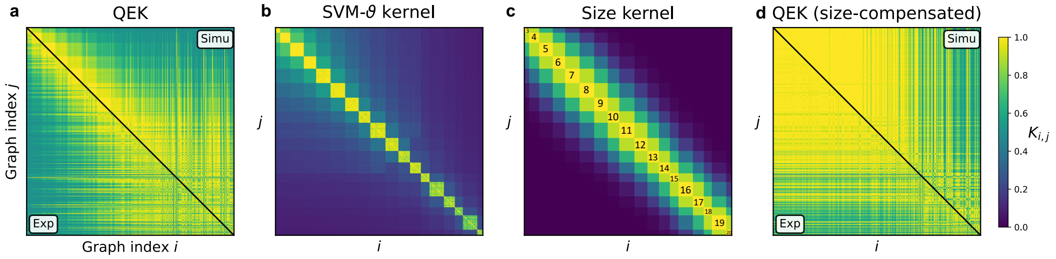

Figure 4: Each kernel is represented by a $M \times M$ matrix where $K_{i,j} = K(\mathcal{G}_i, \mathcal{G}_j)$ as defined in Eq. equation 3. The graph indices are sorted by increasing size. A separation (black line) is drawn between numerically simulated (top right) and experimentally measured (bottom left) QEK matrices. **a.** QEK kernel obtained using directly the raw distributions $\mathcal{P}_i$ and $\mathcal{P}_j$. **b.** Kernel obtained via SVM-ϑ method. **c.** Size kernel obtained with $K^{\text{size}}(\mathcal{G}_i, \mathcal{G}_j) = \exp\left(-\gamma(|\mathcal{G}_i| - |\mathcal{G}_j|)^2\right)$ with $\gamma = 0.1$. **d.** QEK kernel obtained using modified distributions $\tilde{\mathcal{P}}_i$ and $\mathcal{P}_j$, where graphs of smaller sizes are convoluted with binomial distributions when compared to larger graphs.

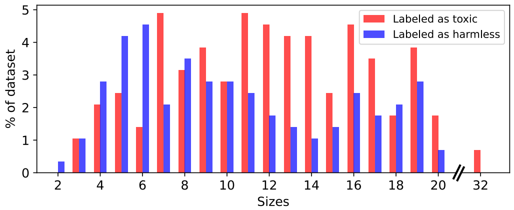

Figure 5: The PTC-FM dataset exhibits a strong size imbalance. For small number of nodes (10) more graphs are labeled as harmless (blue) while it is the opposite for larger graphs, more prone to be labeled as toxic (red).

respectively; and note their respective observable distributions $\mathcal{P}_i$ and $\mathcal{P}_j$. From $\mathcal{P}_i$ we construct $\tilde{\mathcal{P}}_i = \mathcal{P}_i \star b_{\Delta N}^{(i/j)}$ the convolution of $\mathcal{P}_i$ and a binomial distribution :

$$b_{\Delta N}^{(p)}(n) = \binom{\Delta N}{n} p^n (1-p)^{\Delta N - n}. \tag{4}$$

$\tilde{\mathcal{P}}_i$ corresponds to the distribution one would get by adding to the graph ΔN non-interacting qubits, submitted to the same laser pulse as the other. Each of these isolated qubits undergoes Rabi oscillations, induced by the applied pulse sequence. They are therefore measured either in $|0\rangle$ with probability p or in $|1\rangle$ with probability $1-p$, where $p = \sin^2(\pi\Omega T)$ (≈ 0.768 here). We finally define the modified graph kernel as

$$K_{conv}(\mathcal{G}_i, \mathcal{G}_j) = \exp\left[-JS(\tilde{\mathcal{P}}_i, \mathcal{P}_j)\right]. \tag{5}$$

Using this procedure on the data obtained experimentally, we obtain the kernel matrix shown in Fig. 4**d**, with a corresponding F_1-score of $45.1 \pm 3.7\%$. If this size-compensated version of QEK had been implemented without interaction between atoms, its score would be 42%, which is the lowest score reachable by any kernel. We therefore see that this version of QEK cannot capture useful features beyond the graph size, meaning that the presence of interactions by itself is not sufficient to produce an interesting kernel for the task at hand. While the size-compensated QEK does not give results that are comparable with classical kernels, we study in the following part its expressive power, and show that the geometry induced by this method is hardly reproducible by a classical kernel.

We show in Fig. 4**a** the kernel matrix associated with QEK, with indices sorted by increasing size of the graphs. Using the same noise model as in the previous section, we find adequate agreement between the numerically $\mathcal{P}^{num}$ and experimentally $\mathcal{P}^{exp}$ obtained data.

B.2 Dataset and mapping on hardware

In the original PTC-FM dataset, the 349 molecules are represented under the form of graphs where each node is labeled by atomic type and each edge is labeled according to its bond type. We first truncate the dataset to small graph sizes in order to be able to train the kernel in reasonable time. For the $M = 286$ remaining graphs of this dataset, we take into account the adjacency matrix of the graphs representing the compounds and discard the nodes and edges labels. Note that the results of our implementation are therefore not directly comparable to kernel results in the literature which take into account edge and node labels (see Ref. Kriege et al. (2020) for example).

Each node of a graph will be represented by a qubit in the QPU. We first need to determine the positions of these qubits, in order to implement an interaction term in Eq. equation 1 that effectively reflects the graph topology. To this end we design a local optimizer detailed in Appendix G to estimate in free space a preliminary 2D layout for each graph. Starting from a Reingold-Fruchterman layout Fruchterman & Reingold (1991), our optimizer minimizes the average distance between two connected nodes while maximizing the distances between unconnected nodes. Then taking advantage of our ability to tailor the spatial disposition of the tweezers generated by a Spatial Light Modulator (SLM) to fit the optimized layout, we can replicate the graph in the hardware. Following a batching method also detailed in Appendix G, we group similar graphs and superimpose them on the same SLM pattern, effectively mapping the whole dataset on only 6 different SLM patterns over a triangular grid. We therefore reduce the time needed to implement the whole dataset on the QPU.

B.3 Model training

To test the performance of our implementation, we perform a standard procedure called cross-validation. Cross-validation consists of dividing the dataset in 5 equal parts called ‘splits’, and using each split for testing while the rest of the dataset is used for training. During the training phase, we construct for each pulse duration t the corresponding kernel and train a SVM model with it. We then evaluate the F_1-score on the part of the dataset that was left as a test set. We repeat the splitting 10 times, and the cross-validation score is defined as the average of the F_1-score of each split (50 splits in total). We perform a grid search on the penalty hyperparameter C of the SVM on the range $[10^{-3}, 10^{3}]$ such that the final score of a given pulse is the best cross-validation score among the tested values of C, see Appendix F for details.

Including graphs with sizes $|\mathcal{G}| \leq 20$, we numerically compute the score for a nearest-neighbor distance of $r_{NN} = 5.6\,\mu$m and a resonant constant pulse with fixed $\Omega/2\pi = 1$ MHz, and we vary its duration between $t_{min} = 0.1\,\mu$s and $t_{max} = 2.5\,\mu$s. We select the optimal duration $T = 0.66\,\mu s$ that exhibits the maximum F_1-score. We then implement this pulse on the QPU. The whole process is illustrated in Fig. 3.

B.4 Geometric test with respect to classical kernels

In order to obtain an advantage over classical approaches it is not sufficient to implement a quantum feature map based on quantum dynamics that are hard to simulate classically. As shown in Huang et al. (2021), classical ML algorithms can in certain instances learn efficiently from intractable quantum evolutions if they are allowed to be trained on data. The authors consequently propose another metric between kernels in the form of an asymmetric metric function called the geometric difference g_{12}. It compares two kernels K_1 and K_2 in the following way:

$$g_{12} = \sqrt{||\sqrt{K_2}\,(K_1)^{-1}\,\sqrt{K_2}||_\infty} \tag{6}$$

where $||.||_\infty$ is the spectral norm. Intuitively, g_{12} measures the difference between how kernels K_1 and K_2 perceive the relation between data. Precisely, it characterizes the disparity regarding how each of them maps data points to their respective feature spaces. In our case, we take K_1 to be the size-compensated QEK K_{conv}, and K_2 is selected from a set of classical kernels. If the geometric difference is small, it means that there exists no underlying function mapping the data to the targets for which K_{conv} outperforms the classical kernel. On the other hand, a high geometric difference between a quantum and a classical kernel guarantees that there exists such a function for which the quantum model outperforms the classical one. Estimating the geometric difference is therefore a sanity check to stating that the encoding of data to the feature space through the Quantum Evolution Kernel could not be closely replicated by a classical model.

We compute the geometric difference between QEK and various classical kernels over the PTC-FM dataset and report the results in Table 2. The threshold for a high geometric difference is typically taken to be $\sqrt{M}$, where M is the size of the dataset. Here, the obtained g_{12} is always far beyond $\sqrt{M} \sim 10^1$, indicating that the embedding of data through our quantum-enhanced kernel is not trivial and cannot be replicated by a classical machine learning algorithm.

To summarize, while the F_1-score on PTC-FM is rather similar using quantum or classical models, we see nonetheless that the geometry created by our quantum model is non-trivial. A possible interpretation of the non-superiority of quantum approaches on PTC-FM would be that the relationship between the data and the targets is not better captured by our quantum model, although its feature space is not reproducible by classical means. To further confirm this understanding, we find a function that increases and even maximizes the utility of our rich quantum feature space. We build such a function by artificially relabeling the targets according to a procedure presented in Huang et al. (2021) and outlined in Appendix I. We observe that QEK, without retraining, retains an F_1-score of around 99% on the relabeled dataset, while the closest classical kernel reaches a score of at most 82% even after retraining it on the new labels. The results are summarized in Table 3, where the difference in F_1-score between QEK and the various classical kernels is shown.

Geometric Difference w.r.t. QEK	
SVM-ϑ	10^3
Size	10^5
Graphlet Sampling	10^4
Random Walk	10^5
Shortest Path	10^5

Table 2: Order of magnitude of the geometric difference between QEK and various classical kernels.

F_1-score gap (%) w.r.t. QEK (relabeled)	
SVM-ϑ	17.2 ± 4.5
Size	17.8 ± 4.2
Graphlet Sampling	20.1 ± 4.5
Random Walk	17.3 ± 4.3
Shortest Path	18.2 ± 4.4

Table 3: Gap in F_1-score between QEK and various classical kernels after relabeling the dataset.

In light of the geometric difference assessment and the observed gap of F_1-score between QEK and classical kernels on an artificial function, it remains an open question to generally characterize which types of dataset naturally offer a structure that better exploits the geometry offered by our quantum model, without requiring artificial tweaking of the labels. In the following section, we present a synthetic dataset on which QEK is able to outperform classical methods without any relabeling.

B.5 Synthetic dataset

This binary classification dataset is created by sampling weighted random walks on a triangular lattice. In class A, sites belonging to a honeycomb-type sublattice are favored. They are explored with a weight $p_0 = 1$ while the rest of the triangular lattice sites are explored with a weight $p < 1$. Class B is constructed in a similar fashion, but taking a kagome instead of a honeycomb sublattice. The construction of this artificial dataset is illustrated in Fig. 6. In the case where $p = 0$, the differences in their local structure make the two classes easily distinguishable. However, with increasing p, their local structure becomes more and more similar, as additional triangular lattice sites are incorporated. When p is large enough, a lot of triangular local substructures are shared by the two classes, rendering them potentially hard to distinguish by classical methods. At $p = 1$, the underlying triangular lattice is explored uniformly, rendering the datasets indistinguishable.

Building on our ability to distinguish between graphs with similar local structure but globally distinct, we apply QEK on this synthetic dataset. We expect our method to be hardly affected by the presence of sparse defects and therefore be able to outperform classical approaches.

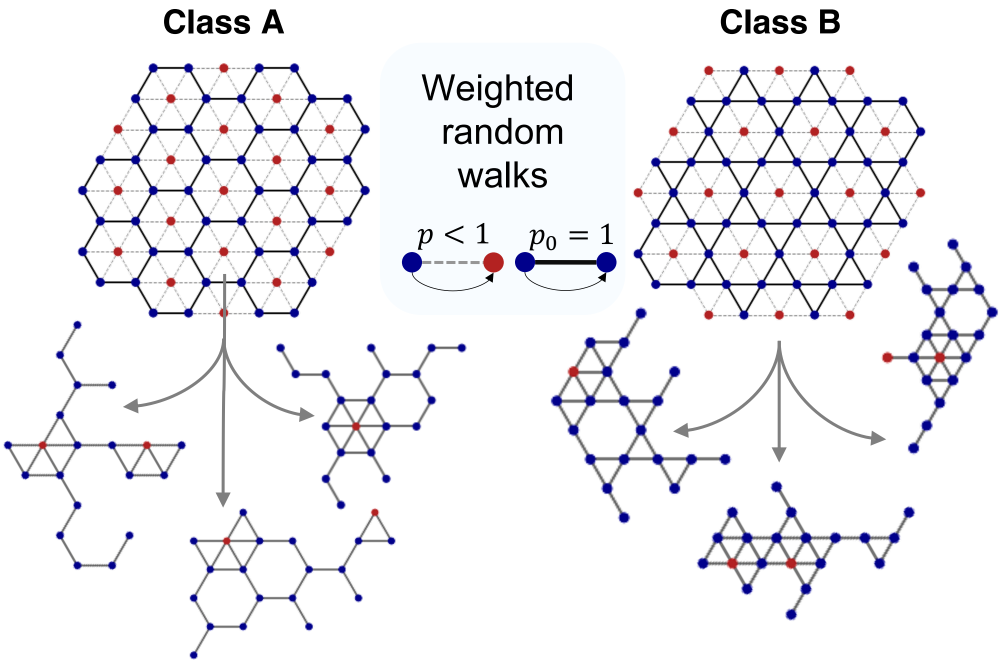

Figure 6: Graphs in Class A contains honeycomb sites (blue) with inclusions of non-honeycomb sites (red) with probability p. Graphs in Class B contains kagome sites (blue) with inclusions of non-kagome sites (red) with probability p. We show examples of generated graph with the aforementioned process.

We investigate numerically this assumption, for several values of p. In each case, we create 200 graphs of 20 nodes each, 100 in each class. The graphs are mapped to a triangular lattice with 5 µm spacing. Here, we consider two alternative schemes of pulse sequences. The first one remains almost the same as the experimentally implemented one, *i.e.* a unique resonant pulse of $\Omega/2\pi = 2$ MHz with parameterized duration up to $8\,\mu$s. The second one is an alternate layer scheme with 4 parameters as described in Henry et al. (2021), where we evaluate 500 random values of the parameters and select the best one. The procedure is designed such that it would be directly implementable on the hardware, as we did for the PTC-FM dataset. We then compare the F_1-score reached by QEK to those reached by other classical kernels, namely: SVM-ϑ, GS, RW and SP. The results are summarized in Fig. 7. With decreasing proportion of defects, all methods perform increasingly better, as expected. Overall, regarding the mean F_1-score reached, the two QEK schemes outperform the four other classical kernels tested for all $p \leq 0.5$. Noticeably, at $p = 0.1$ (*resp* $p = 0.2$), the mean gap in F_1-score between the QEK scheme and the the best classical scheme is 4.5% (*resp* 7.1%) while the mean gap obtained with the alternate QEK scheme is even larger with 13.7% (*resp* 21%), thus showing that QEK can significantly surpass classical approaches on certain types of datasets. When adding too many defects, *i.e.* $p = 0.5$, our Quantum Evolution Kernel exhibits similar performance to the SVM-ϑ.

C Experimental setup

The experimental setup is shown schematically in Fig. 8. It features a magneto-optical trap (MOT), able to cool down and confine a cloud of ^{87}Rb atoms in order to load an array of optical tweezers. They are created by shining a 849 nm laser beam on a Spatial Light Modulator (SLM), and then focusing the beam to a small waist of $\sim 1\,\mu$m with a high numerical aperture (NA) optical system inside the vacuum chamber. The loading of the optical tweezers is stochastic with a probability

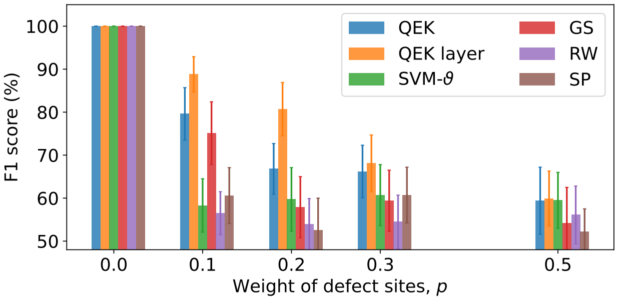

Figure 7: F_1-score (%) reached on the synthetic dataset for different probabilities p of including non-sublattice sites, by the Quantum Evolution Kernel (the alternate scheme is noted QEK layer) as well as by the best SVM-ϑ, GS, RW and SP kernels. The values reported are the average over a 5-fold cross-validation repeated 10 times. Each kernel reaches a F_1-score of 100% when $p = 0$.

$\eta \approx 0.55$ of obtaining one atom per trap. Hence, at each repetition cycle of the experiment, we use a dynamical optical tweezer to move the atoms one by one in order to generate the targeted graph.

The atoms are embedded into a $10\,\mathrm{G}$ magnetic field that sets the quantization axis. The qubits are encoded into the ground state $|0\rangle = \left|5S_{1/2}, F = 2, m_F = 2\right\rangle$ and a Rydberg state $|1\rangle = \left|60S_{1/2}, m_J = 1/2\right\rangle$ of the atoms. They are initialized in the ground state by optical pumping. The qubit transition is then addressed by a two-photon laser excitation, via an intermediate state $6P_{3/2}$. The first (respectively second) photon excitation is generated by a $420\,\mathrm{nm}$ ($1013\,\mathrm{nm}$) σ^+-polarized (σ^--polarized) laser beam with a $1/e^2$ waist radius of $260\,\mu$m ($180\,\mu$m). The two lasers being far-detuned from the intermediate state by $700\,\mathrm{MHz}$, we avoid spurious populating of this state and the three-level system can be faithfully approximated by an effective two-level system. The qubits state is readout in a single step by fluorescence imaging close to resonance at $780\,\mathrm{nm}$, using an EMCCD camera with an integration time of $20\,\mathrm{ms}$.

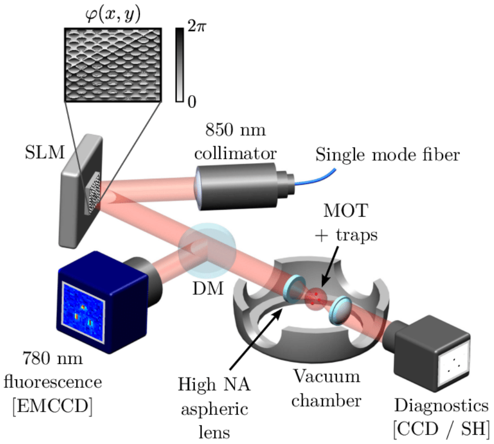

Figure 8: Figure from Nogrette et al. (2014). Microtraps for capturing single atoms are generated by using a SLM. A calculated phase pattern is printed on the $849\ \mathrm{nm}$ laser beam and then focused by the first of two high numerical aperture lenses on the middle of a MOT. The atomic fluorescence at $780\ \mathrm{nm}$ is reflected by a dichroic mirror (DM) and detected with an EMCCD camera. A second aspheric lens (identical to the first) collects the $849\ \mathrm{nm}$ light for, three kind of images, layout loading (tweezers loading quality), register validation (rearrangement successfulness) and register readout (Rydberg excitation discrimination results). The transmitted beam is used for trap diagnostics via a CCD camera or a Shack-Hartmann wavefront sensor (SH).

A set of eight electrodes in an octupole configuration provides active control of the electric field environment around the Rydberg atoms. The durations and shapes of the Rydberg pulses are defined using acousto- and electro-optic modulators, in order to ensure the correct pulse-length used on the measurements.

D Noise model

Despite the precise calibration of the control devices which enable to monitor quantities such as the SLM pattern spacing or the pulse shapes, several experimental imperfections may alter the data measured on the experiment. All experimental data obtained during this study, including those presented in Fig. 2 and Fig. 4, are uncorrected and thus needs to be benchmarked with respect to their simulated counterpart, taking into account the following main sources of noise.

First and foremost, due to the probabilistic nature of the quantum state and the limited budget of shots, measurements are subject to sampling noise. For instance, on average, each of the 25 experimental points on Fig. 2 is obtained using 600 shots and the uncertainty related to this effect (vertical error bars) can be estimated using the Jackknife resampling method SHAO & TU (1995).

The finite sampling is also inherently flawed by several physical processes like atoms thermal motion, background-gas collisions or Rydberg state finite lifetime, whose effects can all be encompassed as first approximation into two detection error terms, ε and ε'. ε (*resp* ε') yield the probability to get false positive (*resp* negative), *i.e.* measure an atom in $|0\rangle$ (*resp* $|1\rangle$) as being in $|0\rangle$ (*resp* $|1\rangle$). ε can be measured with a regular release-and-recapture experiment and ε' with a more advanced method Leseleuc de kerouara (2018) involving π and pushout pulses. To replicate the probabilistic effect of detection errors, the simulated distributions of bitstrings are altered using the following rule to compute the probability of measuring j instead of i:

$$P_{j|i} = \prod_{k} \left(1 - |i-j|_k\right) - (-1)^{|i-j|_k} \left[(1-i_k)\varepsilon + i_k\varepsilon'\right]. \tag{7}$$

$i, j \in \mathbb{B}^N$, $i_k = 0$ (*resp* 1) if atom k is in $|0\rangle$ (*resp* $|1\rangle$). On our device, we measure $\varepsilon \approx 3\%$ and $\varepsilon' \approx 8\%$; thus as an example, we can compute $P_{1001|0101} = \varepsilon\varepsilon'(1-\varepsilon)(1-\varepsilon') \approx 0.2\%$. Those detection errors can deeply modify the measured excitation distributions, with a noticeable effect shown on Fig. 2**b** at $t = 0$ where the simulated $\langle n_j \rangle$ does not start at 0 despite $|\psi(t=0)\rangle = |0\ldots0\rangle$.

Additional errors can also lead to decoherence in the system de Lé séleuc et al. (2018), affecting the atom dynamics in ways costly to emulate. For instance, since the Rydberg transition used is addressed by a two-photon process, misalignments and power fluctuations of the two lasers are twice as likely to occur. Atoms are subject to positional disorder between each shot and their finite velocities make them sensitive to the Doppler effect. Since taking all those effects into consideration becomes quickly intractable, they were only individually simulated in order to assess their limited action on the implemented protocols. However, in order to replicate the experimental data presented in Fig. 2, we resort to an effective decoherence model in the form of solving the Master equation with a relaxation rate of $2\pi \times 0.06$ MHz Bé guin et al. (2013). This value was obtained by fitting with the above model damped Rabi oscillations measured on the same device. Thus, reaching similar behaviour within error bars between numerically simulated and experimentally obtained $JS(\mathcal{P}_1, \mathcal{P}_2)$ was achieved with no free parameter.

E Message passing neural networks

Message passing neural networks (MPNN) Gilmer et al. (2017) is a widely used family of graph neural networks. It was one of the first ones to be developed for graph-structured data, and is still one of the most successful Maskey et al. (2022). It consists of GNNs where the update is made only by aggregating the features of nearest neighbors. In this scheme, the nodes features are multiplied by a trainable weight matrix at each layer, and each node aggregates as a 'message' the features of its neighbors, as illustrated in Fig. 9.

MPNN are closely related to Weisfeiler-Lehman algorithms. In particular, they have been proven to be at most as powerful in distinguishing graph structures Xu et al. (2019). In their standard form, they are then also limited to capture only local features of graphs.

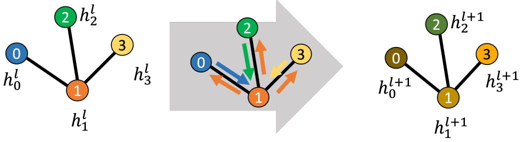

Figure 9: In message passing neural networks, node features vector h_i^l are iteratively updated, from one layer l to the next $l+1$, using only neighboring nodes, similarly to what is done in the WL test.

F Support Vector Machine (SVM) algorithm

The SVM algorithm aims at splitting a dataset into two classes by finding the best hyperplane that separates the data points in the feature space, in which the coordinates of each data point (here each graph) is determined according to the kernel K.

For a training graph dataset $\{\mathcal{G}_i\}_{i=1\ldots M}$, and a set of labels $\mathbf{y} = \{y_i\}_{i=1\ldots M}$ (where $y_i = \pm 1$ depending on which class the graph $\mathcal{G}_i$ belongs to), the dual formulation of the SVM problem consists in finding $\tilde{\boldsymbol{\alpha}} \in \mathcal{A}_C(\mathbf{y}) = \left\{\boldsymbol{\alpha} \in [0, C]^M \,\middle|\, \boldsymbol{\alpha}^T \mathbf{y} = 0\right\}$ such that

$$\frac{1}{2}\tilde{\boldsymbol{\alpha}}^T Q \tilde{\boldsymbol{\alpha}} - \mathbf{e}^T\tilde{\boldsymbol{\alpha}} = \min_{\boldsymbol{\alpha}\in\mathcal{A}_C(\mathbf{y})} \left\{\frac{1}{2}\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} - \mathbf{e}^T\boldsymbol{\alpha}\right\} \tag{8}$$

where $\mathbf{e}$ is the vector of all ones, Q is a $M \times M$ matrix such that $Q_{ij} = y_i y_j K(\mathcal{G}_i, \mathcal{G}_j)$, and $C > 0$ is the penalty hyperparameter, to be adjusted. Setting C to a large value increases the range of possible values of α and therefore the flexibility of the model. But it also increases the training time and the risk of overfitting.

The data points for which $\tilde{\alpha}_i > 0$ are called support vectors (SV). Once the α_i are trained, the class of a new graph $\mathcal{G}$ is predicted by the decision function, given by:

$$y(\mathcal{G}) = \text{sgn}\left\{\langle\phi(\mathcal{G})|\phi_0|\phi(\mathcal{G})|\phi_0\rangle\right\} \tag{9}$$

$$= \text{sgn}\left\{\sum_{i\in SV} y_i\tilde{\alpha}_i K(\mathcal{G}, \mathcal{G}_i)\right\}, \tag{10}$$

with

$$\phi_0 = \sum_{i\in SV} y_i\tilde{\alpha}_i \phi(\mathcal{G}_i). \tag{11}$$

In this case, the training of the kernel amounts to finding the optimal feature vector ϕ_0. It is worth noting that in many cases, Eq. equation 10 is evaluated directly, without explicitly computing ϕ_0.

G Mapping and Batching

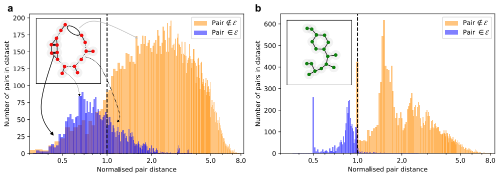

Figure 10: Histograms of normalised pairwise distances between atoms in the 286 graphs of the truncated dataset when performing the embedding with **a.** only a Fruchterman-Reingold layout or **b.** when adding a local optimization step afterwards. For a given graph (insets), two atoms forming a pair $\in \mathcal{E}$ (blue) can be close enough to form a bond via interaction (plain) or too far, creating a missing bond (dotted). Likewise, two atoms forming a pair $\notin \mathcal{E}$ can be placed too close and form a fake edge (thick line).

We present in detail our method to embed the graphs of the PTC-FM dataset. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph of the dataset for which we have a layout of the nodes. Embedding the graph amounts to replace its nodes with atoms, the latter interacting between themselves with the $1/R^6$ dependence. Moving two atoms slightly apart can therefore drastically reduce their interaction strength but it remains non-zero. In order for the Hamiltonian to reflect the topology of $\mathcal{G}$, this $1/R^6$ dependence needs to be approximated by the Heaviside function defined as:

$$h(r) = \begin{cases} \infty & \text{if } r \leq r_b \\ 0 & \text{else} \end{cases} \tag{12}$$

For the Heaviside approximation to be correct, we have to ensure that the largest distance between a pair sharing an edge in the graph is always far less than the shortest distance between a pair not sharing an edge. In other words, in theory, $\min\{U_{ij}, (i,j) \in \mathcal{E}\} / \max\{U_{ij}, (i,j) \notin \mathcal{E}\} \gg 1$.

We use a local optimizer to maximize this ratio and find good solutions in polynomial time. The method optimizes the position of each node in turn, depending on the previously mapped nodes and the presence of cycles in the graph. For the dataset used in this study, we achieve a significant increase of the mean ratio up to 16.8, starting from 5.9 with the classical Fruchterman-Reingold layout. We report that more than half the dataset exhibits a ratio higher than 10 and less than 5% of the dataset is embedded with some defects, *i.e.* a ratio smaller than 1. We also assess the benefit of this approach in Fig. 10 by comparing the distributions of distance of pairs $\in \mathcal{E}$ and pairs $\notin \mathcal{E}$ (**a**) before and (**b**) after the optimization. While some defects, such as fake or missing bonds, frequently appear in the pre-optimisation embedding, the optimised positions are constrained such that a clear cut is visible between the two distributions, easing the approximation.

In principle, we can program a different SLM pattern for the layout of each graph from the dataset. In practice however the SLM calibration step can be quite time-consuming, *i.e.* of the order of the minute. We can compare it to the duration of hundreds of shot, each of which consisting in applying a sequence and measuring a quantum state, performed at a frequency of 1 Hz. Then for each graph, calibrating the SLM and obtaining the probability distribution take approximately the same order of time.

We therefore seek to regroup many graphs onto the same SLM pattern, to be able to reduce the number of calibrations needed for the whole dataset. We do so by clustering the graphs according to similarities in their structures. Because the dataset consists in representations of organic molecules, many of the graphs share common structures. We thus focus on retrieving the presence and multiplicity of pentagons and hexagons. We then build a similarity measure between the graphs. For the pentagons for example, the similarity can be written under the form:

$$s(\mathcal{G}_1, \mathcal{G}_2) = 1 - \exp\bigl(-\alpha |N_1^P - N_2^P|\bigr) \tag{13}$$

where N^P represents the number of pentagons in $\mathcal{G}$ and α is a hyper-parameter. We then use a linear combination of similarity measures in order to build a similarity matrix between all graphs of the dataset. We then apply a k-means clustering algorithm C. (2004) using the similarity matrix in order to separate the graphs into different batches. Furthermore, since the laser power is distributed over all the traps, we want to reduce the total number of traps, in order to maximize the intensity provided to each trap. This ensures that the traps are deep enough to obtain a satisfying filling efficiency ($\sim 55\%$) over the whole pattern. For each batch, we thus apply the following mapping algorithm

[H] Creating a triangular SLM pattern by batching M graphs [1] Graphs $\{\mathcal{G}_1, \ldots, \mathcal{G}_M\}$ in sorted sizes and optimized positions $\{x_1, \ldots, x_M\}$ Single SLM pattern that embeds M graphs with optimal positions on a triangular lattice. $traps = \{\}$ for i in range $1, \cdots, M$:
find $r_{\mathcal{G}_i} = \{r_1, \ldots, r_{|\mathcal{G}_i|}\}$ triangular grid points that best conserve the pairwise distances between points in x_i and maximizes overlap with existing $traps$.
$traps \leftarrow traps + r_{\mathcal{G}_i} \setminus traps$ if $|traps| < 2|\mathcal{G}_M|$, add additional random triangular grid points to guarantee the filling property for re-arrangement.

We successfully map the entire dataset of 286 graphs into only 6 SLM patterns. For example, we batch 66 graphs together onto the 71-trap SLM pattern presented in Fig. 11. On average, the 6 SLM patterns use 70 traps each to encode 48 graphs each.

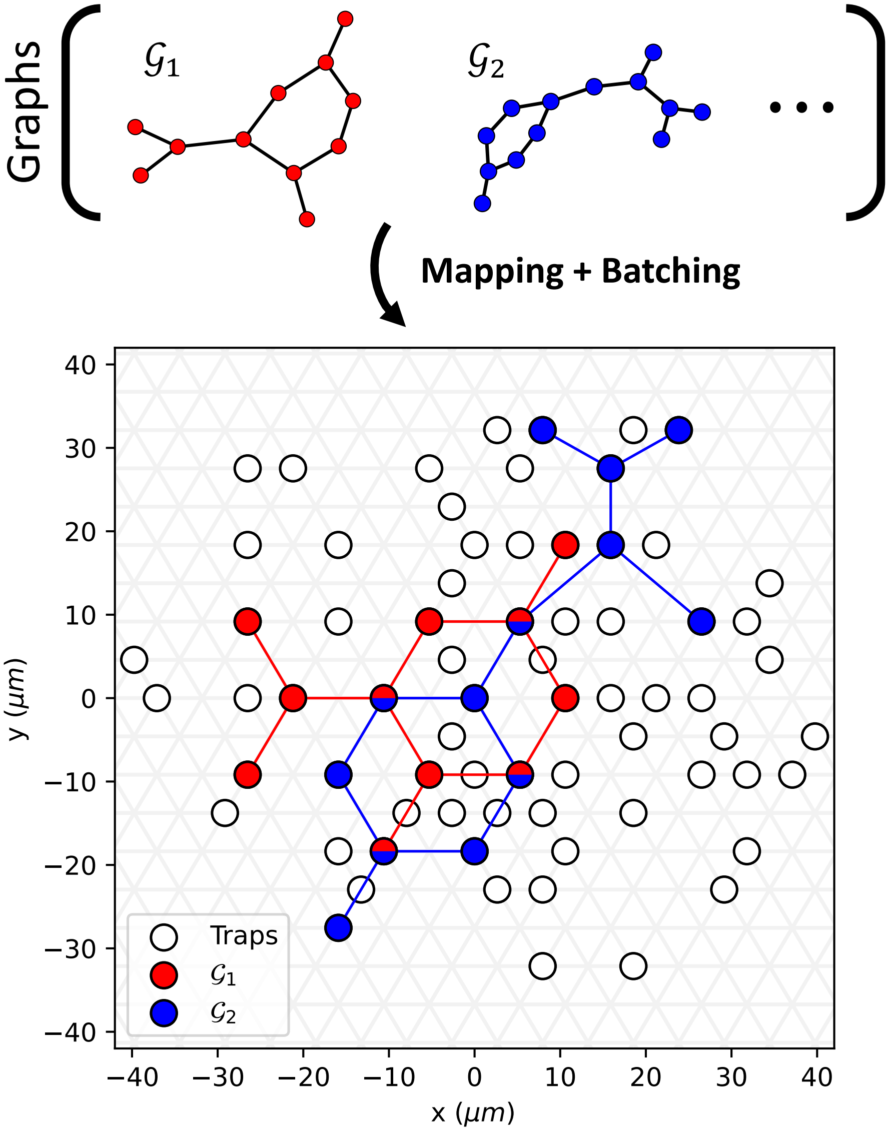

Figure 11: A family of 66 graphs, ranging in sizes from 4 to 19 nodes, is mapped and batched to the same SLM pattern (white dots) over a triangular grid with spacing $5.6\,\mu$m. The traps used when implementing $\mathcal{G}_1$ ($\mathcal{G}_2$) are colored in red (blue). The bi-colored traps are those used for both graphs.

H Classical graph kernels

A variety of classical kernels that do not require node or edge attributes are used in the main text to compare the performance of QEK on the PTC-FM dataset. In the following, a brief description of each is given.

H.1 SVM-ϑ kernel

The SVM-ϑ kernel was proposed as an alternative to the more computationally intensive Lovasz-ϑ kernel. Both ϑ kernels leverage the so-called orthogonal representation of a graph. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the orthogonal representation is an assignment of unit vectors $\{\mathbf{u}_i\}$ to each node of the graph, subject to the constraint that unit vectors associated to vertices that are not joined by an edge are orthogonal: $\langle \mathbf{u}_i, \mathbf{u}_j \rangle = 0$ if $\{i, j\} \notin \mathcal{E}$.

Orthogonal representations are not unique, but there is a particular representation associated with the ϑ number Lovasz (1979) of a graph. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with n vertices, denote $U_\mathcal{G}$ an orthogonal representation of $\mathcal{G}$, and C the space of unit vectors in $\mathbb{R}^n$. The ϑ number is defined as:

$$\vartheta(\mathcal{G}) := \min_{\mathbf{c}\in C} \min_{U_\mathcal{G}} \max_{\mathbf{u}_i \in U_\mathcal{G}} \frac{1}{\langle \mathbf{c}, \mathbf{u}_i \rangle^2}. \tag{14}$$

From now on, we will always be referring to the particular orthogonal representation $U_\mathcal{G}$ that minimizes (14).

Now consider a subset of vertices $B \subset \mathcal{V}$, and call $U_{\mathcal{G}|B}$ the orthogonal representation obtained from $U_\mathcal{G}$ by removing the vectors that are not in B:

$$U_{\mathcal{G}|B} := \{\mathbf{u}_i \in U_\mathcal{G} :\ i \in B\}. \tag{15}$$

Note that $U_{\mathcal{G}|B}$ preserves the global properties encoded in $U_\mathcal{G}$ through the orthogonal constraint, and that $U_{\mathcal{G}|B}$ is not in general the orthogonal representation of the subgraph of $\mathcal{G}$ containing only the vertices in B. Define the ϑ_B number:

$$\vartheta_B(\mathcal{G}) := \min_{\mathbf{c}\in C} \max_{\mathbf{u}_i \in U_{\mathcal{G}|B}} \frac{1}{\langle \mathbf{c}, \mathbf{u}_i \rangle^2}. \tag{16}$$

We are ready now to give the definition of the Lovasz-ϑ kernel. Given two graphs $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$, $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$, define:

$$K_{\text{Lo}}(\mathcal{G}_1, \mathcal{G}_2) := \sum_{B_1 \subset V_1} \sum_{B_2 \subset V_2} \delta_{|B_1|,|B_2|} \frac{1}{Z} k\left(\vartheta_{B_1}, \vartheta_{B_2}\right) \tag{17}$$

where $Z = \binom{|V_1|}{|B_1|}\binom{|V_2|}{|B_2|}$, δ is the Kronecker delta, and k is a freely specifiable kernel (called base kernel) from $\mathbb{R} \times \mathbb{R}$ to $\mathbb{R}$.

The SVM-ϑ kernel is defined as (17), but it uses an approximation for the ϑ numbers. Consider a graph $\mathcal{G}$ with n vertices and adjacency matrix A, and let $\rho \geq -\lambda$, where λ is the minimum eigenvalue of A. The matrix

$$\kappa := \frac{1}{\rho} A + I \tag{18}$$

is positive semi-definite. Define the maximization problem:

$$\max_{\alpha_i \geq 0} 2 \sum_{i=1}^{n} \alpha_i - \sum_{i,j=1}^{n} \alpha_i \alpha_j \kappa_{ij}. \tag{19}$$

If $\{\alpha_i^*\}$ are the maximizers of (19), then it can be proven that on certain families of graphs the quantity $\sum_i \alpha_i^*$ is with high probability a constant factor approximation to $\vartheta(\mathcal{G})$:

$$\vartheta(\mathcal{G}) \leq \sum_{i=1}^{n} \alpha_i^* \leq \gamma \vartheta(\mathcal{G}) \tag{20}$$

for some γ. The SVM-ϑ kernel then replaces the ϑ_B numbers on subgraphs with:

$$\vartheta_B(\mathcal{G}) \to \sum_{j \in B} \alpha_j^*. \tag{21}$$

The SVM-ϑ kernel requires a choice of base kernel $k : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. We choose a translation invariant universal kernel Micchelli et al. (2006) $k(x, y) = (\beta + ||x - y||^2)^{-\alpha}$, where α and β are two trainable hyperparameters.

H.2 Size kernel

Given two graphs $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$, the Size kernel is defined as:

$$K_{\text{size}}(\mathcal{G}_1, \mathcal{G}_2) := e^{-\gamma(|\mathcal{V}_1| - |\mathcal{V}_2|)^2} \tag{22}$$

with a choice of hyperparameter $\gamma > 0$.

H.3 Graphlet Sampling kernel

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and $H = (\mathcal{V}_H, \mathcal{E}_H)$ be two graphs. We say that $\mathcal{H}$ is a subgraph of $\mathcal{G}$ if there exists an injective map $\alpha : \mathcal{V}_{\mathcal{H}} \to \mathcal{V}$ such that $(u, v) \in \mathcal{E}_H \iff (\alpha(u), \alpha(v)) \in \mathcal{E}$. In general it might be possible to map $\mathcal{H}$ into $\mathcal{G}$ in several different ways, *i.e.* the mapping α, if it exists, is not necessarily unique.

Given two graphs $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$, the idea behind the Graphlet kernel is to pick an integer $k < \min\{|\mathcal{V}_1|, |\mathcal{V}_2|\}$, enumerate all possible graphs of size k and find the number of ways they can be mapped to $\mathcal{G}_1$ and $\mathcal{G}_2$. Denote by $f^{(k)}_{\mathcal{G}_i}$ the vector where each entry counts the way a specific graph of size k can be mapped as a subgraph of $\mathcal{G}_i$. A kernel can then be defined as the dot product $f^{(k)}_{\mathcal{G}_1} \cdot f^{(k)}_{\mathcal{G}_2}$ between the two vectors.

The complexity of computing such a kernel scales as $O(n^k)$, as there are $\binom{n}{k}$ size-k subgraphs in a graph of size n. For this reason it is preferable to resort to sampling rather than complete enumeration Shervashidze et al. (2009). Given a choice of integer N, graphs $g_1, \ldots, g_N$ of size between 3 and k are randomly sampled. The number of ways each g_i can be mapped as a subgraph of $\mathcal{G}_j$ is computed and stored in a vector $f_{\mathcal{G}_j}$, and the Graphlet Sampling kernel is defined as the dot product:

$$K_{\text{GS}}(\mathcal{G}_1, \mathcal{G}_2) := f_{\mathcal{G}_1} \cdot f_{\mathcal{G}_2} \tag{23}$$

To account for the different size of $\mathcal{G}_1$ and $\mathcal{G}_2$, each vector can be normalized by the total number of its subgraphs.

H.4 Random Walk kernel

The Random Walk kernel is one of the oldest and most studied graph kernels Gärtner et al. (2003). Given two graphs $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$, the idea is to measure the probability of simultaneous random walks of a certain length between two vertices in $\mathcal{G}_1$ and $\mathcal{G}_2$.

Simultaneous random walks can be conveniently encoded in powers of the adjacency matrix on the product graph. The product graph $\mathcal{G}_1 \times \mathcal{G}_2 = \mathcal{G}_\times = (\mathcal{V}_\times, \mathcal{E}_\times)$ is defined as follows:

$$\mathcal{V}_\times := \{(u_i, u_r) \mid u_i \in \mathcal{V}_1,\ u_r \in \mathcal{V}_2\} \tag{24}$$

$$\begin{aligned}\mathcal{E}_\times := \{\big((u_i, u_r), (v_j, v_s)\big) \mid (u_i, v_j) \in \mathcal{E}_1,& \\ (u_r, v_s) \in \mathcal{E}_2\}.&\end{aligned} \tag{25}$$

In other words, an edge in the product graph indicates that an edge exists between the endpoints in both $\mathcal{G}_1$ and $\mathcal{G}_2$. If $A_\times$ is the adjacency matrix of the product graph, then the entries of $A^k_\times$ indicate the probability of a simultaneous random walk of length k between two vertices $u_i, v_j \in \mathcal{V}_1$ and $u_r, v_s \in \mathcal{V}_2$.

If $p, q \in \mathbb{R}^{|\mathcal{V}_\times|}$ are vectors representing the probability distribution of respectively starting or stopping the walk at a certain node of $\mathcal{V}_\times$, the first idea for a kernel would be to compute the sum $\sum_k q^{\mathrm{T}} A^k_\times p$, which however may fail to converge. A simple modification to make the sum convergent is to choose an appropriate length-dependent weight $\mu(k)$:

$$K(\mathcal{G}_1, \mathcal{G}_2) := \sum_{k=0}^{\infty} \mu(k)\ q^{\mathrm{T}} A^k_\times p. \tag{26}$$

The Geometric Random Walk kernel is obtained by choosing the weights to be the coefficients of a geometric series $\mu(k) = \lambda^k$, and p, q to be uniform. If λ is tuned in such a way as to make the series convergent, the kernel reads:

$$K_{\text{RW}}(\mathcal{G}_1, \mathcal{G}_2) := \sum_{k=0}^{\infty} \lambda^k\ e^{\mathrm{T}} A^k_\times e = e^{\mathrm{T}} \left(I - \lambda A_\times\right)^{-1} e \tag{27}$$

where e denote vectors with all the entries equal to 1.

The cost of matrix inversion scales as the cube of the matrix size. If $|\mathcal{V}_1| = |\mathcal{V}_2| = n$, then the cost of the algorithm scales as $O(n^6)$, as it involves the inversion of an adjacency matrix of size $n^2 \times n^2$.

Several methods are proposed in Vishwanathan et al. (2010) to make the computation faster. The Spectral Decomposition method in particular allows to reduce the complexity for unlabeled graphs to $O(n^3)$. Essentially, one exploits the fact that the adjacency matrix of the product graph can be decomposed in the tensor product of the individual adjacency matrices:

$$A_\times = A_1 \otimes A_2 \tag{28}$$

which allows to diagonalize each $n \times n$ adjacency matrix in $O(n^3)$ time and perform the inversion only on the diagonal components.

H.5 Shortest Path kernel

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, an edge path between two vertices $u, v \in \mathcal{V}$ is a sequence of edges $(e_1, \dots, e_n)$ such that $u \in e_1$, $v \in e_n$, e_i and e_{i+1} are contiguous (*i.e.* they have one of the endpoints in common) and $e_i \neq e_j$ for $i \neq j$. Computing the shortest edge path between any two nodes of a graph can be done in polynomial time with the Dijkstra Dijkstra (1959) or Floyd-Warshall Floyd (1962) algorithms, which makes it a viable feature to be probed by a graph kernel.

The first step of the Shortest Path kernel is to transform the graphs into shortest path graphs. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the shortest path graph $\mathcal{G}^S = (\mathcal{V}^S, \mathcal{E}^S)$ associated to $\mathcal{G}$ is defined as:

$$\begin{aligned} \mathcal{V}^S &= \mathcal{V} \qquad (29) \\ \mathcal{E}^S &= \{(u, v) \mid \exists \text{ an edge path } (e_1, \dots, e_n) \\ &\quad \text{between } u \text{ and } v \text{ in } \mathcal{G}\} \qquad (30) \end{aligned}$$

In addition, to each edge $e \in \mathcal{E}^S$ a label $l(e)$ is assigned given by the length of the shortest path in $\mathcal{G}$ between its endpoints. The Shortest Path kernel is then defined as:

$$K_{\text{SP}}(\mathcal{G}_1, \mathcal{G}_2) := \sum_{e \in \mathcal{E}_1^S} \sum_{p \in \mathcal{E}_2^S} k(e, p) \tag{31}$$

with k being a kernel between edge paths such as the Brownian bridge kernel:

$$k(e, p) := \max\{0,\ c - |l(e) - l(p)|\} \tag{32}$$

for a choice of c.

I Geometric difference and maximum quantum-classical separation

Given two kernel functions K_1 and K_2, the geometric difference $g(K_1||K_2) = g_{12}$ described in Huang et al. (2021) is an asymmetric distance function that quantifies whether or not the kernel K_2 has the potential to resolve data better than K_1 on some dataset. In its simplest form, the geometric difference is defined as:

$$g_{12} = \sqrt{||\sqrt{K_2}\,(K_1)^{-1}\,\sqrt{K_2}||_\infty} \tag{33}$$

where $||\cdot||_\infty$ denotes the spectral norm.

The geometric difference becomes an especially useful metric when $K_1 = K_C$ is a classical kernel and $K_2 = K_Q$ is a quantum kernel. If N is the size of the dataset, a value of g_{CQ} of order $\sqrt{N}$ or greater indicates that the geometry of the feature space induced by the quantum kernel is rich enough to be hard to learn classically, and the quantum kernel can potentially perform better than classical kernels. In that case, it is possible to artificially relabel the dataset in order to maximally separate the kernels' performance. Such a relabeling process is a constructive proof of the existence of a certain dataset on which one kernel performs much better than the other. If v is the eigenvector of $\sqrt{K_2}\,(K_1)^{-1}\,\sqrt{K_2}$ corresponding to the eigenvalue g_{12}^2, the vector of new labels is given by $y_{\text{new}} = \sqrt{K_2}v$.

When dealing with a finite amount of training data, equation (33) should be regularized in order to stabilize the inversion of K_1. The regularized expression reads:

$$g_{12}(\lambda) = \sqrt{||\sqrt{K_2}\sqrt{K_1}\,(K_1 + \lambda I)^{-2}\,\sqrt{K_1}\sqrt{K_2}||_\infty} \tag{34}$$

where λ is the regularization parameter. The geometric difference $g_{12}(\lambda)$ has a plateau for small λ, when the regularization parameter becomes smaller than the smallest eigenvalue of K_1, and decreases for increasing λ. The effect of λ is to introduce a certain amount of training error. The training error can be upper bounded by a quantity proportional to:

$$g_{\text{tra}}(\lambda)^2 = \lambda^2 || \sqrt{K_2} \left(K_1 + \lambda I\right)^{-2} \sqrt{K_2} ||_\infty. \tag{35}$$

Practically, one should look at the regime where g_{12} has not plateaued but the training error is still small enough.

A regularization should be introduced also in the relabeling procedure. The new labels are taken to be $y_{\text{new}} = \sqrt{K_Q} v$, where v is the eigenvector of the regularized matrix

$$\sqrt{K_Q}\sqrt{K_C} \left(K_C + \lambda I\right)^{-2} \sqrt{K_C}\sqrt{K_Q}$$

corresponding to the eigenvalue $g_{12}(\lambda)^2$.