

# Fermionic Machine Learning

J. Gince<sup>1</sup>, J.-M. Pagé<sup>2</sup>, M. Armenta<sup>3</sup>, A. Sarkar<sup>1</sup> and S. Kourtis<sup>1</sup>

<sup>1</sup> *Département de physique & Institut quantique,*

*Université de Sherbrooke, Sherbrooke, QC, J1K 2R1, Canada*

<sup>2</sup> *Département de mathématique, Université de Sherbrooke, Sherbrooke, QC, J1K 2R1, Canada*

<sup>3</sup> *AlgoLab, Institut quantique, Université de Sherbrooke, Sherbrooke, QC, J1K 2R1, Canada*

(Dated: May 1, 2024)

We introduce *fermionic machine learning* (FermiML), a machine learning framework based on fermionic quantum computation. FermiML models are expressed in terms of parameterized matchgate circuits, a restricted class of quantum circuits that map exactly to systems of free Majorana fermions. The FermiML framework allows for building “fermionic” counterparts of any quantum machine learning (QML) model based on parameterized quantum circuits, including models that produce highly entangled quantum states. Importantly, matchgate circuits are efficiently simulable classically, thus rendering FermiML a flexible framework for utility benchmarks of QML methods on large real-world datasets. We initiate the exploration of FermiML by benchmarking it against unrestricted PQCs in the context of classification with random quantum kernels. Through experiments on standard datasets (Digits and Wisconsin Breast Cancer), we demonstrate that FermiML kernels are on-par with unrestricted PQC kernels in classification tasks using support-vector machines. Furthermore, we find that FermiML kernels outperform their unrestricted candidates on multi-class classification, including on datasets with several tens of relevant features. We thus show how FermiML enables us to explore regimes previously inaccessible to QML methods.

Quantum machine learning (QML) is an interdisciplinary area of research that explores the potential uses of quantum computing in machine learning (ML) [1–6]. QML aspires to utilize quantum phenomena (superposition, entanglement, interference) to process data fundamentally differently from classical computers. Although QML concepts have been theorized since the early days of quantum computing [7–12], recent advances in quantum computing hardware have sharply increased the interest for development of practical QML applications in a short- to mid-term horizon [2–4, 13].

A hotly debated question has been whether there exist QML approaches that operate on noisy quantum processors lacking fault tolerance and that can outperform classical algorithms in specific ML tasks, yielding what is often termed as quantum advantage. While claims that the answer to this question may be affirmative have surfaced [14–17, 19–26], the highly tailored benchmarks involved in these proposals cast doubts to the relevance of existing QML techniques in practical settings, with some even questioning the pursuit of quantum advantage as it is currently construed [27, 28].

Near-term QML methods are predominantly based on variational quantum algorithms (VQAs) [29]. In VQAs, parameterized quantum circuits are tuned via standard classical optimization performed by a classical computer and using probability outcomes of specific quantum circuits to evaluate a task-dependent loss function that is being optimized. Near-term QML methods based on PQCs face two major challenges, namely, barren plateaus and lack of scalability [30].

Barren plateaus are parameter regimes in which gradient-based optimization becomes inefficient due to vanishingly small gradients. It has been observed that

barren plateaus are widespread whenever QML is applied beyond the toy-problem scale, thus plaguing most QML approaches [30–34]. While barren plateaus pose an obstacle to the deployment of QML, it is yet unclear whether they are a generic feature of QML based on PQCs rather than a side effect of model choice, with recent work pointing to models in which barren plateaus are provably absent [35, 36].

On the other hand, demonstrating the performance of QML protocols upon scaling towards realistic data remains intractable. As far as real-world datasets are concerned, both experiments on near-term QPUs and classical simulation of existing QML models are likely to remain intractable in the foreseeable future. Due to this, the capacity to determine the combinations of quantum resources that can meaningfully boost ML performance in practical settings is severely limited. This lack of scalability remains an obstacle in benchmarking QML even in the absence of barren plateaus.

With this challenge in mind, we introduce fermionic machine learning (FermiML), a machine learning framework based on fermionic quantum computation. The FermiML framework allows for building key ingredients of quantum mechanics - superposition, interference, and arbitrarily high entanglement - into QML models that are scalable. FermiML models are expressed in terms of parameterized matchgate circuits, a restricted class of quantum circuits that map exactly to systems of free fermions in one dimension and are hence efficiently simulable classically [51]. FermiML thus offers a toolbox for benchmarking QML protocols on realistic data sets by restricting the corresponding quantum circuits inside the matchgate manifold. On one hand, FermiML establishes a yardstick of performance for any QML algorithm,

since any algorithm would need to outperform FermiML to be considered useful. On the other hand, FermiML can be incorporated into classical (deep) learning models, leading to novel quantum-inspired classical algorithms for ML, in the vein of a growing body of classical algorithms that emulate quantum principles to outperform the previous state-of-the-art [37–42].

In this work, we begin the exploration of FermiML in the context of quantum kernel learning (QKL) [43]. QKL is a straightforward extension of kernel methods in the quantum realm. In QKL, classical data points  $x$  are mapped to density matrices through a map  $\rho : x \mapsto \rho(x) \in \mathcal{H} = \mathbb{C}^{2^N \times 2^N}$  [44]. The mapping is achieved through evolving an easy-to-prepare quantum state over  $N$  qubits, typically  $|\mathbf{0}\rangle = |0\rangle^{\otimes N}$ , with a parameterized quantum circuit  $U(x)$ , so that  $\rho(x) = |\psi(x)\rangle\langle\psi(x)|$  with  $|\psi(x)\rangle = U(x)|\mathbf{0}\rangle$ . The dimension of the feature space is thus determined by the number of qubits. The encoding of  $x$  into  $\rho(x)$  is determined by the geometry and parameterized gates of the quantum circuit, as we discuss below. The quantum kernel is then defined as  $\mathcal{U}(x, x') = \text{Tr}[\rho(x)\rho(x')]$  and measures the similarity between the quantum states associated with  $x$  and  $x'$  [44]. This kernel is evaluated to a given accuracy by repeated measurements of  $|\psi(x)\rangle$ .

Once a kernel for a given dataset is obtained, kernel methods can be employed to perform standard ML tasks, such as classification or regression [45]. The parameters of a quantum kernel can either be judiciously predetermined or variationally optimized to better align the kernel with the data, but the latter strategy potentially gives rise to barren plateaus. Since our goal in this work is to tackle the scalability of QML to real datasets, below we study exclusively predetermined kernels for which the barren plateau issue is irrelevant.

FermiML is based on fermionic quantum computation (FermiQC) [46], a restricted class of gate-based quantum computations grounded in the physics of free fermion systems [47–54]. FermiQC contains the set of computations performed by so-called holographic algorithms [55–58], originally geared towards the solution of counting problems [59–63]. FermiQC is implemented by circuits of matchgates, which are algebraically restricted parity-preserving nearest-neighbour quantum gates. The form of a 2-qubit matchgate is

$$U(A, W) = \begin{bmatrix} a & 0 & 0 & b \\ 0 & w & x & 0 \\ 0 & y & z & 0 \\ c & 0 & 0 & d \end{bmatrix}, \quad (1)$$

where  $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$  and  $W = \begin{pmatrix} w & x \\ y & z \end{pmatrix}$  satisfy the constraint  $\det A = \det W$ . The matrices  $A$  and  $W$  represent the two sub-blocks in even and odd parity subspaces of a 2-qubit Hilbert space in the ordered basis  $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ . Circuits built of such gates acting exclusively on pairs of

neighboring qubits arranged in a linear array are classically simulable in polynomial time [51].

The connection with the physics of free fermions is established by expressing each matchgate  $U = e^{iH}$ , where  $H$  may be expanded as a linear combination of quadratic Majorana monomials [48, 51],

$$H = i \sum_{\mu \neq \nu=1}^{2N} h_{\mu\nu} c_\mu c_\nu, \quad (2)$$

where the range of the summation has been extended over all  $N$  qubits. The coefficients  $h_{\mu\nu}$  are elements of a real, antisymmetric,  $2N \times 2N$  matrix. The  $c_\mu$  are Majorana spinors whose components obey anti-commutation relations:  $\{c_\mu, c_\nu\} = c_\mu c_\nu + c_\nu c_\mu = 2\delta_{\mu\nu}I$ , for  $\mu, \nu = 1, \dots, 2N$ . The linear span of the Majorana spinors is preserved under conjugation by any matchgate unitary:

$$U c_\mu U^\dagger = \sum_{\nu=1}^{2N} R_{\mu\nu} c_\nu, \quad (3)$$

where  $R$  is a  $2N \times 2N$  matrix determined by reversing the above equation, namely,

$$R_{\mu\nu} = \frac{1}{4} \text{Tr}\{(U c_\mu U^\dagger) c_\nu\}. \quad (4)$$

Using  $R$ , the fermionic kernel can be expressed as,

$$\mathcal{U}(x, x') = \sum_{\mu_1, \nu_1, \dots, \mu_N, \nu_N}^{2N} T_{j_1, \mu_1} T_{j_1, \nu_1}^* \dots T_{j_N, \mu_N}^* T_{j_N, \nu_N} \times \langle \mathbf{0} | c_{\mu_1} c_{\nu_1} \dots c_{\nu_N} c_{\mu_N} | \mathbf{0} \rangle, \quad (5)$$

where for the  $j^{\text{th}}$  qubit line, we define

$$T_{j, \nu} = \frac{1}{2} (R_{2j-1, \nu}^\top + i R_{2j, \nu}^\top). \quad (6)$$

The summation in Eq. (5) can be performed efficiently (see Appendix).

As depicted in Figure 1, our kernels are constructed through a series of alternating layers of parameterized  $U(R_y(\theta_1), R_y(\theta_2))$  and  $U(R_z(\theta_1), R_z(\theta_2))$  gates and non-parameterized  $U(H, H)$  and  $U(Z, X)$  gates, where  $U$  comes from Eq. (1). Classical data is loaded into the kernel circuit via the encoding

$$\theta_j = c_\theta \theta_r + c_x x_j, \quad (7)$$

where  $\theta_r \in [0, 1]$  is a randomly chosen parameter,  $x_j$  corresponds to the  $j^{\text{th}}$  feature of the data point  $x$ , and  $c_\theta$  and  $c_x$  serve as scaling factors, both set to  $\pi/2$  [66]. The depth of the kernel circuit is

$$d = \left\lceil \frac{\chi}{N} \right\rceil, \quad (8)$$

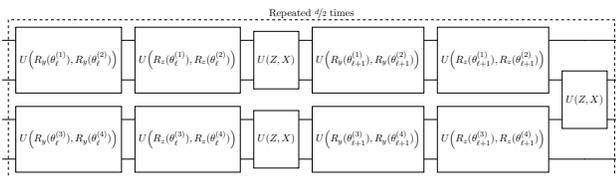


FIG. 1. The fPQC has  $N$  qubits and  $d/2$  layers of parameterized  $U(R_y(\theta_1), R_y(\theta_2))$  and  $U(R_z(\theta_1), R_z(\theta_2))$  gates followed by entangling gates such as  $U(Z, X)$ .

where  $\chi$  is the number of features in the dataset. We compare FermiML kernels with  $U(H, H)$  (denoted as hfPQC) and  $U(Z, X)$  (denoted as fPQC) entangling layers against unrestricted kernels (PQC) with respect to classification accuracy. We also include unentangled fermionic ( $\otimes$ fPQC) kernels as a point of reference. Fermionic circuits are simulated efficiently using our own MatchCake library [69] whereas for generic quantum kernels we use the PennyLane statevector simulator [70]. We use standard techniques to train a support vector machine (SVM) for each dataset kernel [45].

We use two standard classical datasets as benchmarks for classification with FermiML: Wisconsin Breast Cancer (WBC) dataset [67] (569 samples, 30 features, 2 classes), and Digits dataset [68] (1797 samples, 64 features, 10 classes). For all datasets and kernel types, we perform 5-fold cross-validation. Performance comparisons on training and test data for various models corresponding to both WBC and Digits datasets are given in Fig. 3 of the Appendix.

For WBC, the plot in Fig. 2(a) demonstrates test accuracies for various models against the number of qubits  $N$  and indicates accuracies close to 95%. The unentangled matchgate kernel ( $\otimes$ fPQC) shows consistently lower accuracies compared to entangling kernels. While the matchgate models exhibit slightly lower accuracies compared to the PQC with traditional quantum unitaries, simulating PQCs beyond  $N = 16$  qubits becomes highly resource-intensive due to the exponential scaling of the statevector. Conversely, simulation runtime scales polynomially for all FermiML models, making it feasible to easily scale to  $N = 30$ , matching this dataset’s feature count and promising an edge of FermiML for QML tasks that require large numbers of qubits. In Figure 2(b), the test accuracies of different models on the Digits dataset are presented. Here, both fPQC and hfPQC exhibit slightly better accuracy compared to the PQC. When considering models with fewer qubits, up to about  $N = 26$ , the performance of  $\otimes$ fPQC is consistently inferior, indicating that entanglement plays an important role in this regime. As the number of qubits increases, the accuracies of fPQC, hfPQC, and the unentangled  $\otimes$ fPQC all converge, meeting within error bars at around  $N = 30$ . Additionally, here too, we are able to

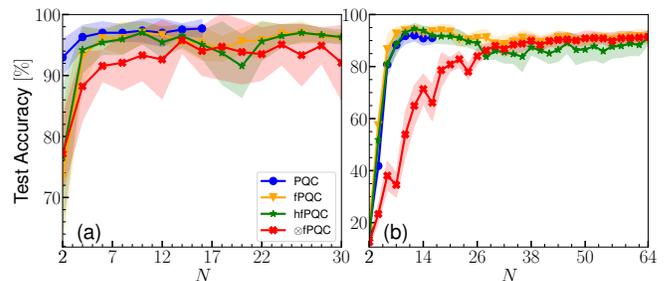


FIG. 2. Test accuracies of various models for WBC (a) and Digits (b) datasets as a function of the number of qubits  $N$ . Shaded regions represent the standard deviation of the accuracy. Note the different axis ranges in the two panels.

scale up the number of qubits in FermiML models beyond what is feasible with PQC to match the feature count of the dataset.

FermiML is a QML framework rooted in the physics of free fermions. In this work, we have demonstrated its usefulness as a benchmark of QML approaches on realistic datasets, achieving high levels of accuracy in both binary (WBC) and multi-class (Digits) classification. The latter has seldom been addressed in QML benchmarking [71]. In contrast to all other existing QML implementations which are limited by either the capabilities of quantum hardware or the intractability of classical simulation, FermiML is scalable, operating within polynomial runtimes on classical computers.

Our findings prompt further exploration of FermiML models. QML is often advertised as a paradigm that employs quantum resources (entanglement, superposition, interference) to accelerate AI tasks. If FermiML can achieve performance on par with proposed QML methods, then the utility of the latter is questionable. FermiML hence offers a practical tool to benchmark putative QML breakthroughs in realistic settings, thus operationalizing recent skepticism with respect to the applicability of QML as currently construed. [35, 71]. Along this line, it is tempting to benchmark FermiML surrogates of more intricate QML models on large datasets.

Another line of investigation is the use of FermiML for classical “warm-starting” of QML protocols deployed on actual quantum hardware. This is particularly intriguing given that barren plateaus have so far posed a troublesome obstacle to QML [30–34]. Finally, FermiML allows for efficiently incorporating quantum mechanical effects in classical deep learning models, along the lines of recent work on quantum-inspired algorithms [37–42].

We acknowledge useful discussions with A. Chapman and critical reading by S. M. Fadaie. This work was supported by the Ministère de l’Économie, de l’Innovation et de l’Énergie du Québec through its Research Chair in Quantum Computing, an NSERC Discovery grant. JG acknowledges support from the QSciTech CREATE-

NSERC program. AS acknowledges support from the Canada First Research Excellence Fund through an Institut quantique postdoctoral fellowship. This work also made use of computational resources by Calcul Québec and the Digital Research Alliance of Canada.

- 
- [1] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum algorithms for supervised and unsupervised machine learning, arXiv:1307.0411 (2013).
- [2] P. Wittek, Quantum machine learning: what quantum computing means to data mining, Academic Press, (2014).
- [3] M. Schuld, I. Sinayskiy, and F. Petruccione, An introduction to quantum machine learning, Contemporary Physics **56**, 172 (2015).
- [4] M. Schuld and F. Petruccione, Supervised learning with quantum computers (eBook), Springer, (2018).
- [5] C. Ciliberto et.al., Quantum machine learning: a classical perspective, Proc. R. Soc. A **474**, (2018).
- [6] A. Zeguendry, Z. Jarir, and M. Quafafou, Quantum machine learning: A review and case studies, Entropy **25(2)**, 287, (2023).
- [7] R. A. Servedio and S. J. Gortler, Equivalences and separations between quantum and classical learnability, SIAM J. Comput. **33**, 1067 (2004).
- [8] G. Bonnell and G. Papini, Quantum neural network, Int. J. Theor. Phys **36**, 2855 (1997).
- [9] D. Ventura and T. Martinez, Quantum associative memory, Inf. Sci **124**, 273 (2000).
- [10] R. Schützhold, Pattern recognition on a quantum computer, Phys. Rev. A **67**, 062311 (2003).
- [11] S. C. Kak, Quantum neural computing, pp. 259–313, Advances in Imaging and Electron Physics, Elsevier (1995).
- [12] N. H. Bshouty and J. C. Jackson, Learning DNF over the uniform distribution using a quantum example oracle, SIAM J. Comput. **28**, 1136 (1998).
- [13] V. Dunjko and P. Wittek, A non-review of quantum machine learning: trends and explorations, Quantum Views **4** (2020).
- [14] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, Phys. Rev. Lett. **103**, 150502 (2009).
- [15] N. Wiebe, D. Braun, and S. Lloyd, Quantum algorithm for data fitting, Phys. Rev. Lett. **109**, 050505 (2012).
- [16] A. Zlokapa, H. Neven, and S. Lloyd, A quantum algorithm for training wide and deep classical neural networks, arXiv:2107.09200 (2021).
- [17] P. Rebentrost, M. Mohseni, and S. Lloyd, Quantum support vector machine for big data classification, Phys. Rev. Lett. **113**, 130503 (2014).
- [18] J. Adcock et. al., Advances in quantum machine learning, arXiv:1512.02900v1 (2015).
- [19] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, Circuit-centric quantum classifiers, Phys. Rev. A **101**, 032308 (2020).
- [20] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, Parameterized quantum circuits as machine learning models, Quantum Sci. Technol. **4**, 043001 (2019).
- [21] N. Wiebe, A. Kapoor, and K. M. Svore, Quantum deep learning, arXiv:1412.3489 (2015).
- [22] I. Kerenidis and A. Prakash, Quantum machine learning with subspace states, arXiv:2202.00054 (2022).
- [23] K. Wold, Parameterized quantum circuits for machine learning, Master thesis, University of Oslo (2021).
- [24] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, The power of quantum neural networks, Nat. Comput. Sci **1**, 403 (2021).
- [25] A. P. Lund, M. J. Bremner, and T. C. Ralph, Quantum sampling problems, Boson Sampling and quantum supremacy, npj Quantum Inf **3**, 15 (2017).
- [26] A. Morvan and et.al., Phase transition in random circuit sampling, arXiv:2304.11119 (2023).
- [27] Y. Liu, S. Arunachalam, and K. Temme, A rigorous and robust quantum speed-up in supervised machine learning, Nat. Phys. **17**, 1013 (2021).
- [28] M. Schuld and N. Killoran, Is quantum advantage the right goal for quantum machine learning?, PRX Quantum **3**, 030101 (2022).
- [29] M. Cerezo et. al., Variational quantum algorithms, Nat. Rev. Phys. **625** (2021).
- [30] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush and H. Neven. Barren plateaus in quantum neural network training landscapes, Nat Commun **9**, 4812 (2018).
- [31] E. Grant, L. Wossnig, M. Ostaszewski, and M. Benedetti, An initialization strategy for addressing barren plateaus in parametrized quantum circuits, Quantum **3**, 214 (2019).
- [32] A. Arrasmith, M. Cerezo, P. Czarnik, L. Cincio, and P. J. Coles, Effect of barren plateaus on gradient-free optimization, Quantum **5**, 558 (2021).
- [33] M. Cerezo, A. Sone, T. Volkoff, L. Cincio and P. J. Coles, Cost function dependent barren plateaus in shallow parametrized quantum circuits. Nat Commun **12**, 1791 (2021).
- [34] H. Qi, L. Wang, H. Zhu, A. Gani and C. Gong, The barren plateaus of quantum neural networks: review, taxonomy and trends. Quantum Inf Process **22**, 435 (2023).
- [35] M. Cerezo et. al., Does provable absence of barren plateaus imply classical simulability? or, why we need to rethink variational quantum computing, arXiv:2312.09121 (2023).
- [36] N. L. Diaz and D. García-Martín and S. Kazi, M. Larocca and M. Cerezo, Showcasing a barren plateau theory beyond the dynamical lie algebra, arXiv:2310.11505 (2023).
- [37] A. Gilyén, S. Lloyd, and E. Tang, Quantum-inspired low-rank stochastic regression with logarithmic dependence on the dimension, arXiv:1811.04909 (2018).
- [38] E. Tang, A quantum-inspired classical algorithm for recommendation systems, in Proceedings of the 51st annual ACM SIGACT symposium on theory of computing pp. 217–228 (2019).
- [39] A. Gilyén, Z. Song, and E. Tang, An improved quantum-inspired algorithm for linear regression, Quantum **6**, 754 (2022).
- [40] J. M. Arrazola, A. Delgado, B. R. Bardhan, and S. Lloyd, Quantum-inspired algorithms in practice, Quantum **4**, 307 (2020).
- [41] Y. Tene-Cohen, T. Kelman, O. Lev, and A. Makmal, A variational qubit-efficient maxcut heuristic algorithm, arXiv:2308.10383 (2023).
- [42] A. Misra-Spieldenner, T. Bode, P. K. Schuhmacher, T. Stollenwerk, D. Bagrets, and F. K. Wilhelm, Mean-field approximate optimization algorithm, PRX Quantum **4**, 030335 (2023).

- [43] M. Schuld, Supervised quantum machine learning models are kernel methods, arXiv:2101.11020 (2021).
- [44] J. M. Kübler, S. Buchholz, and B. Schölkopf, The inductive bias of quantum kernels, arXiv:2106.03747 (2021).
- [45] J. Shawe-Taylor and N. Cristianin, Kernel Methods for Pattern Analysis, Cambridge University Press (2004).
- [46] S. B. Bravyi and A. Y. Kitaev, Fermionic quantum computation, Ann. Phys. (N. Y.) **298**, 210 (2002).
- [47] E. Knill, Fermionic linear optics and matchgates, arXiv:quant-ph/0108033 (2001).
- [48] B. M. Terhal and D. P. DiVincenzo, Classical simulation of noninteracting-fermion quantum circuits, Phys. Rev. A **65**, 032325 (2002).
- [49] S. Bravyi and D. Gosset, Complexity of quantum impurity problems, Commun. Math. Phys **356**, 451 (2017).
- [50] M. Hebenstreit, R. Jozsa, B. Kraus, and S. Strelchuk, Computational power of matchgates with supplementary resources, Phys. Rev. A **102**, 052604 (2020).
- [51] R. Jozsa and A. Miyake, Matchgates and classical simulation of quantum circuits, Proc. R. Soc. A. **464**, 3089 (2008).
- [52] R. Jozsa, Embedding classical into quantum computation, arXiv:0812.4511 (2008).
- [53] D. J. Brod and E. F. Galvão, Extending matchgates into universal quantum computation, Phys. Rev. A **84**, 022310 (2011).
- [54] D. J. Brod, Efficient classical simulation of matchgate circuits with generalized inputs and measurements, Phys. Rev. A **93**, 062332 (2016).
- [55] L. G. Valiant, Quantum computers that can be simulated classically in polynomial time, Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, STOC '01 (Association for Computing Machinery, New York, NY, USA) p. 114–123 (2001).
- [56] L. G. Valiant, Expressiveness of matchgates, Theoretical Computer Science **289**, 457 (2002).
- [57] L. G. Valiant, Holographic circuits, pp. 1–15., Automata, Languages and Programming, Springer Berlin, (2005).
- [58] L. G. Valiant, Holographic algorithms, SIAM J. Comput. **37**, 1565 (2008).
- [59] J.-Y. Cai and V. Choudhary, Valiant’s holant theorem and matchgate tensors, Theor. Comput. Sci. **384**, 22 (2007).
- [60] J.-Y. Cai and P. Lu, Holographic algorithms with unsymmetric signatures. In Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms (SODA '08). Society for Industrial and Applied Mathematics, USA, 54–63 (2008).
- [61] J.-Y. Cai and P. Lu, Holographic algorithms: The power of dimensionality resolved, Theor. Comput. Sci. **410**, 1618 (2009).
- [62] J.-Y. Cai and P. Lu, On Symmetric Signatures in Holographic Algorithms. In: Thomas, W., Weil, P. (eds) STACS 2007. STACS 2007. Lecture Notes in Computer Science, vol 4393. Springer, Berlin, Heidelberg (2007).
- [63] J.-Y. Cai, P. Lu, and M. Xia, Holant problems and counting CSP, Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing, p. 715–724 (2009).
- [64] J.-Y. Cai, V. Choudhary and P. Lu, On the Theory of Matchgate Computations, Theory of Computing Systems **46**, 398 (2010).
- [65] G. Matos, C. N. Self, Z. Papić, K. Meichanetzidis, and H. Dreyer, Characterization of variational quantum algorithms using free fermions, Quantum **7**, 966 (2023).
- [66] T. Haug, C. N. Self, and M. Kim, Quantum machine learning of large datasets using randomized measurements, Mach. Learn.: Sci. Technol. **4**, 015005 (2023).
- [67] W. Wolberg, O. Mangasarian, N. Street, and W. Street, Breast Cancer Wisconsin (Diagnostic), UCI Machine Learning Repository (1995).
- [68] E. Alpaydin and C. Kaynak, Optical Recognition of Handwritten Digits, UCI Machine Learning Repository (1998).
- [69] Code available at <https://github.com/MatchCake/MatchCake>
- [70] V. Bergholm et.al, PennyLane: Automatic differentiation of hybrid quantum-classical computations, arXiv:1811.04968 (2022).
- [71] J. Bowles, S. Ahmed, and M. Schuld, Better than classical? the subtle art of benchmarking quantum machine learning models, arXiv:2403.07059 (2024).

### Preliminaries for Matchgates and mapping to free fermions

Matchgates can be represented as a linear combination of the six operators in the set  $\mathcal{M}_G = \{X_k X_{k+1}, X_k Y_{k+1}, Y_k Y_{k+1}, Y_k X_{k+1}, Z_k I_{k+1}, I_k Z_{k+1}\}$ . Equivalently, any matchgate acting on nearest-neighbor qubits  $k$  and  $k + 1$  is a Gaussian operation  $U = e^{iH}$  generated from a Hamiltonian  $H$  represented as a sum of three interactions:

$$\begin{aligned} H &= H_1 + H_2 + H_3, \\ H_1 &= \alpha_1 Z_k \otimes I_{k+1} + \beta_1 I_k \otimes Z_{k+1}, \\ H_2 &= \alpha_2 X_k \otimes X_{k+1} + \beta_2 Y_k \otimes Y_{k+1}, \\ H_3 &= \alpha_3 X_k \otimes Y_{k+1} + \beta_3 Y_k \otimes X_{k+1}, \end{aligned} \quad (9)$$

where  $\alpha_j, \beta_j$  are real coefficients and  $X_k, Y_k$  and  $Z_k$  are Pauli matrices. The Pauli operators are mapped to fermionic creation and annihilation operators via the Jordan-Wigner transformation. Operators related to the  $j^{\text{th}}$  fermionic mode are expressed in terms of creation ( $a_j^\dagger$ ) and annihilation ( $a_j$ ) operators obeying canonical anti-commutation relations,

$$\{a_k, a_j\} = 0, \{a_k^\dagger, a_j^\dagger\} = 0, \{a_k, a_j^\dagger\} = \delta_{kj} I, \quad (10)$$

where  $j, k = 1, \dots, N$  and  $\delta$  denotes the Kronecker delta. With these relations, the Hamiltonian terms can be rewritten as [48, 65]

$$\begin{aligned} H_1 &= 2\alpha_1 a_k^\dagger a_k + 2\beta_1 a_{k+1}^\dagger a_{k+1}, \\ H_2 &= \alpha_2 (a_k^\dagger - a_k)(a_{k+1}^\dagger + a_{k+1}) - \beta_2 (a_k^\dagger + a_k)(a_{k+1}^\dagger - a_{k+1}), \\ H_3 &= -i\alpha_3 (a_k^\dagger - a_k)(a_{k+1}^\dagger - a_{k+1}) - i\beta_3 (a_k^\dagger + a_k)(a_{k+1}^\dagger + a_{k+1}) \end{aligned} \quad (11)$$

and  $H$  is then a sum of nearest-neighbor fermionic interactions, quadratic in creation and annihilation operators. Each fermion operator can be split into Majorana operators  $c_\mu$  as

$$\begin{aligned} a_k &= \frac{c_{2k-1} + ic_{2k}}{2}, a_k^\dagger = \frac{c_{2k-1} - ic_{2k}}{2} \\ c_{2k-1} &= a_k + a_k^\dagger, \quad c_{2k} = -i(a_k - a_k^\dagger), \end{aligned} \quad (12)$$

and the  $c_\mu$ 's anti-commute:  $\{c_\mu, c_\nu\} = c_\mu c_\nu + c_\nu c_\mu = 2\delta_{\mu\nu}I$ ,  $\mu, \nu = 1, \dots, 2N$ . The summation implies that the Majorana monomials are linearly independent, and the associated vector space  $\mathcal{C}_{2N}$  has dimension  $2^{2N} = 2^N \times 2^N$ . Consequently, matrix representations of the  $c_\mu$ 's, are of dimensions,  $2^N \times 2^N$ . Invoking the Jordan-Wigner transformation once more, the  $2N$ -dimensional Majorana operators can be represented in recognised forms of  $N$ -qubit Pauli operators [51] as

$$\begin{aligned} c_{2k-1} &= Z^{\otimes(k-1)} \otimes X \otimes I^{\otimes(N-k)}, \\ c_{2k} &= Z^{\otimes(k-1)} \otimes Y \otimes I^{\otimes(N-k)}. \end{aligned} \quad (13)$$

The six terms in the matchgate family are then

$$\begin{aligned} Z_k I_{k+1} &= -i c_{2k-1} c_{2k}, \\ X_k X_{k+1} &= i c_{2k} c_{2k+1}, \\ Y_k Y_{k+1} &= i c_{2k-1} c_{2k+2}, \\ Y_k X_{k+1} &= i c_{2k-1} c_{2k+1}, \\ X_k Y_{k+1} &= -i c_{2k} c_{2k+2}, \\ I_k Z_{k+1} &= -i c_{2k+1} c_{2k+2} \end{aligned}$$

for  $k = 1, \dots, N$ .

### Simulating output probabilities

Nearest-neighbor matchgate circuits are recognized for their polynomial-time simulability, attributed to their link with perfect matchings of graphs [49] and their mapping to free-fermionic Hamiltonians. Using this mapping it has been shown that the output probabilities  $|\langle y|U|x\rangle|^2$  for any bitstrings  $x$  and  $y$  can be evaluated efficiently in polynomial time [48, 65]. In the following discussion, we recapitulate this method and in the end, provide a detailed explanation tailored to our specific scenario. Our approach leverages matchgates to implement the PQCs, subsequently employed in the execution of the kernel method.

To simulate the output probabilities for any input state  $|x\rangle$  of length  $N$  and Hamming weight  $\ell$  it is first expanded into fermionic creation operators acting on the vacuum state (the number of creation operators are determined by the Hamming weight of the bitstring) and in turn are mapped to Majorana operators,

$$\begin{aligned} |x\rangle &= a_{p_1}^\dagger \dots a_{p_\ell}^\dagger |\mathbf{0}\rangle \\ &= c_{2p_1} \dots c_{2p_\ell} |\mathbf{0}\rangle, \text{ with } p_1 < p_2 \dots < p_\ell. \end{aligned} \quad (14)$$

The probability that a certain subset  $k$  of the total  $N$  qubits is in a particular state  $y^*$  (meaning, the output bitstring is of length  $k$ ) for a given input state  $|x\rangle$  is therefore given by,

$$\begin{aligned} p(y^*|x) &= \langle x|c_{2p_\ell} \dots c_{2p_1} (U^\dagger a_{j_1} U) (U^\dagger a_{j_1}^\dagger U) \times \dots \\ &\times (U^\dagger a_{j_k}^\dagger U) (U^\dagger a_{j_k} U) c_{2p_1} \dots c_{2p_\ell} |x\rangle. \end{aligned} \quad (15)$$

$k < j$	$j$		
	$c_{m_\beta}$	$c_{n_\beta}$	$c_{2p_\beta}$
$c_{m_\alpha}$	$(TBT^\top)_{j_\alpha, j_\beta}$	$(TBT^\top)_{j_\alpha, j_\beta}$	$(TB)_{j_\alpha, 2p_\beta}$
$c_{n_\alpha}$	$(T^*BT^\top)_{j_\alpha, j_\beta}$	$(T^*BT^\top)_{j_\alpha, j_\beta}$	$(T^*B)_{j_\alpha, 2p_\beta}$
$c_{2p_\alpha}$	$(BT^\top)_{2p_\alpha, j_\beta}$	$(BT^\dagger)_{2p_\beta, j_\beta}$	$\delta_{\alpha, \beta}$

TABLE I. This lookup table is used to find the matrix elements  $M_{kj}$  for  $k < j$ .  $B$  is given in Eq. (19).

Using Eqs. (3) and (6) and the conjugation relations for the fermionic operators

$$U^\dagger a_j U = \sum_{\nu=1}^{2N} T_{j,\nu} c_\nu, \quad (16)$$

$$U^\dagger a_j^\dagger U = \sum_{\nu=1}^{2N} T_{j,\nu}^* c_\nu, \quad (17)$$

the summation in Eq. 15 maybe rewritten as,

$$\begin{aligned} p(y^*|x) &= \prod_{\gamma=1}^k \sum_{m_\gamma, n_\gamma=1}^{2N} T_{j_\gamma, m_\gamma} T_{j_\gamma, n_\gamma}^* \\ &\times \left\langle \mathbf{0} \left| c_{2p_\ell} \dots c_{2p_1} \left( \prod_{\gamma=1}^k c_{m_\gamma} c_{n_\gamma} \right) c_{2p_1} \dots c_{2p_\ell} \right| \mathbf{0} \right\rangle. \end{aligned} \quad (18)$$

For finding the fully contracted terms in the above summation, the Majorana operators are at first expanded in terms of the fermionic creation and annihilation operators. Wick's theorem of normal ordering of fermions is then applied to each of the terms. All of these fully contracted terms are generated by contraction strategies given in Eqs.(43) and (44) in Ref. [48]. The same is summarised using a matrix  $B$ ,  $c_j c_j = B_{ij}$ , where  $B$  is a  $2N \times 2N$  Hermitian block diagonal matrix, containing  $N$  number of  $2 \times 2$  blocks

$$B = \begin{pmatrix} 1 & i & & & \\ -i & 1 & & & \\ & & \ddots & & \\ & & & 1 & i \\ & & & -i & 1 \end{pmatrix} = \bigoplus_{l=1}^N \begin{pmatrix} 1 & i \\ -i & 1 \end{pmatrix}. \quad (19)$$

Subsequently Table. I [48] is utilised to construct a  $2(\ell + k) \times 2(\ell + k)$  skew-symmetric matrix  $M$ . The final output probabilities are then simulated using  $p(y^*|x) = \text{Pf}(M) = \sqrt{\det(M)}$  where Pf is the Pfaffian of the matrix.

### Measurement Scheme for Kernel Method

Since we employ kernel methods, the output of the circuit is the expectation value of the vacuum state  $|\mathbf{0}\rangle$ ,

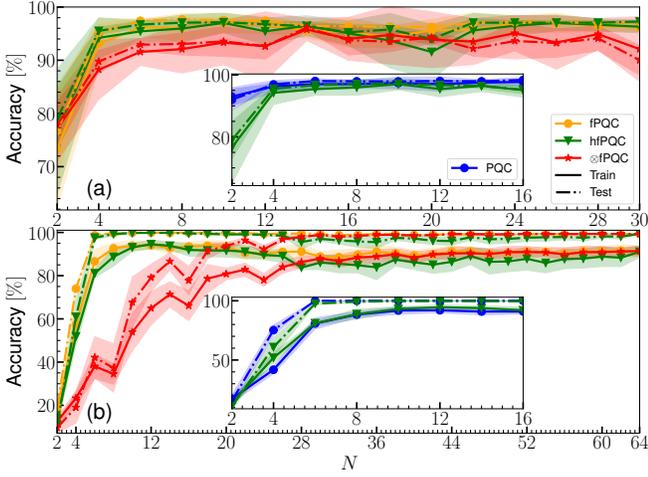


FIG. 3. Comparison of training and test accuracies for fPQC, hfPQC, and  $\otimes$ fPQC for WBC (a) and Digits (b). For each scenario, the insets display a comparative analysis of training and test accuracies for unrestricted PQC and hfPQC. The shaded areas depict the standard deviation of the accuracies.

and corresponds to  $\langle \mathbf{0} | U(x) U^\dagger(x') | \mathbf{0} \rangle$  where  $U(x)$  is the parameterized match-circuit with input data  $x$  and  $x'$ . The kernel is then defined as,

$$\mathcal{U}(x, x') = |\langle \mathbf{0} | U^\dagger(x') U(x) | \mathbf{0} \rangle|^2. \quad (20)$$

and is a similarity measure, portraying the overlap among quantum states related to the data points  $(x, x')$ . In this

specific instance, the Hamming weight remains zero as the expectation value is consistently calculated over the vacuum state and therefore the length of both  $y$  and  $x$  is  $N$ . The final matrix  $M$  is of dimension  $2N$ . The Majorana operators originating from the input bitstring  $(c_{2p_1}, \dots, c_{2p_\ell})$  are therefore absent, leaving only those derived from the conjugation relations in Eqs.(16) and (3). Therefore Eq. (18) reduces to,

$$p(\mathbf{0} | \mathbf{0}) = \prod_{\gamma=1}^N \sum_{m_\gamma, n_\gamma}^{2N} T_{j_\gamma, m_\gamma} T_{j_\gamma, n_\gamma}^* \left\langle \mathbf{0} \left| \prod_{\gamma=1}^N c_{n_\gamma} c_{m_\gamma} \right| \mathbf{0} \right\rangle \quad (21)$$

This equation is equivalent to the one in Eq. (18) and it is equal to the kernel  $\mathcal{U}(x, x')$ .

### Generalization analysis

In Fig. 3 we provide the training and test accuracies for SVMs optimized with PQC, fPQC, hfPQC, and  $\otimes$ fPQC kernels for WBC (top row) and Digits (bottom row) datasets. FermiML models show generalization performance (difference between train and test performance) on par with PQCs. Note that the  $x$ -axes represent the number of qubits in the kernel (and not training iterations as is customary in the ML field).