



Pulse-efficient quantum machine learning

André Melo^{1,2}, Nathan Earnest-Noble ³, and Francesco Tacchino ⁴

¹Kavli Institute of Nanoscience, Delft University of Technology, P.O. Box 4056, 2600 GA Delft, The Netherlands

²IBM Quantum, IBM Netherlands, Amsterdam, NH 1066 VH, The Netherlands

³IBM Quantum, IBM T. J. Watson Research Center, Yorktown Heights, New York 10598, USA

⁴IBM Quantum, IBM Research Europe – Zurich, 8803 Rüschlikon, Switzerland

Quantum machine learning algorithms based on parameterized quantum circuits are promising candidates for near-term quantum advantage. Although these algorithms are compatible with the current generation of quantum processors, device noise limits their performance, for example by inducing an exponential flattening of loss landscapes. Error suppression schemes such as dynamical decoupling and Pauli twirling alleviate this issue by reducing noise at the hardware level. A recent addition to this toolbox of techniques is pulse-efficient transpilation, which reduces circuit schedule duration by exploiting hardware-native cross-resonance interaction. In this work, we investigate the impact of pulse-efficient circuits on near-term algorithms for quantum machine learning. We report results for two standard experiments: binary classification on a synthetic dataset with quantum neural networks and handwritten digit recognition with quantum kernel estimation. In both cases, we find that pulse-efficient transpilation vastly reduces average circuit durations and, as a result, significantly improves classification accuracy. We conclude by applying pulse-efficient transpilation to the Hamiltonian Variational Ansatz and show that it delays the onset of noise-induced barren plateaus.

1 Introduction

Quantum machine learning (QML) is a nascent area of research that has seen rapid developments

André Melo: am@andremelo.org

Francesco Tacchino : fta@zurich.ibm.com

over the last decade [1, 2, 3, 4]. Initial works in the field focused on developing quantum versions of existing classical algorithms, thereby achieving asymptotically faster runtimes [5, 6, 7]. However, these algorithms are beyond the capabilities of current quantum hardware, as they require the execution of rather deep quantum circuits and often rely on specific assumptions like quantum memories [8] to overcome bottlenecks in data loading and readout. More recently, the fast technological progress and wide availability of noisy quantum processors [9, 10] motivated the emergence of a second generation of QML algorithms based on parameterized quantum circuits (PQCs) [11, 12, 13, 14, 15, 16, 17, 18]. In this alternative QML paradigm, quantum computers may function as a co-processor working in tandem with classical computers. Because the circuit Ansätze can be shallow in depth, PQC-based algorithms are in principle compatible with the existing generation of quantum devices.

An important obstacle to the viability of QML in the near term is the presence of hardware noise [19]. Coherent errors often simply shift the position of minima in the loss landscape, in which case they can be trained away [20]. In contrast, errors arising from incoherent noise have more adverse effects that hinder trainability and performance. As an example, incoherent errors cause the loss function of a large family of Ansätze to vanish exponentially with increasing number of layers, a phenomenon known as noise-induced barren plateaus (NIBP) [21]. A similar effect occurs in kernel-based methods where noise can lead to an exponential concentration of the kernel values [22].

A common approach to mitigate the effects of device noise is to use protocols that estimate improved expectation values through classical post-processing, a procedure known as er-

arXiv:2211.01383v2 [quant-ph] 5 Oct 2023

ror mitigation [23, 24, 25, 26, 27, 28]. While schemes such as zero-noise extrapolation [23, 24] and virtual distillation [25] can significantly enhance the performances of noisy processors [29], they also introduce additional experimental or computational overhead and can be ineffective in preventing noise-induced cost concentration [30, 22, 31, 32]. A complementary strategy is to apply error suppression schemes that suppress noise at the hardware level, such as dynamical decoupling [33, 34] and Pauli twirling [35]. A recently-introduced tool for hardware-level error suppression is pulse-efficient (PE) transpilation of cross-resonance gates [36, 37]. The core idea of this technique is to decompose two-qubit gates into hardware-native ones, such as echoed cross-resonance pulses on superconducting architectures based on fixed-frequency qubits. The echoes are then exposed to the transpiler to remove redundant single-qubit rotations. The resulting circuits often have significantly lower schedule durations, thereby mitigating errors introduced by finite coherence times. Crucially, PE transpilation requires no additional overhead or calibration. Refs. [36, 38] applied this technique to combinatorial optimization tasks and demonstrated significant improvements over conventional CNOT-based transpilation. Related ideas were also explored in Refs. [39, 40, 41], which directly optimized pulse parameters to address variational problems. However, a comprehensive study of the impact of PE transpilation on the performance of paradigmatic QML algorithms powered by parameterized quantum circuits was not available until now.

In this work, we demonstrate the application of PE transpilation to three paradigmatic QML tasks. To highlight the general applicability and versatility of our method, we conduct our experiments across three different IBM Quantum [42] backends using qiskit [43]. We begin by training Quantum Neural Networks (QNNs) on a synthetic dataset and observe that PE transpilation significantly improves the resulting training loss and classification accuracy. Afterwards, we apply PE transpilation to a quantum kernel circuit that we use to classify all 10 digits of the commonly-used MNIST dataset. When compared to CNOT-based transpilation, PE transpilation allows us to significantly extend the width of the kernel circuits and achieve a classification accuracy of

$\approx 90\%$. Finally, we explicitly compute the effect of PE transpilation on NIBP. In particular, we study how the loss function of the Hamiltonian Variational Ansatz [44] evolves for increasing number of layers and find that PE transpilation slows down the onset of the NIBP.

2 Application to quantum neural networks

QNNs are one of the leading variational algorithms for QML [45, 46, 11, 47, 48]. A typical QNN architecture for classification tasks is comprised of three main steps. First, classical data $\vec{x} \in \mathbb{R}^n$ is encoded onto a quantum state through a parameterized feature map $\mathcal{U}_{\text{FM}}(\vec{x})$. The resulting state $\mathcal{U}_{\text{FM}}(\vec{x})|0\rangle^{\otimes n}$ is then fed to a variational ansatz $\mathcal{U}_v(\vec{\theta})$, where $\vec{\theta}$ denotes a set of trainable parameters. Finally, the expectation value of an observable \mathcal{O} is measured and classically post-processed to yield the model predictions $y_p = f(\vec{x}|\vec{\theta})$ and the loss function $L(f(\vec{x}|\vec{\theta}), \vec{y})$, where y are the true labels. A classical optimization algorithm then iteratively varies $\vec{\theta}$ to minimize L .

We begin by studying the impact of PE transpilation on a binary classification task with the QNN shown in Fig. 1 (see App. A for a description of the underlying mechanism which enables our specific PE method). We consider an architecture similar to the one reported in Ref. [17] but restrict entangling operations to neighboring pairs of qubits in order to avoid prohibitively large circuit depths. A forward pass of the QNN starts with two layers of the feature map proposed in Ref. [14] (Fig. 1 (b)). A single layer applies Hadamard and R_Z gates on all qubits, where the R_Z rotation angle is related to the feature values by the relation $2x_i$. This is followed by R_{ZZ} operations on neighboring pairs of qubits, where the rotation angle $x_{ij} = 2(\pi - x_i)(\pi - x_j)$ depends on products of the features. We then apply a variational ansatz that consists of parameterized R_Y gates applied on every qubit, followed by CNOTs on neighboring qubits, and a final set of parameterized R_Y (Fig. 1 (a)). The angles θ_i of the R_Y operations are the training parameters that are optimized classically to fit a given target function. Finally, we measure the parity of the output

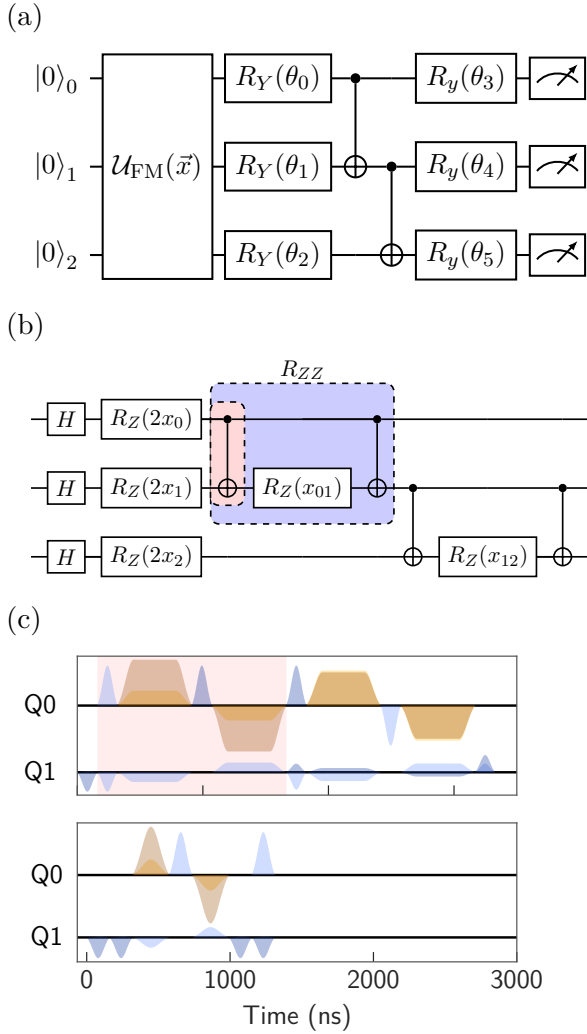


Figure 1: (a) Quantum neural network architecture for binary classification. A forward pass of the network begins with an encoding stage that maps feature vectors \vec{x} to a quantum state through a parameterized feature map $\mathcal{U}_{\text{FM}}(\vec{x})$. The second stage of the network consists of a variational form with parameterized R_Y (whose angles are optimized with a classical routine) and CNOT gates applied on neighboring qubits. Finally, we measure the parity of the qubits in the Z basis, with the fraction of even (odd) bitstrings representing the probability of class 0 (1). (b) A single layer of the parameterized feature map $\mathcal{U}_{\text{FM}}(\vec{x})$. Hadamard gates are applied on every qubit, followed by R_{ZZ} rotations on every pair of neighboring qubits (highlighted in blue). (c) Pulse schedules that implement an $R_{ZZ}(0.5)$ gate on *ibmq_guadalupe* through a conventional CNOT-based approach (top panel) and pulse-efficient transpilation (bottom panel). The section highlighted in red in the top panel corresponds to the pulses that implement a single CNOT gate through cross-resonance. Pulse-efficient transpilation significantly decreases the schedule duration resulting in higher circuit fidelities.

bit strings

$$m(\vec{x}, \vec{\theta}) = |\langle 0 | \mathcal{U}_{\text{FM}}^\dagger(\vec{x}) \mathcal{U}_V^\dagger(\vec{\theta}) P \mathcal{U}_V(\vec{\theta}) \mathcal{U}_{\text{FM}}(\vec{x}) | 0 \rangle|^2 \quad (1)$$

where $P = \prod_i Z_i$ is the parity operator and Z_i is the standard Pauli Z operator acting on the i -th qubit. We associate even parity with class 0 and odd parity with class 1.

We benchmark the QNN performance on a synthetic two-class dataset with standard and PE transpilation. To ensure the QNN can distinguish between the two classes with high accuracy, the dataset is generated via the QNN model by fixing the trainable parameters $\vec{\theta}_s$ in such a way that the separation of the classes is maximised. The training procedure is then carried out starting from a new, random initialisation of the parameters, and should ideally recover the set used to generate the data. More formally, we uniformly sample 600 feature vectors $\vec{x} \in [0, 1]^n$ and compute their parity $m(\vec{x}, \vec{\theta})$ through noiseless simulations. Using the L-BFGS-B optimizer [49], we search for QNN parameters $\vec{\theta}_s$ that maximize the average absolute parity $\frac{1}{600} \sum_i |m(\vec{x}_i, \vec{\theta}_s)|$. Out of this set of feature vectors, we further select the 50 samples for each class with the largest absolute parity expectation value. Fig. 2(a) shows an example of the resulting dataset for 2 qubits alongside the probability of observing class 0 $p(y = 0, \vec{x}, \vec{\theta}_s)$. We observe that the decision boundary of the QNN correctly separates the two classes.

We train the QNNs on *ibmq_jakarta* using a cross-entropy loss function and 50 iterations of Spall's SPSA stochastic gradient descent algorithm [50] with an automated calibration phase [44] of 50 iterations. Moreover, we apply readout error mitigation [51] and use 100 samples both for training and testing the networks. We show example training curves for $n = 3$ qubits in Fig. 2(b) which converge almost immediately after the initial calibration stage. In Fig. 3(a-b), we compare the training loss and testing classification accuracy of pulse-efficient and regular QNNs with $n = 2$ to 5 qubits. While the performance of the standard QNN deteriorates rapidly after $n = 2$, the PE QNN remains competitive with the performance of the noiseless simulation throughout the whole range of n . We attribute this improvement to a reduction in incoherent error due to the shorter schedule duration in PE circuits (Fig. 3(c)). More specifically, as we show in Fig. 1(c) PE transpilation significantly short-

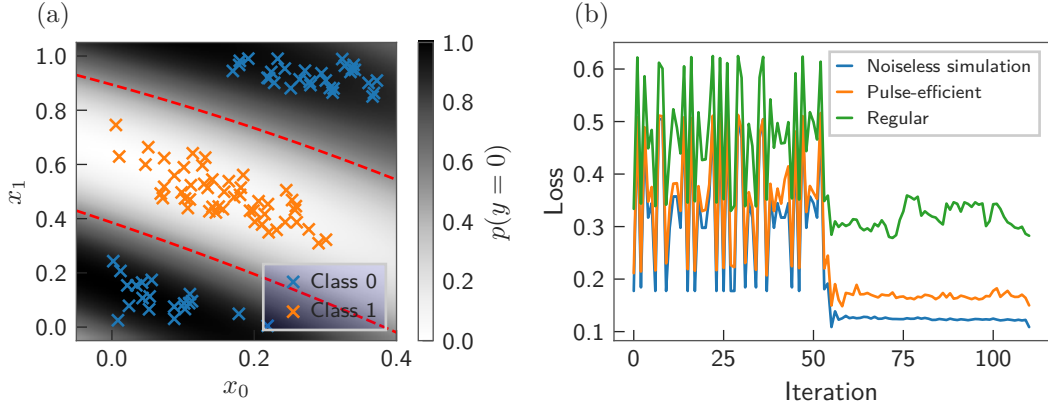


Figure 2: Training a quantum neural network on *ibmq_jakarta*. (a) Example of a synthetic two-class dataset generated from the QNN shown in Fig. 1 with $n = 2$ features. The heat map represents the probability assigned to class 0 by the QNN with the weights set to those used to generate the binary dataset. The red dashed lines correspond to the decision boundaries $p(y = 0) = p(y = 1) = 0.5$ of the model. (b) Convergence of the training loss on $n = 4$ qubits dataset after 120 iterations of the SPSA algorithm. The blue, orange and green curves show the training loss of the simulated, pulse-efficient, and regular quantum neural networks, respectively.

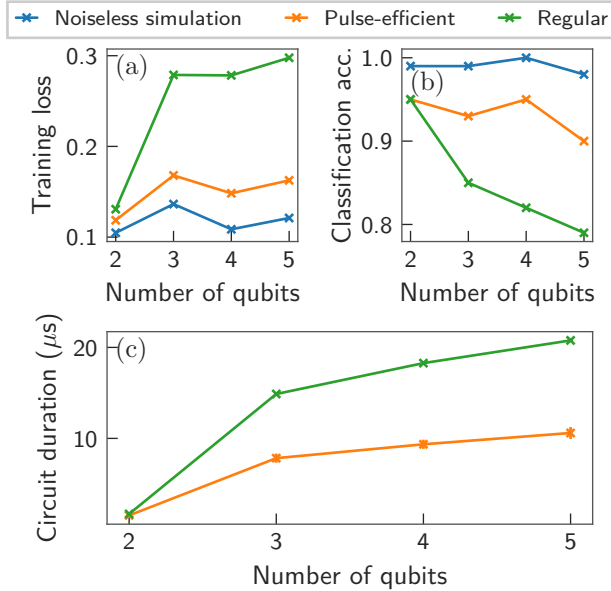


Figure 3: Comparison of the performance of quantum neural networks with pulse-efficient and regular transpilation. (a) Training loss after 120 iterations of the SPSA algorithm. (b) Classification accuracy on the test set. (c) Average schedule duration of QNN circuit. The pulse-efficient circuits have significantly lower schedule duration, which improves circuit fidelity and classification accuracy.

ens the duration of R_{ZZ} gates [36] and hence of the feature map portion of the circuit.

3 Application to quantum kernels

We now turn to investigating the impact of PE transpilation on fidelity-based quantum kernel classification. This class of algorithms uses a quantum feature map to compute a similarity measure between input data points [15, 14, 52, 18, 53]

$$K(\vec{x}_i, \vec{x}_j) = |\langle 0 | \mathcal{U}_{\text{FM}}^\dagger(\vec{x}_i) \mathcal{U}_{\text{FM}}(\vec{x}_j) | 0 \rangle|^2. \quad (2)$$

The resulting Gram matrix K is then fed to a classical kernel method such as a support vector machine [54] to predict the corresponding labels \vec{y} .

For this experiment we use the same feature map as in the QNN case presented above, but increase the depth to 4 in order to achieve higher classification accuracy. Following the approach outlined in Ref. [14], we estimate the feature vector kernel function for all pairs of training data \vec{x}_i, \vec{x}_j using 8192 shots. Specifically, we apply the circuit $\mathcal{U}_{\text{FM}}^\dagger(\vec{x}_i) \mathcal{U}_{\text{FM}}(\vec{x}_j) | 0 \rangle^{\otimes n}$ and then measure all qubits in the Z basis. The kernel entry $K(\vec{x}_i, \vec{x}_j)$ then corresponds to the frequency of the zero bitstring 0^n . Having repeated this process for all the training data, we feed the resulting kernel matrix to a conventional support vector machine implemented with *scikit-learn* [55]. For this classification task, we choose the MNIST dataset, a popular real-world database of handwritten digits [56], and use 10 training and testing

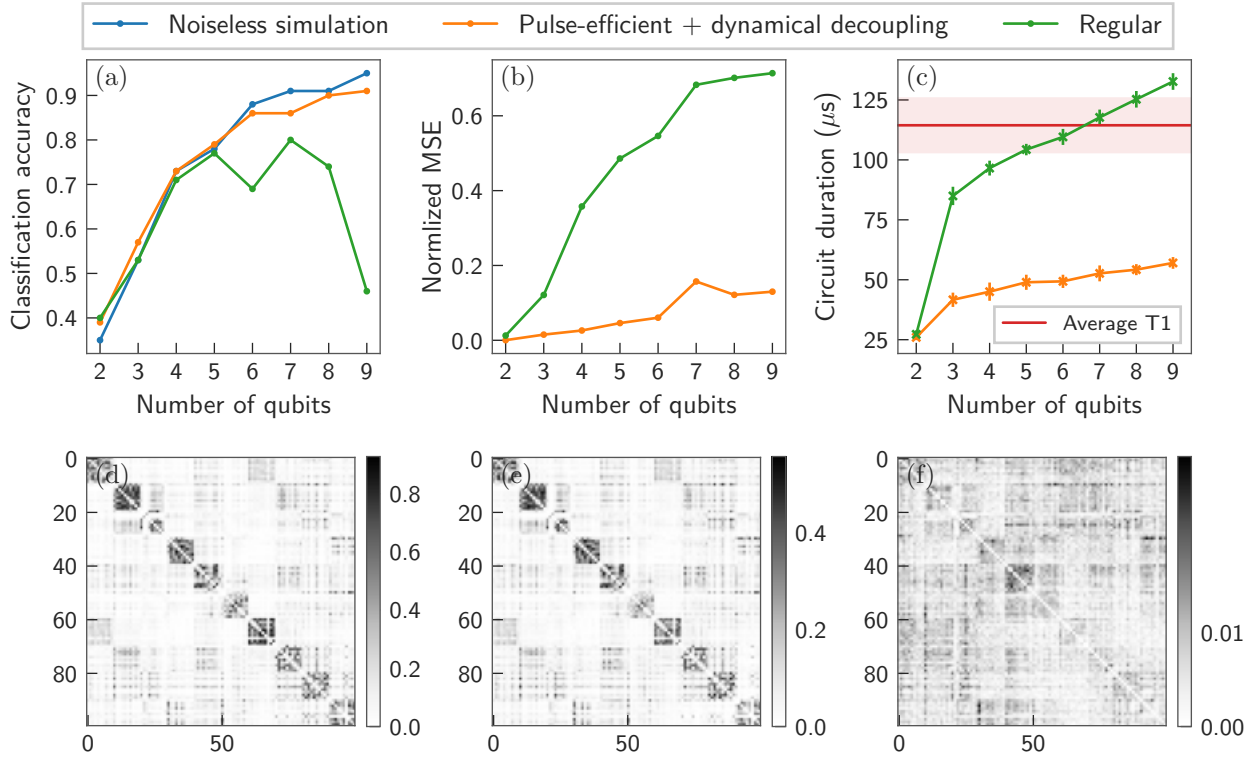


Figure 4: Impact of pulse-efficient transpilation on quantum kernel classification run on *ibmq_montreal*. (a) Classification accuracy on test dataset for a noiseless simulation, pulse-efficient, and regular circuits. (b) Normalized mean square error of the training kernel matrix compared to the simulated kernel matrix. (c) Average circuit duration for pulse-efficient and regular circuits compared to the T_1 . In the bottom panel we show the training kernel matrices at $n = 9$ qubits obtained with (d) noiseless simulations, (e) pulse-efficient circuits, and (f) regular circuits.

samples for each of the ten digits. The kernel chosen for this experiment uses a the number of input features equal to the number of qubits. In the case of the MNIST dataset, the resolution of the images exceeds the number of qubits we use. We therefore reduce the number of features through a truncated singular value decomposition, a standard dimensionality reduction procedure.

Due to the sparsity of the kernel circuits, the qubits experience large idle times that lead to error accumulation [57]. To mitigate this source of noise, we combine PE transpilation with a dynamical decoupling protocol. Whenever qubit i has an idle time T_{idle} larger than twice the single qubit gate time, we apply a dynamical decoupling sequence $\tau/2 - X_p - \tau - X_m - \tau/2$ with delay times $\tau = T_{\text{idle}} - 2T_{X_{p/m}}$ where $X_{p/m}$ are positive/negative π pulses around the x -axis and $T_{X_{p/m}}$ their duration.

We run the kernel circuits on linearly connected subsets of qubits on *ibmq_montreal* with PE and regular transpilation, along with noiseless simulations. Fig. 4(a) shows the testing classification

accuracy as a function of the number of qubits for all three methods. Focusing first on the simulated curve, the classification accuracy increases monotonically with the number of qubits. This occurs because the number of qubits increases concomitantly with the number of training features, which makes it easier to distinguish different digits. Turning to the device runs, the performance of the regular circuit remains very close to the ideal curve up to 5 qubits, after which it degrades rapidly and stays below 80%. This sharp turning point coincides with the average circuit duration becoming comparable with the average device T_1 , see Fig 4(c). In contrast, the PE transpilation circuit durations are always well below the coherence limit of the device, thereby yielding classification accuracies that reach 90% and closely track the simulated values. To further quantify the performance of the device runs, in Fig 4(c) we show the normalized mean square error of the experimental kernel matrices compared

to the simulated matrix K^{sim} :

$$\text{NMSE}(K) = \frac{\sum_{ij}(K_{ij}^{\text{sim}} - K_{ij})^2}{\sum_{ij}(K_{ij}^{\text{sim}})^2}. \quad (3)$$

The error curves show a similar trend to the average circuit durations, with the error of the regular circuits increasing much faster than the PE circuits. In Fig. 4(d-f) we show training kernel matrices with $n = 9$ qubits for all three methods. The PE transpilation kernel matrix is close to the simulated matrix and has an approximately block-diagonal structure, indicating the feature map is capable of separating the digits with high accuracy. In contrast, the regular transpilation matrix is mostly devoid of structure, which results in significantly lower classification accuracy. Moreover, its matrix elements are close to 0, signaling that the underlying bitstring distribution is extremely noisy.

In App. B we perform additional experiments to estimate how much of the performance boost we observe can be attributed to PE transpilation versus dynamical decoupling and find that PE transpilation is the main driver of the improvement.

4 Impact on noise-induced barren plateaus

In our last experiment, we investigate the impact of PE transpilation on NIBP. We implement the Hamiltonian Variational Ansatz for the transverse field Ising model as considered in Refs. [58] and [21]. We consider a linearly connected chain of spins, such that a single layer of the ansatz is given by

$$U_{\text{TFIM}} = \exp(-iH_x) \exp(-iH_{zz}), \quad (4)$$

where

$$H_x = \sum_{i=0}^{N-1} \gamma_i \sigma_i^x, \quad (5)$$

$$H_{zz} = \sum_{i=0}^{N-2} \beta_{i,i+1} \sigma_i^z \sigma_{i+1}^z, \quad (6)$$

and σ_α^i are the conventional Pauli matrices acting on the i -th qubit. To study the onset of NIBP, we study the behavior of a local observable with increasing number of qubits n and ansatz layers L . Following [21], we measure the local parity of the

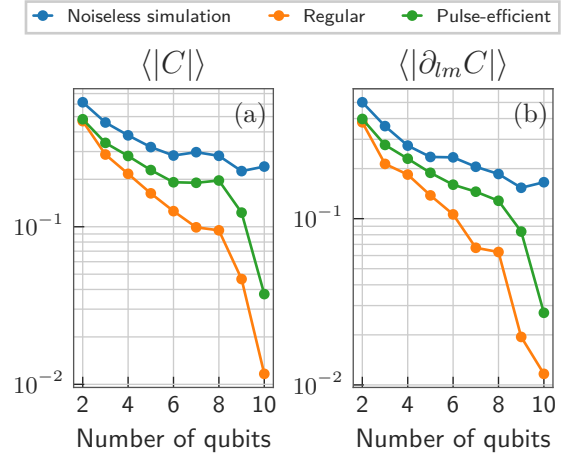


Figure 5: Mitigation of noise-induced barren plateaus through pulse-efficient transpilation. We consider an ansatz similar to the Hardware Variational Ansatz with a number of layers that increases linearly with the number of qubits. By sampling over 100 sets of random parameters we compute the average (a) loss function, and (b) partial derivative with respect to the last $\beta_{i,i+1}$ parameter.

first two qubits $O = Z_0 Z_1$ along with its derivative with respect to the last $\beta_{i,i+1}$ parameter. Further, we set the number of layers to increase linearly with the number of qubits $L = 2(n - 1)$ and perform our runs on *ibmq_montreal*.

In Figure 5 we show the cost function and its partial derivative with respect to the last $\beta_{i,i+1}$ parameter averaged over 100 random parameter sets. In the case of noiseless simulations, both curves appear to slowly decay polynomially with increasing number of qubits. On the other hand, the device runs show a noticeable exponential decay starting at around $n = 8$ qubits, which we attribute to the onset of NIBP. However, both the average loss function and derivative of the PE transpilation circuits are consistently above those of the regular circuits. Although PE transpilation does not remove NIBP, it has more favorable scaling which would allow going to higher depths of number of qubits when compared with regular transpilation.

5 Discussion

In this work, we studied the impact of PE transpilation on the performance of near-term QML algorithms. We began by performing binary classification of a synthetic dataset with a QNN, where we found that PE circuits achieved significantly

higher classification accuracy and lower training loss. Secondly, we used quantum kernel estimation to classify a real-world dataset of handwritten digits. Combining PE transpilation with dynamical decoupling allowed us to accurately estimate kernels up to 9 qubits and achieve 90% classification accuracy, whilst regular circuits remain below 80%. Lastly, we studied the onset of NIBP on a commonly-used ansatz for quantum chemistry. Our results show that PE transpilation slows down the onset of NIBP, which allows executing variational quantum algorithms at higher numbers of qubits when compared with regular transpilation.

Our results highlight a key advantage of PE transpilation, namely that it introduces no additional overhead or calibrations and is compatible with most qiskit-based programs with minimal modifications. Furthermore, we observe that it consistently improves circuit performance across different models and devices. This makes our proposed approach particularly appealing for applications and use cases that rely on remote device access and control. We expect that these improvements extend to most protocols featuring parameterized $R_{ZX}(\theta)$ gates: these natively appear in a broad class of quantum algorithms, such as Hamiltonian simulation schemes [59, 60] – with potential applications to optimization [61] and sampling problems [62, 63] – and unitary coupled cluster circuits in quantum chemistry.

While finalizing this work, we became aware of a recent preprint [64] that also applies PE transpilation to PQCs designed for quantum chemistry and optimization tasks.

Acknowledgements

We are thankful to Caroline Tornow, Daniel J. Egger and Kunal Sharma for useful discussions. We acknowledge the use of IBM Quantum services for this work. IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. The current list of IBM trademarks is available at <https://www.ibm.com/legal/copytrade>.

References

- [1] Peter Wittek. “Quantum machine learning”. Elsevier. (2014).
- [2] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. “Quantum machine learning”. *Nature* **549**, 195–202 (2017).
- [3] Maria Schuld and Francesco Petruccione. “Machine learning with quantum computers”. Springer. (2021).
- [4] S. Mangini, F. Tacchino, D. Gerace, D. Bajoni, and C. Macchiavello. “Quantum computing models for artificial neural networks”. *Europhysics Letters* **134**, 10002 (2021).
- [5] Nathan Wiebe, Daniel Braun, and Seth Lloyd. “Quantum Algorithm for Data Fitting”. *Phys. Rev. Lett.* **109**, 050505 (2012).
- [6] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. “Quantum algorithms for supervised and unsupervised machine learning” (2013). [arXiv:1307.0411](https://arxiv.org/abs/1307.0411).
- [7] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. “Quantum Support Vector Machine for Big Data Classification”. *Phys. Rev. Lett.* **113**, 130503 (2014).
- [8] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. “Quantum random access memory”. *Phys. Rev. Lett.* **100**, 160501 (2008).
- [9] John Preskill. “Quantum Computing in the NISQ era and beyond”. *Quantum* **2**, 79 (2018).
- [10] Antonio D. Corcoles, Abhinav Kandala, Ali Javadi-Abhari, Douglas T. McClure, Andrew W. Cross, Kristan Temme, Paul D. Nation, Matthias Steffen, and Jay M. Gambetta. “Challenges and opportunities of near-term quantum computing systems”. *Proceedings of the IEEE* **108**, 1338–1352 (2020).
- [11] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. “Parameterized quantum circuits as machine learning models”. *Quantum Science and Technology* **4**, 043001 (2019).
- [12] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. “Variational quantum algorithms”. *Nature Reviews Physics* **3**, 625–644 (2021).

- [13] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermann Heimonen, Jakob S. Kottmann, Tim Menke, Wai-Keong Mok, Sukin Sim, Leong-Chuan Kwek, and Alán Aspuru-Guzik. “Noisy intermediate-scale quantum algorithms”. *Rev. Mod. Phys.* **94**, 015004 (2022).
- [14] 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**, 209–212 (2019).
- [15] Maria Schuld and Nathan Killoran. “Quantum Machine Learning in Feature Hilbert Spaces”. *Phys. Rev. Lett.* **122**, 040504 (2019).
- [16] Francesco Tacchino, Stefano Mangini, Panagiotis Kl. Barkoutsos, Chiara Macchiavello, Dario Gerace, Ivano Tavernelli, and Daniele Bajoni. “Variational learning for quantum artificial neural networks”. *IEEE Transactions on Quantum Engineering* **2**, 1–10 (2021).
- [17] Amira Abbas, David Sutter, Christa Zoufal, Aurelien Lucchi, Alessio Figalli, and Stefan Woerner. “The power of quantum neural networks”. *Nature Computational Science* **1**, 403–409 (2021).
- [18] 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**, 2631 (2021).
- [19] Daniel Stilck França and Raul Garcia-Patron. “Limitations of optimization algorithms on noisy quantum devices”. *Nature Physics* **17**, 1221–1227 (2021).
- [20] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. “The theory of variational hybrid quantum-classical algorithms”. *New Journal of Physics* **18**, 023023 (2016).
- [21] Samson Wang, Enrico Fontana, M. Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J. Coles. “Noise-induced barren plateaus in variational quantum algorithms”. *Nature Communications* **12**, 6961 (2021).
- [22] Supanut Thanasilp, Samson Wang, M. Cerezo, and Zoë Holmes. “Exponential concentration and untrainability in quantum kernel methods” (2022). [arXiv:2208.11060](https://arxiv.org/abs/2208.11060).
- [23] Kristan Temme, Sergey Bravyi, and Jay M. Gambetta. “Error mitigation for short-depth quantum circuits”. *Phys. Rev. Lett.* **119**, 180509 (2017).
- [24] Suguru Endo, Simon C. Benjamin, and Ying Li. “Practical quantum error mitigation for near-future applications”. *Phys. Rev. X* **8**, 031027 (2018).
- [25] William J. Huggins, Sam McArdle, Thomas E. O’Brien, Joonho Lee, Nicholas C. Rubin, Sergio Boixo, K. Birgitta Whaley, Ryan Babbush, and Jarrod R. McClean. “Virtual distillation for quantum error mitigation”. *Phys. Rev. X* **11**, 041036 (2021).
- [26] Bálint Koczor. “Exponential error suppression for near-term quantum devices”. *Phys. Rev. X* **11**, 031057 (2021).
- [27] Philippe Suchsland, Francesco Tacchino, Mark H. Fischer, Titus Neupert, Panagiotis Kl. Barkoutsos, and Ivano Tavernelli. “Algorithmic Error Mitigation Scheme for Current Quantum Processors”. *Quantum* **5**, 492 (2021).
- [28] Zhenyu Cai, Ryan Babbush, Simon C. Benjamin, Suguru Endo, William J. Huggins, Ying Li, Jarrod R. McClean, and Thomas E. O’Brien. “Quantum error mitigation” (2023). [arXiv:2210.00921](https://arxiv.org/abs/2210.00921).
- [29] Abhinav Kandala, Kristan Temme, Antonio D Córcoles, Antonio Mezzacapo, Jerry M Chow, and Jay M Gambetta. “Error mitigation extends the computational reach of a noisy quantum processor”. *Nature* **567**, 491–495 (2019).
- [30] Samson Wang, Piotr Czarnik, Andrew Arasmith, M. Cerezo, Lukasz Cincio, and Patrick J. Coles. “Can error mitigation improve trainability of noisy variational quantum algorithms?” (2021). [arXiv:2109.01051](https://arxiv.org/abs/2109.01051).
- [31] Yihui Quek, Daniel Stilck França, Sumeet Khatri, Johannes Jakob Meyer, and Jens Eisert. “Exponentially tighter bounds on limitations of quantum error mitigation” (2023). [arXiv:2210.11505](https://arxiv.org/abs/2210.11505).
- [32] Kento Tsubouchi, Takahiro Sagawa, and Nobuyuki Yoshioka. “Universal cost bound of quantum error mitigation based on quantum estimation theory” (2023). [arXiv:2208.09385](https://arxiv.org/abs/2208.09385).
- [33] Lorenza Viola, Emanuel Knill, and Seth Lloyd. “Dynamical decoupling of open quantum systems”. *Phys. Rev. Lett.* **82**, 2417–2421 (1999).

- [34] Bibek Pokharel, Namit Anand, Benjamin Fortman, and Daniel A. Lidar. “Demonstration of fidelity improvement using dynamical decoupling with superconducting qubits”. *Phys. Rev. Lett.* **121**, 220502 (2018).
- [35] Zhenyu Cai, Xiaosi Xu, and Simon C Benjamin. “Mitigating coherent noise using pauli conjugation”. *npj Quantum Information* **6**, 1–9 (2020).
- [36] Nathan Earnest, Caroline Tornow, and Daniel J Egger. “Pulse-efficient circuit transpilation for quantum applications on cross-resonance-based hardware”. *Phys. Rev. Research* **3**, 043088 (2021).
- [37] John P. T. Stenger, Nicholas T. Bronn, Daniel J. Egger, and David Pekker. “Simulating the dynamics of braiding of Majorana zero modes using an IBM quantum computer”. *Phys. Rev. Research* **3**, 033171 (2021).
- [38] Siyuan Niu and Aida Todri-Sanial. “Effects of Dynamical Decoupling and Pulse-Level Optimizations on IBM Quantum Computers”. *IEEE Transactions on Quantum Engineering* **3**, 1–10 (2022).
- [39] Oinam Romesh Meitei, Bryan T Gard, George S Barron, David P Pappas, Sophia E Economou, Edwin Barnes, and Nicholas J Mayhall. “Gate-free state preparation for fast variational quantum eigensolver simulations”. *npj Quantum Information* **7**, 1–11 (2021).
- [40] Z. Liang, H. Wang, J. Cheng, Y. Ding, H. Ren, Z. Gao, Z. Hu, D. S. Boning, X. Qian, S. Han, W. Jiang, and Y. Shi. “Variational Quantum Pulse Learning”. *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 556–565 (2022).
- [41] Zhiding Liang, Jinglei Cheng, Hang Ren, Hanrui Wang, Fei Hua, Zhixin Song, Yongshan Ding, Fred Chong, Song Han, Yiyu Shi, and Xuehai Qian. “NAPA: Intermediate-level Variational Native-pulse Ansatz for Variational Quantum Algorithms” (2023). [arXiv:2208.01215](https://arxiv.org/abs/2208.01215).
- [42] “IBM Quantum”. url: <https://quantum-computing.ibm.com/>.
- [43] Gadi Aleksandrowicz, Thomas Alexander, Panagiotis Barkoutsos, Luciano Bello, Yael Ben-Haim, David Bucher, Francisco Jose Cabrera-Hernández, Jorge Carballo-Franquis, Adrian Chen, Chun-Fu Chen, Jerry M. Chow, and *et al.* “Qiskit: An Open-source Framework for Quantum Computing”. url: <https://doi.org/10.5281/zenodo.2562111>.
- [44] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta. “Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets”. *Nature* **549**, 242–246 (2017).
- [45] Edward Farhi and Hartmut Neven. “Classification with quantum neural networks on near term processors” (2018). [arXiv:1802.06002](https://arxiv.org/abs/1802.06002).
- [46] Maria Schuld, Alex Bocharov, Krysta M. Svore, and Nathan Wiebe. “Circuit-centric quantum classifiers”. *Phys. Rev. A* **101**, 032308 (2020).
- [47] Junyu Liu, Francesco Tacchino, Jennifer R. Glick, Liang Jiang, and Antonio Mezzacapo. “Representation learning via quantum neural tangent kernels”. *PRX Quantum* **3**, 030323 (2022).
- [48] Junyu Liu, Khadijeh Najafi, Kunal Sharma, Francesco Tacchino, Liang Jiang, and Antonio Mezzacapo. “Analytic theory for the dynamics of wide quantum neural networks”. *Phys. Rev. Lett.* **130**, 150601 (2023).
- [49] Ciyu Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. “Algorithm 778: L-bfgsb: Fortran subroutines for large-scale bound-constrained optimization”. *ACM Transactions on mathematical software (TOMS)* **23**, 550–560 (1997).
- [50] James C Spall. “Adaptive stochastic approximation by the simultaneous perturbation method”. *IEEE transactions on automatic control* **45**, 1839–1853 (2000).
- [51] Sergey Bravyi, Sarah Sheldon, Abhinav Kandala, David C. McKay, and Jay M. Gambetta. “Mitigating measurement errors in multiqubit experiments”. *Phys. Rev. A* **103**, 042605 (2021).
- [52] C. M. Wilson, J. S. Otterbach, N. Tezak, R. S. Smith, A. M. Polloreno, Peter J. Karalekas, S. Heidel, M. Sohaib Alam, G. E. Crooks, and M. P. da Silva. “Quantum Kitchen Sinks: An algorithm for machine learning on near-term quantum computers” (2019). [arXiv:1806.08321](https://arxiv.org/abs/1806.08321).
- [53] Maria Schuld. “Supervised quantum machine learning models are kernel methods” (2021). [arXiv:2101.11020](https://arxiv.org/abs/2101.11020).
- [54] Corinna Cortes and Vladimir Vapnik.

- “Support-vector networks”. *Mach. Learn.* **20**, 273–297 (1995).
- [55] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. “Scikit-learn: Machine learning in Python”. *Journal of Machine Learning Research* **12**, 2825–2830 (2011).
- [56] Dheeru Dua and Casey Graff. “UCI machine learning repository”. url: <http://archive.ics.uci.edu/ml>.
- [57] Jennifer R. Glick, Tanvi P. Gujarati, Antonio D. Corcoles, Youngseok Kim, Abhinav Kandala, Jay M. Gambetta, and Kristan Temme. “Covariant quantum kernels for data with group structure” (2022). [arXiv:2105.03406](https://arxiv.org/abs/2105.03406).
- [58] Roeland Wiersema, Cunlu Zhou, Yvette de Sereville, Juan Felipe Carrasquilla, Yong Baek Kim, and Henry Yuen. “Exploring entanglement and optimization within the hamiltonian variational ansatz”. *PRX Quantum* **1**, 020319 (2020).
- [59] Alexander Miessen, Pauline J Ollitrault, Francesco Tacchino, and Ivano Tavernelli. “Quantum algorithms for quantum dynamics”. *Nature Computational Science* **3**, 25–37 (2023).
- [60] Youngseok Kim, Christopher J. Wood, Theodore J. Yoder, Seth T. Merkel, Jay M. Gambetta, Kristan Temme, and Abhinav Kandala. “Scalable error mitigation for noisy quantum circuits produces competitive expectation values”. *Nature Physics* **19**, 752–759 (2023).
- [61] Johannes Weidenfeller, Lucia C. Valor, Julien Gacon, Caroline Tornow, Luciano Bello, Stefan Woerner, and Daniel J. Egger. “Scaling of the quantum approximate optimization algorithm on superconducting qubit based hardware”. *Quantum* **6**, 870 (2022).
- [62] Guglielmo Mazzola. “Sampling, rates, and reaction currents through reverse stochastic quantization on quantum computers”. *Phys. Rev. A* **104**, 022431 (2021).
- [63] David Layden, Guglielmo Mazzola, Ryan V Mishmash, Mario Motta, Pawel Wocjan, Jin-Sung Kim, and Sarah Sheldon. “Quantum-enhanced markov chain Monte Carlo”. *Nature* **619**, 282–287 (2023).
- [64] Mohammad M. Ibrahim, Hamed Mohammadbagherpoor, Cynthia Rios, Nicholas T. Bronn, and Gregory T. Byrd. “Evaluation of parameterized quantum circuits with cross-resonance pulse-driven entanglers”. *IEEE Transactions on Quantum Engineering* **3**, 1–13 (2022).
- [65] Jerry M Chow, Antonio D Corcoles, Jay M Gambetta, Chad Rigetti, Blake R Johnson, John A Smolin, Jim R Rozen, George A Keefe, Mary B Rothwell, Mark B Ketchen, et al. “Simple all-microwave entangling gate for fixed-frequency superconducting qubits”. *Phys. Rev. Lett.* **107**, 080502 (2011).
- [66] Easwar Magesan and Jay M Gambetta. “Effective hamiltonian models of the cross-resonance gate”. *Phys. Rev. A* **101**, 052308 (2020).
- [67] Moein Malekakhlagh, Easwar Magesan, and David C McKay. “First-principles analysis of cross-resonance gate operation”. *Phys. Rev. A* **102**, 042605 (2020).
- [68] Sarah Sheldon, Easwar Magesan, Jerry M Chow, and Jay M Gambetta. “Procedure for systematically tuning up cross-talk in the cross-resonance gate”. *Phys. Rev. A* **93**, 060302 (2016).
- [69] Neereja Sundaresan, Isaac Lauer, Emily Pritchett, Easwar Magesan, Petar Jurcevic, and Jay M Gambetta. “Reducing unitary and spectator errors in cross resonance with optimized rotary echoes”. *PRX Quantum* **1**, 020318 (2020).

A Pulse-efficient cross-resonance circuits

In this appendix we briefly review pulse-efficient transpilation for cross-resonance-based hardware (typically coupled fixed frequency transmons). The cross-resonance interaction arises by driving a control qubit at the target qubit’s frequency. Within the two-level approximation, the resulting time-independent Hamiltonian reads

$$H_{CR} = \frac{1}{2} (Z \otimes B + I \otimes C), \quad (7)$$

where $B = \omega_{ZI}I + \omega_{ZX}X + \omega_{ZY}Y$, $C = \omega_{IX}X + \omega_{IY}Y + \omega_{IZ}Z$, and I, X, Y, Z are Pauli matrices [65, 66, 67]. By using echoed cross-resonance

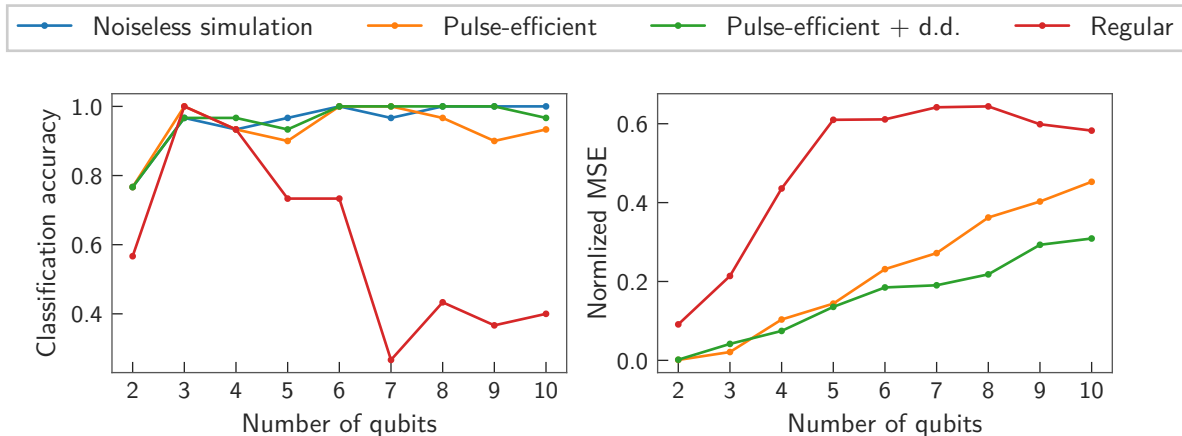


Figure 6: Impact of pulse-efficient transpilation on quantum kernel classification run on *ibmq_gaudalupe*.

pulses with rotary tones [68, 69], it is possible to isolate the ZX interaction and thus implement the unitary $R_{ZX}(\theta) = \exp\{-iZX\theta/2\}$ with good accuracy. To first approximation, the rotation angle of this conditional rotation is given by $\theta = t_{CR}\omega_{ZX}(A)$, where $\omega_{ZX}(A)$ is a non-linear interaction term that depends on the amplitude of the cross-resonance pulse. IBM Quantum backends leverage the cross-resonance interaction to implement CNOT gates constructed with echoed rotations $R_{ZX}(\pi/2) = \text{CR}(\pi/4)X\text{CR}(-\pi/4)X$. Here, CR are the non-echoed cross-resonance pulses, typically shaped as flat-top gaussians. Together with a complete set of one-qubit gates, this CNOT gate is then used as a primitive to synthesize arbitrary two-qubit gates. However, it is possible to implement arbitrary rotations $R_{ZX}(\theta)$ by appropriately scaling the cross-resonance pulses [36]. The core idea of PE transpilation is to leverage this native parametric gate to decrease circuit duration and achieve higher fidelities. The scheme works as follows. Using Cartan’s decomposition, we first rewrite a CNOT-transpiled circuit in terms of parameterized, non-echoed $R_{ZX}(\theta)$ rotations. Then, we expand the $R_{ZX}(\theta)$ gates and expose its echoed implementation $\text{CR}(\theta/2)X\text{CR}(-\theta/2)X$. A final transpilation pass removes redundant single-qubit rotations, leaving at most one single-qubit rotation between non-echoed R_{ZX} pulses. Though non-linear behaviour can result in coherent over or under rotations, this method achieves significant reductions in circuit duration for certain gates (such as the $R_{ZZ}(\theta)$ interaction used in this work) compared to conventional CNOT-based transpi-

lation.

B Pulse-efficient kernel classification without dynamical decoupling

In Section 3, we showed kernel estimation results for circuits with PE transpilation and dynamical decoupling. A natural follow-up question is how much of the performance improvement can be attributed to each of the two error suppression strategies. To address this question, we run a smaller set of experiments on *ibmq_gaudalupe* to classify digits 0, 7, and 9. We execute the kernels with PE transpilation with and without dynamical decoupling and show the resulting classification accuracy and NMSE in Fig. 6. Although dynamical decoupling has a sizeable effect, we conclude that PE transpilation is the primary driver of the performance improvement over regular circuits.