

# Quantum Machine Learning for Climate Modelling

Mierk Schwabe\*, Lorenzo Pastori\*, Valentina Sarandrea\*, and Veronika Eyring\*<sup>†</sup>

\*Deutsches Zentrum für Luft- und Raumfahrt e.V.

Institut für Physik der Atmosphäre, Oberpfaffenhofen, Germany

<sup>†</sup>University of Bremen, Institute of Environmental Physics (IUP), Bremen, Germany

Email: mierk.schwabe@dlr.de

**Abstract**—Quantum machine learning (QML) is making rapid progress, and QML-based models hold the promise of quantum advantages such as potentially higher expressivity and generalizability than their classical counterparts. Here, we present work on using a quantum neural net (QNN) to develop a parameterization of cloud cover for an Earth system model (ESM). ESMs are needed for predicting and projecting climate change, and can be improved in hybrid models incorporating both traditional physics-based components as well as machine learning (ML) models. We show that a QNN can predict cloud cover with a performance similar to a classical NN with the same number of free parameters and significantly better than the traditional scheme. We also analyse the learning capability of the QNN in comparison to the classical NN and show that, at least for our example, QNNs learn more consistent relationships than classical NNs.

**Index Terms**—quantum machine learning, explainable ai, climate modelling

## I. INTRODUCTION

Earth system models (ESMs) are global coupled models of the different components of the Earth system and their interactions, including the atmosphere, ocean, land, and cryosphere. They are used to project future climate change for better climate change mitigation and adaptation strategies. ESMs used for projections of the future climate typically have resolutions of the order of 10s of km. Higher resolutions are possible, but are extremely computationally demanding and thus cannot be used for ensemble generation, which is needed for uncertainty quantification. Even high resolution models cannot resolve all processes, such as cloud microphysics and turbulence, and some long-standing biases remain [3], [4]. At the typical coarse resolutions, also various further dynamical effects are not resolved, e.g., gravity waves and convection. The influences of these unresolved processes on the climate are taken into account with empirical *parameterizations*. The use of these parameterizations leads to uncertainties and biases in climate models. Machine learning (ML) has already demonstrated the capability to enhance physics-based ESMs [5]. In hybrid (ML & physics) models individual parameterizations or combinations of various parameterizations are replaced with

This project was made possible by the DLR Quantum Computing Initiative and the Federal Ministry for Economic Affairs and Climate Action; qci.dlr.de/projects/klim-qml. V.E. was additionally supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) through the Gottfried Wilhelm Leibniz Prize awarded to Veronika Eyring (Reference No. EY 22/2-1).

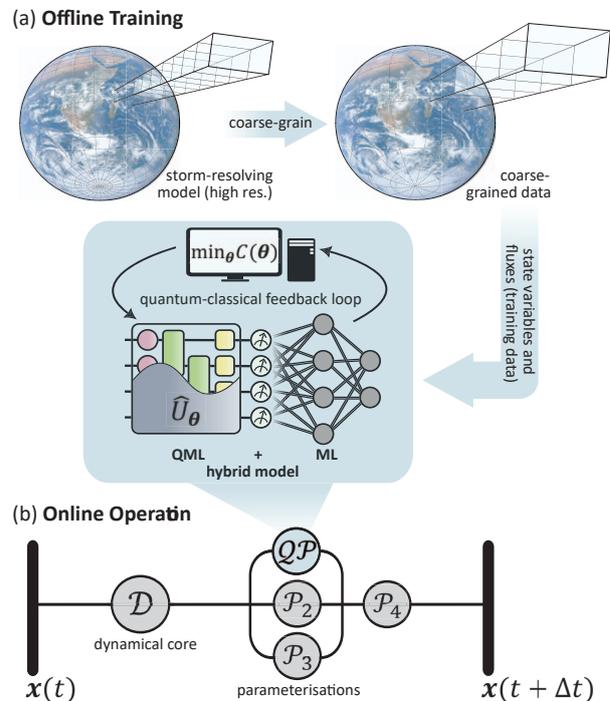


Fig. 1. Hybrid quantum-classical approach for QML-based parameterisations, analogous to the approach typically taken for developing classical ML-based parameterisations. (a) Training data are generated with short high-resolution (storm-resolving) climate model runs, or with Earth observations. Next, these data are coarse-grained to the resolution of the target climate model, and the target output variables are calculated, for instance the flux due to convection. The data are then used for offline training of the QML model, typically in a hybrid fashion together with a classical computer. (b) The resulting QML model of a given process replaces a conventional parameterization and is run at every time step in the coarse climate model after the dynamical core that solves equations describing the resolved dynamics. From [1], CC BY license.

ML-based models [6]. The ML models can be trained on short high-resolution data or Earth observations, or combinations thereof (e.g., first training on global high-resolution simulations and then refining the model with observations (e.g., [7])). Another possibility is to replace the entire model with an ML-based emulator (e.g., [8]). Both approaches, however, have difficulties when generalizing beyond the training regimes, for instance, in a changing climate where the mean and extremes

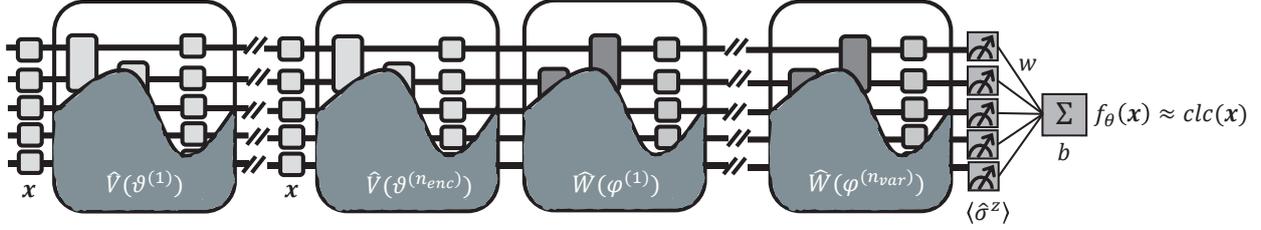


Fig. 2. Sketch of the QNN architecture. The number of used qubits (here shown schematically as 5) is equal to the number of input variables used, either 8 or 6. The data  $\mathbf{x}$  are uploaded  $n_{\text{enc}}$  times as angles of single-qubit rotations, each variable to a specific qubit. These re-uploading gates are interleaved with variational blocks  $\hat{V}(\vartheta^{(k)})$  containing entangling gates and trainable parameters  $\vartheta^{(k)}$ , ( $k = 1, \dots, n_{\text{enc}}$ ), then, another sequence of  $n_{\text{var}}$  variational blocks  $\hat{W}(\varphi^{(\ell)})$ , ( $\ell = 1, \dots, n_{\text{var}}$ ) are applied. Finally, the expectation values of  $\hat{\sigma}^z$  on all qubits are measured, and a weighted average with trainable weights  $w$  and a bias term  $b$  is calculated. During training, the parameters  $\theta = \{\{\vartheta^{(k)}\}_k, \{\varphi^{(\ell)}\}_\ell, \mathbf{w}, b\}$  are adjusted such that the result  $f_\theta(\mathbf{x})$  approximates the cloud cover  $\text{clc}(\mathbf{x})$ . Following [2].

of climate variable distributions are shifting [9].

A hierarchy of climate models will continue to be required, ideally comprising hybrid models [5]. Quantum computing offers additional potential to improve and accelerate climate models. Potential application areas range from accelerating the resolution of the underlying fundamental equations to tuning the climate models (i.e., finding the best set of free parameters to match observations), to using QML approaches to replace traditional parameterizations, analogous to classical ML approaches. Several works have demonstrated good generalization capabilities of quantum neural networks [10]–[13], higher expressivity [14], [15], and potential good trainability [16]. Here, we explore using QML for climate modelling, in particular for a cloud cover parameterization for a climate model.

In the typical training process for a data-driven parameterization based on high-resolution data, the training data are coarse-grained to the resolution of the target climate model (Fig. 1). This makes it possible to use the coarse-grained state variables as input to the QML model during training, serving as a proxy for the state variables in the coarse climate model. The output during the training represents the subgrid-scale process that the target parameterization models, e.g., fluxes due to unresolved convection, or the cloud cover in a given climate model cell. The data-driven parameterization thus learns these unresolved processes as a function of the coarse state of the Earth system in the given cell. After training, the QML model is then coupled to the climate model and run at every time step together with the other parameterizations after the dynamical core (representing the resolved processes).

Bazgir and Zhang [17] recently developed QML models to represent target values such as snow and rain rates, based on the ClimSim dataset [18]. This dataset is produced with a multiscale modeling framework (MMF) that embeds a kilometer-resolution cloud-resolving model within each atmospheric column of a coarser host climate model to replace traditional convection and cloud parameterizations [19]. Bazgir and Zhang [17] trained various QML models to represent the high-resolution embedded output as a function of the coarse state. They tested a quantum convolutional neural network,

a quantum multilayer perceptron, and a quantum encoder-decoder, and compared them to their classical counterparts. They found that the quantum models typically outperformed the classical ones.

In our work, we investigate a quantum neural network for the representation of cloud cover in a climate model.

## II. METHODS

### A. QML and ML models of cloud cover

We used high-resolution climate model data as training data [2], in contrast to training on the MMF as above. This has the advantage that we were able to use the same climate model to generate the training data as the target model (ICON [20]), and the data are more realistic as no scale-separation is assumed a priori [21], but the training data are less obvious to prepare. In this work, global high-resolution (2.5 km in the horizontal) simulations with the ICON climate model from the DYAMOND project [3] are used. As in [22], in the high-resolution data, a grid cell is considered to be fully cloudy when specific cloud condensate (water or ice) content exceeds a threshold, and cloud-free otherwise. The data are then coarse-grained to a horizontal resolution of 80 km [23]. The cloud cover after coarse-graining is between 0 and 1 and represents the cloudy fraction of the cell. The QML models are trained to learn the cloud fraction as a function of the coarse state variables (input features for the QML model) specific humidity  $q_v$ , specific cloud water content  $q_c$ , specific cloud ice content  $q_i$ , air temperature  $T$ , pressure  $p$ , and magnitude of horizontal wind  $h_w$ , as well as geometric height  $z_g$  and latitude  $\phi$ , or a subsection thereof (excluding height and latitude). A parameterized quantum circuit is used as QML model, with the number of qubits equal to the number of input features (Fig. 2). The qubit register is initialized in the  $|0\rangle$  state. After suitable rescaling, the input features  $\mathbf{x}$  are encoded as angles of single-qubit rotations in multiple encoding layers  $\hat{S}(\mathbf{x})$

$$\hat{S}(\mathbf{x}) = \prod_{n=1}^N \exp -i \frac{x_n}{2} \hat{\sigma}_n^x \equiv \hat{R}_x(\mathbf{x}) \quad (1)$$

with  $x_n$  denoting the  $n$ th component of the input  $\mathbf{x}$  with  $N$  features, and  $x$  the rotation axis.

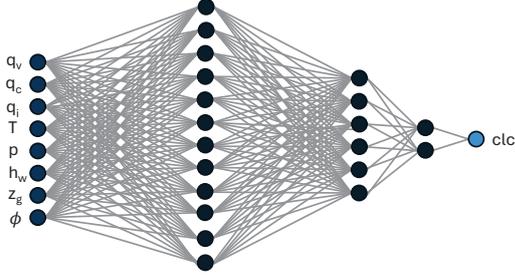


Fig. 3. Architecture of the classical neural net with eight input features used for comparison. After the input layer there are three hidden layers with 12, 6, 2 nodes, followed by a single node for output of the predicted cloud cover.

Data reuploading [24], [25] is used to increase the number of Fourier frequencies that the model is able to capture [25], [26]. These encoding layers are interleaved with variational blocks  $\hat{V}(\vartheta^{(k)})$  with the index  $k$  and trainable parameters  $\vartheta^{(k)}$ . The  $\hat{V}(\vartheta^{(k)})$  also contain entangling operations. We tested various circuit architectures [2]. For the results presented in this paper (in blue in Figs. 4, 5, 6)

$$\begin{aligned} \hat{V}(\vartheta) = & \hat{R}_{yy}(\vartheta_{(2N-1) \rightarrow (3N-3)}) \times \\ & \hat{R}_{xx}(\vartheta_{N \rightarrow (2N-2)}) \times \\ & \hat{R}_{zz}(\vartheta_{1 \rightarrow (N-1)}) \end{aligned} \quad (2)$$

with  $\hat{R}_{\alpha\alpha}(\vartheta) = \prod_{n=1}^{N-1} \exp -i \frac{\vartheta_n}{2} \hat{\sigma}_n^\alpha \hat{\sigma}_{n+1}^\alpha$ ,  $\alpha = x, y, z$ , and  $\vartheta_{i \rightarrow j}$  the slice of  $\vartheta$  from the  $i$ th to the  $j$ th component.

After  $n_{\text{enc}}$  applications of the blocks  $\hat{S}(\mathbf{x})$  and  $\hat{V}(\vartheta^{(k)})$ , another  $n_{\text{var}}$  variational blocks  $\hat{W}(\varphi^{(\ell)})$  with index  $\ell$  are applied to introduce additional trainable parameters  $\varphi^{(\ell)}$  and entangling operations. For the results in blue in Figs. 4, 5, 6

$$\begin{aligned} \hat{W}(\varphi) = & \hat{R}_x(\varphi_{(3N-2) \rightarrow (4N-3)}) \times \\ & \hat{R}_{yy}(\varphi_{(2N-1) \rightarrow (3N-3)}) \times \\ & \hat{R}_{xx}(\varphi_{N \rightarrow (2N-2)}) \hat{R}_{zz}(\varphi_{1 \rightarrow (N-1)}). \end{aligned} \quad (3)$$

After the computation, the expectation values of all Pauli  $Z$  operators are measured on the output state, and their weighted average with trainable weights  $\mathbf{w}$  and bias  $b$  is computed and interpreted as the fractional cloud cover in the cell. The QNNs are numerically simulated in Python using the PennyLane library [27] and optimized with JAX [28]. For QNNs implemented on a physical quantum device, the parameters  $\theta$ , summarizing the trainable parameters  $\vartheta, \varphi, \mathbf{w}$  and  $b$ , are trained in a quantum-classical feedback loop [29], [30]: The QNN is run on the quantum computer for a given set of  $\theta$ , and a cost function is calculated. Depending on this cost function, a new set of  $\theta$  is proposed, and the QNN is run in the next iteration with the new parameters. In our simulated training, we use the mean squared error (MSE) over the training dataset as cost function and update the parameters  $\theta$  using gradient descent. For QNNs the parameter-shift rule [31], [32] can be used to calculate the gradients of the cost function with respect to the parameters.

The networks are trained for 200 epochs with  $N_{\text{train}} = 2 \times 10^5$  training data. Each epoch consists of 1000 batch updates with a batch size of 100. The QML models have 200 resp. 201 trainable parameters when using the full set of eight input features, depending on the exact architecture choices. They are compared with a classical NN (Fig. 3) with 203 free parameters (using eight input features). As is visible in the different architecture layouts, the classical part of the QNN is minimal: the classical operations performed at the end of the QNN correspond to a simple weighted average of the QNN measurement outputs (with trainable weights plus a bias), i.e., to a one-node linear layer in the NN terminology. The fully-classical NN used in our work instead contains multiple layers consisting of several nodes, each including a non-linear activation function (either leaky-ReLU or tanh), which is indispensable to the operation of the NN. Therefore, we ensure that the quantum part of the QNN is essential in performing the calculations.

### B. Shapley values for explainability

Explainability methods for classical machine learning are already well established, even though advances in deep learning lead to a performance-explainability trade-off [33]. The application of explainability techniques to quantum machine learning is an active field of research [34]–[37].

In this paper, we apply a model-agnostic explainability method, namely Shapley values [38]. This method scores input features according to the effect of adding or removing them in the output of the model [36]. They are calculated for quantum and classical machine learning models using the SHapley Additive exPlanations (SHAP) library [39].

A SHAP value  $\text{shap}(x = x_0, y)$  gives the deviation in the output variable  $y$  for a specific value  $x_0$  of the variable  $x$  from the average prediction of  $y$  over a given dataset. It allows to determine the importance of the various input features for a specific output value and to interpret the relationships discovered by the data-driven model [40]. SHAP values offer both global and individual explanations of feature importance [35]. Specifically, in our work, we used KernelSHAP, a model-agnostic method that approximates Shapley values by treating the model as a black box and estimating contributions through perturbations of the input features [41]. We select 100 representative samples from the training dataset as background dataset to approximate feature attributions. Once the explainer is initialized, Shapley values are computed for the test dataset of 100,000 samples.

## III. RESULTS

The resulting QML model performs similarly well as a classical neural net (NN) with a similar number of free parameters, with the classical NN slightly outperforming the QNN. Both classical and quantum NN perform significantly better than the traditionally used Xu-Randall parameterization scheme [42] (Fig. 4).

The results presented so far were simulated without any noise. We also investigated the dependence of the results on

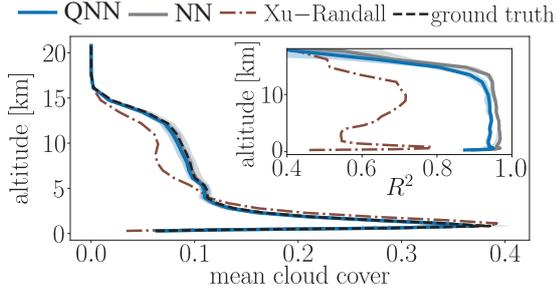


Fig. 4. Mean cloud cover as a function of altitude learned by a QNN model (blue solid line) and a classical NN with a comparable number of free parameters (grey solid line), compared with the ground truth (black dashed line) and the Xu-Randall scheme typically used in climate models (brown dash-dotted line). The shaded areas correspond to the spread over an ensemble of 20 training instances. Inset: Coefficients of determination  $R^2$  for the various models as a function of altitude. Adapted from [2].

shot noise. Training QNNs from the same initial parameter set, but with different shot noise realizations shows that training is successful and closely follows the noiseless training curve for a sufficiently large number of shots ( $n_{\text{shots}} > 10^4$ ) [2]. When a smaller number of  $n_{\text{shots}}$  is used, initially the mean squared error is minimized, but then the training becomes unstable. This happens when the magnitude of the gradients becomes comparable to the noise in the gradients' evaluation.

We also analysed the performance on a test dataset with a varying number of shots  $n_{\text{shot}}$  (Fig. 5). The performance, as measured by the coefficient of determination  $R^2$ , rapidly degrades for smaller  $n_{\text{shot}}$ , but is stable for  $n_{\text{shot}} > 10^4$ . Techniques such as variance regularization [43] can be used to minimize the number of shots needed, both during and after training [2]. These results demonstrate the usefulness of quantum computing for developing parameterizations in climate models.

Next, we investigate the learning capabilities of the QML model in comparison with the classical NN with SHAP values [41]. In the framework of explainability of ML models one of the first questions one may ask concerns the importance ranking of the input features in the model predictions, to see whether this ranking aligns with our physical expectations. In our case, in order to make a comparison between the quantum and classical models we use SHAP values since they are model agnostic, and their definition and computation does apply to both quantum and classical models.

In general, the analysis reveals that both classical and quantum NN place the highest importance on temperature and specific humidity, in accordance with our physical understanding. For a more thorough discussion of the specific rankings in relation with our physical expectations see [41]. Here, we explore whether the feature ranking is stable across different training experiments, which can be considered a proxy for the stability of the functional input-output relationships learnt by the model. We train two sets of ten networks starting from randomly initialized parameters. The resulting networks have

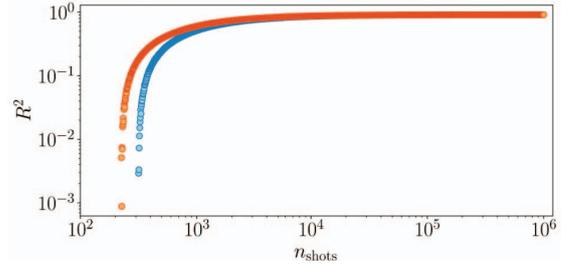


Fig. 5. Test set performance of two different architectures of QNNs for cloud cover using six input features (blue and orange, for details see [2]) as function of the number of shots  $n_{\text{shots}}$  used at inference. The QNNs were trained in the noiseless regime. The test set had a size of  $2 \times 10^5$ . Adapted from [2].

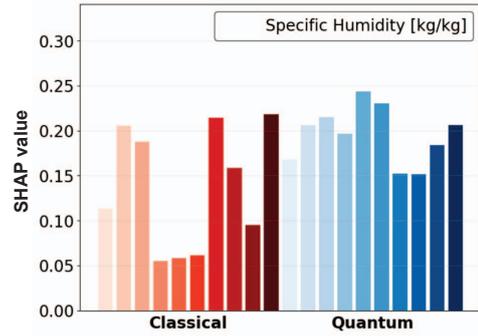


Fig. 6. Feature importances revealed in SHAP values of specific humidity for ten instances of classical (red, left) and quantum (blue, right) neural networks. Adapted from [41].

very similar performance ( $R^2$  between 0.92 and 0.94 for the classical NNs and between 0.89 and 0.91 for the quantum NNs).

Despite the comparable performances, a SHAP analysis reveals that feature importances for the classical neural networks differ vastly among each other, while the feature importances for the quantum NNs are more stable (see Fig. 6 for the feature importances found for specific humidity as an example, similar differences in variance of the feature importances are observed for other variables, especially temperature and pressure).

This means that, in this case, at least some well-performing classical neural networks learn potentially unphysical relationships – the feature importance of specific humidity should be consistently high, since it is the most important feature determining cloud cover. The training of the quantum neural networks in the analysed example is more robust and consistent.

#### IV. CONCLUSION AND OUTLOOK

In conclusion, we developed QML models to represent subgrid-scale cloud cover in a climate model. We showed that the QML models perform almost as well as classical neural nets with a similar number of trainable parameters and significantly better than the conventional parameterization used in climate models. We showed that, even taking into

account shot noise, the QML models arrive at good predictions with a reasonable number of shots. We also investigated the robustness of training both the classical and the quantum models and found in limited tests that the quantum models arrive at more stable feature importances than classical models. Whether these results hold for further examples, and what the influence of the different architectures and of training procedures is, is subject of future work.

As next step, the methods presented here could be extended to larger problem sizes, especially moving away from the cell-based approach used here by considering also neighboring cells as in [23]. Extending the work to further architectures could also prove promising, for example to quantum convolutional neural networks [44] which do not suffer from problems with barren plateaus [45]. QML approaches can also be applied for more complex climate model parameterizations, such as turbulence in the boundary layer. Another interesting application would be to learn the underlying statistics in the parameterization, for instance as introduced with coarse-graining. For this, the obvious choice would be quantum generative models [46]–[49].

These models can be developed already on current noisy intermediate-scale quantum (NISQ) devices. However, running the QML-based parameterizations coupled to the climate model requires considerable quantum and classical runtime with overheads due to the quantum-classical coupling [1]. While work on this coupling is ongoing, surrogates or shadows of QML models [50]–[53] could be used instead of running the QML models on real quantum hardware – classical models emulating the outputs of previously trained QML models, retaining some of the potential benefits while requiring a quantum device only during the training stage.

All these research efforts still face numerous challenges, but they also offer exciting potential to significantly boost the development of climate model parameterizations, in addition to current efforts on hybrid ML-enhanced ESMs [5] and high resolution modelling [54].

## REFERENCES

- [1] M. Schwabe et al., “Opportunities and challenges of quantum computing for climate modeling,” *Env. Data Sci.*, vol. 4, p. e35, 2025.
- [2] L. Pastori, A. Grundner, V. Eyring, and M. Schwabe, “Quantum neural network based cloud cover parameterization for climate models,” *arXiv:2502.10131*, 2025.
- [3] B. Stevens et al., “DYAMOND: the DYNAMICS of the atmospheric general circulation modeled on non-hydrostatic domains,” *Prog. Earth Planet. Sci.*, vol. 6, no. 1, p. 61, 09 2019.
- [4] C. Kodama et al., “The Nonhydrostatic ICosahedral Atmospheric Model for CMIP6 HighResMIP simulations (NICAM16-S): experimental design, model description, and impacts of model updates,” *Geosci. Model Dev.*, vol. 14, no. 2, pp. 795–820, Feb. 2021.
- [5] V. Eyring, P. Gentine, G. Camps-Valls, D. M. Lawrence, and M. Reichstein, “AI-empowered next-generation multiscale climate modeling for mitigation and adaptation,” *Nat. Geosci.*, vol. 17, no. 10, pp. 963–971, 09 2024.
- [6] A. Bracco, J. Brajard, H. A. Dijkstra, P. Hassanzadeh, C. Lessig, and C. Monteolani, “Machine learning for the physics of climate,” *Nat. Rev. Phys.*, Nov. 2024.
- [7] A. Grundner, T. Beucler, P. Gentine, and V. Eyring, “Data-driven equation discovery of a cloud cover parameterization,” *JAMES*, vol. 16, no. 3, Feb. 2024.
- [8] C. Bodnar et al., “A foundation model for the Earth system,” *arXiv:2405.13063*, 2024.
- [9] P. Gentine, V. Eyring, and T. Beucler, “Deep learning for the parametrization of subgrid processes in climate models,” in *Deep Learning for the Earth Sciences, Second Edition*, G. Camps-Valls, D. Tuia, X. X. Zhu, and M. Reichstein, Eds. Wiley, 2021.
- [10] L. Banchi, J. Pereira, and S. Pirandola, “Generalization in quantum machine learning: A quantum information standpoint,” *PRX Quantum*, vol. 2, no. 4, p. 040321, Nov. 2021.
- [11] M. C. Caro, E. Gil-Fuster, J. J. Meyer, J. Eisert, and R. Sweke, “Encoding-dependent generalization bounds for parametrized quantum circuits,” *Quantum*, vol. 5, p. 582, Nov. 2021.
- [12] M. C. Caro, H.-Y. Huang, M. Cerezo, K. Sharma, A. Sornborger, L. Cincio, and P. J. Coles, “Generalization in quantum machine learning from few training data,” *Nat. Commun.*, vol. 13, no. 1, p. 4919, 08 2022.
- [13] T. Haug and M. Kim, “Generalization of quantum machine learning models using quantum Fisher information metric,” *Phys. Rev. Lett.*, vol. 133, no. 5, p. 050603, Jul. 2024.
- [14] Y. Du, M.-H. Hsieh, T. Liu, and D. Tao, “Expressive power of parametrized quantum circuits,” *Phys. Rev. Res.*, vol. 2, p. 033125, 07 2020.
- [15] Z. Yu et al., “Provable advantage of parameterized quantum circuit in function approximation,” *arXiv:2310.07528*, 2023.
- [16] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, “The power of quantum neural networks,” *Nature Comp. Sci.*, vol. 1, pp. 403–409, 2021.
- [17] A. Bazgir and Y. Zhang, “QESM: A leap towards quantum-enhanced ML emulation framework for Earth and climate modeling,” *arXiv:2410.01551*, p. 2410.01551, 2024.
- [18] S. Yu et al., “Climsim: A large multi-scale dataset for hybrid physics-ml climate emulation,” in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 22 070–22 084.
- [19] Z. Hu et al., “Stable machine-learning parameterization of subgrid processes in a comprehensive atmospheric model learned from embedded convection-permitting simulations,” *arXiv:2407.00124*, 2024.
- [20] W. A. Müller et al., “ICON: Towards vertically integrated model configurations for numerical weather prediction, climate predictions and projections,” *BAMS*, vol. 106, no. 6, p. E1017–E1031, 2025.
- [21] T. Beucler et al., “Climate-invariant machine learning,” *Science Advances*, vol. 10, no. 6, p. ead7250, Feb. 2024.
- [22] M. A. Giorgetta et al., “ICON-A, the atmosphere component of the ICON Earth System Model: I. Model description,” *JAMES*, vol. 10, no. 7, pp. 1613–1637, Jul. 2018.
- [23] A. Grundner, T. Beucler, P. Gentine, F. Iglesias-Suarez, M. A. Giorgetta, and V. Eyring, “Deep learning based cloud cover parameterization for ICON,” *JAMES*, vol. 14, no. 2, p. e2021MS002959, 12 2022.
- [24] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre, “Data re-uploading for a universal quantum classifier,” *Quantum*, vol. 4, p. 226, Feb. 2020.
- [25] M. Schuld, R. Sweke, and J. J. Meyer, “Effect of data encoding on the expressive power of variational quantum-machine-learning models,” *Phys. Rev. A*, vol. 103, p. 032430, 03 2021.
- [26] B. Casas and A. Cervera-Lierta, “Multidimensional Fourier series with quantum circuits,” *Phys. Rev. A*, vol. 107, no. 6, p. 062612, Jun. 2023.
- [27] V. Bergholm et al., “Pennylane: Automatic differentiation of hybrid quantum-classical computations,” *arXiv:1811.04968*, 2018.
- [28] J. Bradbury et al., “JAX: composable transformations of Python+NumPy programs,” 2018. [Online]. Available: <http://github.com/google/jax>
- [29] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, “The theory of variational hybrid quantum-classical algorithms,” *New J. Phys.*, vol. 18, no. 2, p. 023023, 02 2016.
- [30] M. Cerezo et al., “Variational quantum algorithms,” *Nat. Rev. Phys.*, vol. 3, pp. 625–644, 2021.
- [31] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, “Quantum circuit learning,” *Phys. Rev. A*, vol. 98, p. 032309, 09 2018.
- [32] M. Schuld and N. Killoran, “Quantum machine learning in feature Hilbert spaces,” *Phys. Rev. Lett.*, vol. 122, p. 040504, 02 2019.
- [33] A. Barredo Arrieta et al., “Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI,” *Information Fusion*, vol. 58, pp. 82–115, Jun. 2020.
- [34] P. Steinmüller, T. Schulz, F. Graf, and D. Herr, “EXplainable AI for quantum machine learning,” *arXiv:2211.01441*, 2022.

- [35] L. Power and K. Guha, "Feature importance and explainability in quantum machine learning," *arXiv:2405.08917*, 2024.
- [36] E. Gil-Fuster, J. R. Naujoks, G. Montavon, T. Wiegand, W. Samek, and J. Eisert, "Opportunities and limitations of explaining quantum machine learning," *arXiv:2412.14753*, 2024.
- [37] R. Heese, T. Gerlach, S. Mücke, S. Müller, M. Jakobs, and N. Piatkowski, "Explaining quantum circuits with Shapley values: towards explainable quantum machine learning," *Quantum Mach. Intell.*, vol. 7, no. 1, p. 27, Feb. 2025.
- [38] L. S. Shapley, *Contributions to the Theory of Games (AM-28), Volume II*. Princeton University Press, 1953, ch. A value for n-person games, p. 307–317.
- [39] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *31st conference on neural information processing systems*, 2017.
- [40] H. Heuer, M. Schwabe, P. Gentine, M. A. Giorgetta, and V. Eyring, "Interpretable multiscale machine learning-based parameterizations of convection for ICON," *JAMES*, vol. 16, no. 8, Aug. 2024.
- [41] V. Sarandrea, "Explainable quantum machine learning for cloud cover parametrization," Master's thesis, Ludwig-Maximilians-Universität, Munich, 2025. [Online]. Available: <https://elib.dlr.de/213916/>
- [42] K.-M. Xu and D. A. Randall, "A semiempirical cloudiness parameterization for use in climate models," *J. Atmos. Sci.*, vol. 53, no. 21, pp. 3084–3102, Nov. 1996.
- [43] D. A. Kreplin and M. Roth, "Reduction of finite sampling noise in quantum neural networks," *Quantum*, vol. 8, p. 1385, Jun. 2024.
- [44] I. Cong, S. Choi, and M. D. Lukin, "Quantum convolutional neural networks," *Nat. Phys.*, vol. 15, pp. 1273–1278, 2019.
- [45] A. Pesah, M. Cerezo, S. Wang, T. Volkoff, A. T. Sornborger, and P. J. Coles, "Absence of barren plateaus in quantum convolutional neural networks," *Phys. Rev. X*, vol. 11, p. 041011, 10 2021.
- [46] M. H. Amin, E. Andriyash, J. Rolfe, B. Kulchytskyy, and R. Melko, "Quantum Boltzmann machine," *Phys. Rev. X*, vol. 8, p. 021050, 05 2018.
- [47] P.-L. Dallaire-Demers and N. Killoran, "Quantum generative adversarial networks," *Phys. Rev. A*, vol. 98, p. 012324, 07 2018.
- [48] B. Coyle, D. Mills, V. Danos, and E. Kashefi, "The Born supremacy: quantum advantage and training of an Ising Born machine," *npj Quantum Inf.*, vol. 6, p. 60, 2020.
- [49] X. Gao, E. R. Anschuetz, S.-T. Wang, J. I. Cirac, and M. D. Lukin, "Enhancing generative models via quantum correlations," *Phys. Rev. X*, vol. 12, no. 2, p. 021037, May 2022.
- [50] J. Landman, S. Thabet, C. Dalyac, H. Mhiri, and E. Kashefi, "Classically approximating variational quantum machine learning with random Fourier features," *arXiv:2210.13200*, 2022.
- [51] F. J. Schreiber, J. Eisert, and J. J. Meyer, "Classical surrogates for quantum learning models," *Phys. Rev. Lett.*, vol. 131, p. 100803, 09 2023.
- [52] S. Jerbi, C. Gyurik, S. C. Marshall, R. Molteni, and V. Dunjko, "Shadows of quantum machine learning," *Nat. Comm.*, vol. 15, p. 5676, 2024.
- [53] R. Sweke et al., "Potential and limitations of random Fourier features for dequantizing quantum machine learning," *Quantum*, vol. 9, p. 1640, Feb. 2025.
- [54] B. Stevens et al., "Earth virtualization engines (EVE)," *Earth Sys. Sci. Data*, vol. 16, no. 4, pp. 2113–2122, Apr. 2024.