

# Application of Quantum Machine Learning to High Energy Physics Analysis at LHC using IBM Quantum Computer Simulators and IBM Quantum Computer Hardware

J. Chan<sup>a</sup>, W. Guan<sup>a</sup>, S. J. Sun<sup>a</sup>, A. Wang<sup>a</sup>, S. L. Wu<sup>a</sup>, C. Zhou<sup>\*a</sup>, M. Livny<sup>b</sup>, F. Carminati<sup>c</sup>, A. D. Meglio<sup>c</sup>, A. C.Y. Li<sup>d</sup>, J. Lykken<sup>d</sup>, P. Spentzouris<sup>d</sup>, S. Y.-C. Chen<sup>e</sup>, S. Yoo<sup>e</sup>, T.-C. Wei<sup>f</sup>

<sup>a</sup>Physics Department of University of Wisconsin-Madison, US<sup>†</sup>

<sup>b</sup>Computer Sciences Department of University of Wisconsin-Madison, US

<sup>c</sup>CERN openlab, IT Department, CERN, CH

<sup>d</sup>Quantum Institute, Fermilab, US

<sup>e</sup>Computational Science Initiative, BNL, US

<sup>f</sup>C.N. Yang Institute for Theoretical Physics, State University of New York at Stony Brook, US

E-mail: [chen.zhou@cern.ch](mailto:chen.zhou@cern.ch)

One of the major objectives of the experimental programs at the LHC is the discovery of new physics. This requires the identification of rare signals in immense backgrounds. Using machine learning algorithms greatly enhances our ability to achieve this objective. With the progress of quantum technologies, quantum machine learning could become a powerful tool for data analysis in high energy physics. In this study, using IBM gate-model quantum computing systems, we employ the quantum variational classifier method and the quantum kernel estimator method in two recent LHC flagship physics analyses:  $t\bar{t}H$  (Higgs boson production in association with a top quark pair) and  $H \rightarrow \mu^+\mu^-$  (Higgs boson decays to two muons). We have obtained early results with 10 qubits on the IBM quantum simulator and the IBM quantum hardware. On the quantum simulator, the quantum machine learning methods perform similarly to classical algorithms such as SVM (support vector machine) and BDT (boosted decision tree), which are often employed in LHC physics analyses. On the quantum hardware, the quantum machine learning methods have shown promising discrimination power, comparable to that on the quantum simulator. This study demonstrates that quantum machine learning has the ability to differentiate between signal and background in realistic physics datasets.

40th International Conference on High Energy Physics - ICHEP2020 -  
28 July-6 August, 2020  
Prague, Czech Republic

\*Speaker.

<sup>†</sup>Partially supported by DOE DE-SC0020416

## 1. Introduction

The discovery of new physics at the Large Hadron Collider (LHC) requires the identification of rare signals against immense backgrounds. Using machine learning greatly enhances our ability to achieve this objective. The intersection between machine learning and quantum computing has been referred to as quantum machine learning, and can possibly offer a valuable alternative to classical machine learning by providing more efficient solutions. In 2018, a quantum variational classifier method and a quantum kernel estimator method were implemented with a quantum circuit of 2 qubits on a superconducting processor and successfully tested on synthetic datasets [1].

In this study, we employ the quantum variational classifier method and the quantum kernel estimator method in a  $t\bar{t}H$  ( $H \rightarrow 2$  photons) physics analysis and a  $H \rightarrow \mu^+\mu^-$  physics analysis, two recent flagship physics analyses at the LHC, using IBM gate-model quantum computers. Our goal is to explore and to demonstrate, as a proof of principle, that the potential of quantum computers can be a new computational paradigm for big data analysis in high energy physics.

## 2. Two recent LHC flagship physics analyses:

The observation of  $t\bar{t}H$  (Higgs boson production in association with a top quark pair) by the ATLAS and CMS experiments [2, 3] directly constituted the interaction between the Higgs boson and the top quark, the heaviest known fundamental particle. Furthermore, the  $H \rightarrow \mu^+\mu^-$  (Higgs boson decays to two muons) results of the ATLAS and CMS experiments [4, 5] indicated for the first time that the Higgs boson interacts with second-generation elementary particles.

Here we construct event classifiers for the two analyses. For  $t\bar{t}H$  production, we focus on a channel where the Higgs boson decays into two photons and the two top quarks decay into hadrons. For  $H \rightarrow \mu^+\mu^-$  decay, we focus on a channel where the Higgs boson is produced by the vector boson fusion (VBF) mode. In each analysis, the signal and background processes are generated using Madgraph5\_aMC@NLO [6] + Pythia6 [7], and simulated using Delphes [8]. Kinematic variables (such as transverse momentum of final state particles) are compressed to 10 training variables, using a Principal Component Analysis (PCA) method.

## 3. Quantum variational classifier algorithm

Following [1], we use the quantum variational classifier algorithm to classify physics events of interest from background events. It can be summarized in four main steps:

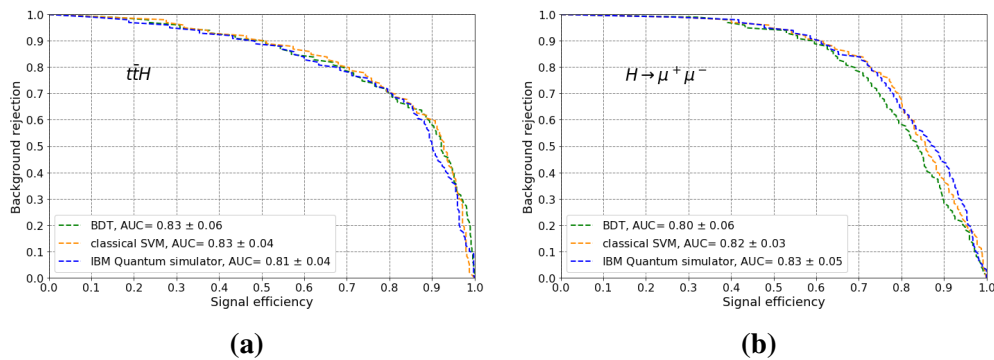
1. Apply a feature map circuit  $U_{\Phi(\vec{x})}$  to encode the input data  $\vec{x}$  into a quantum state  $|\Phi(\vec{x})\rangle$ .
2. Apply a quantum variational circuit  $W(\vec{\theta})$  parameterized by gate angles  $\vec{\theta}$ .
3. Measure the qubit state in the computational basis.
4. Classify the state (as "signal" or "background") through the action of a diagonal operator  $f$  in computational basis.

During the training phase, a set of input data  $\vec{x}$  and corresponding outputs  $y$  are used to train the circuit  $W(\vec{\theta})$  to reproduce the correct classification. The set of optimized parameters  $\vec{\theta}$  is then kept fixed for all future classifications of the physical data.

#### 4. Quantum variational classifier result from IBM Quantum Computer Simulator

We employ quantum variational classifier with 10 qubits on the *ibmq QasmSimulator* for the  $t\bar{t}H$  analysis and the  $H \rightarrow \mu^+\mu^-$  analysis. The *ibmq QasmSimulator* simulates executions and measurements on quantum circuits. The simulation incorporates a noise model generated from the properties of real hardware device. In each analysis, we apply the quantum variational classifier algorithm to ten independent datasets, each consisting of 100 events for training and 100 events for testing. The quantum circuits are optimized to best fit the constraints imposed by the hardware (e.g., qubit connectivity, gate set availability, and hardware noise), as well as the nature of the data. With the same ten datasets and the same 10 PCA variables, we also train a classical SVM classifier using the scikit-learn package [9] and a BDT classifier using the XGBoost package [10]. The classical SVM and the BDT serve as benchmarks for classical machine learning algorithms.

To study the discrimination power of each algorithm for both  $t\bar{t}H$  and  $H \rightarrow \mu^+\mu^-$ , the testing events of the ten datasets are used to make Receiver Operating Characteristic (ROC) curves in the plane of background rejection versus signal efficiency, as shown in Figure 1. We observe that in both the  $t\bar{t}H$  analysis and the  $H \rightarrow \mu^+\mu^-$  analysis, the quantum variational classifier method on the *ibmq QasmSimulator* (blue) performs similarly to the classical SVM (yellow) and the BDT (green). We quantify the discrimination power of each classifier by the AUC (area under the ROC curve). In the  $t\bar{t}H$  analysis, the AUC for the quantum variational classifier method reaches 0.81, compared to 0.83 for the classical SVM and 0.83 for the BDT. Similarly, in the  $H \rightarrow \mu^+\mu^-$  analysis, the AUC for the quantum variational classifier method reaches 0.83, compared to 0.82 for the classical SVM and 0.80 for the BDT. This demonstrates the quantum algorithm can accurately distinguish signal from background on realistic physics datasets.



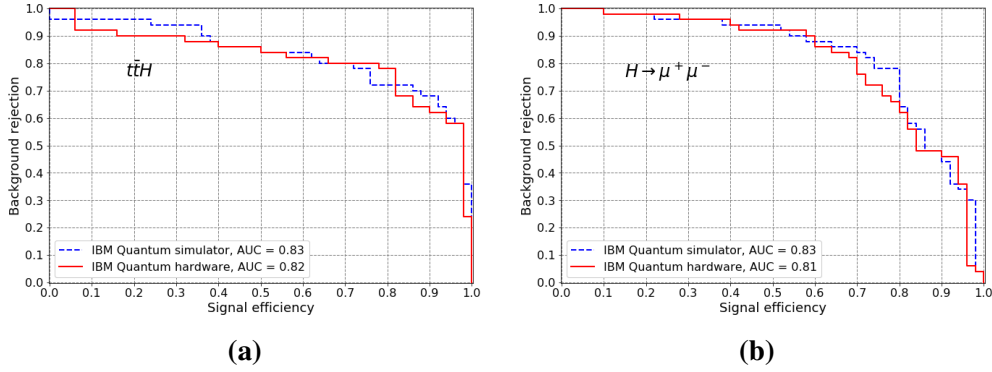
**Figure 1:** The Receiver Operating Characteristic (ROC) curves (in the plane of background rejection versus signal efficiency) of the quantum variational classifier method on the *ibmq QasmSimulator* (blue), the classical SVM (yellow), and the BDT (green) for (a) the  $t\bar{t}H$  analysis and (b) the  $H \rightarrow \mu^+\mu^-$  analysis.

#### 5. Quantum variational classifier result from IBM Quantum Computer Hardware

For the  $t\bar{t}H$  analysis and the  $H \rightarrow \mu^+\mu^-$  analysis, we employ the quantum variational classifier algorithm with 10 qubits on the “*ibmq\_boeblingen*” and “*ibmq\_paris*” quantum computer hardware. “*ibmq\_boeblingen*” is a 20-qubit quantum processor and “*ibmq\_paris*” is a 27-qubit

quantum processor. Both are based on superconducting electronic circuits. Due to current limitation of the available resources, the quantum variational classifier algorithm is only applied to one of the ten datasets for each physics analysis. The circuit configuration on the hardware is kept the same as for the simulator jobs.

The ROC curves of the quantum variational classifier method on the “*ibmq\_boeblingen*” quantum hardware (for  $t\bar{t}H$ ) and “*ibmq\_paris*” quantum hardware (for  $H \rightarrow \mu^+\mu^-$ ) are shown in red in Figure 3. The ROC curves for the *ibmq QasmSimulator* with the same datasets are overlaid in blue. We observe that for the quantum variational classifier method, the quantum simulator and quantum hardware results appear to be in good agreement. In the  $t\bar{t}H$  analysis, the quantum hardware AUC is 0.82, while the quantum simulator AUC is 0.83. Similarly, in the  $H \rightarrow \mu^+\mu^-$  analysis, the quantum hardware AUC is 0.81, while the quantum simulator AUC is 0.83. The gate-model quantum computers have achieved reasonable performance to distinguish signal from background.



**Figure 2:** The Receiver Operating Characteristic (ROC) curves of the quantum variational classifier method with the quantum computer hardware (red) and with the *ibmq QasmSimulator* (blue) for (a) the  $t\bar{t}H$  analysis (using “*ibmq\_boeblingen*”) and (b) the  $H \rightarrow \mu^+\mu^-$  analysis (using “*ibmq\_paris*”).

## 6. Quantum Kernel Estimator Algorithm

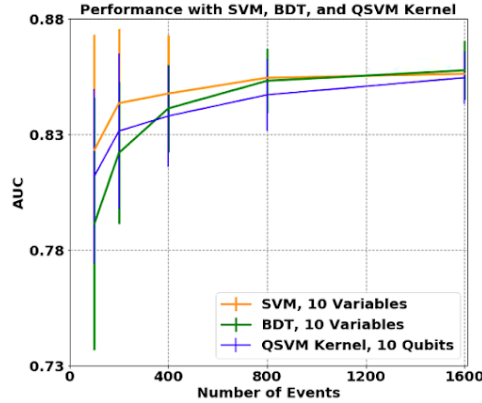
The quantum kernel estimator algorithm [1] maps classical data  $\vec{x}$  non-linearly to a quantum state  $|\Phi(\vec{x})\rangle$ , and calculates the kernel entry for data  $\vec{x}$  and  $\vec{z}$ , as  $|\langle\Phi(\vec{x})|\Phi(\vec{z})\rangle|^2 = |\langle 0^n | \mathcal{U}_{\Phi(\vec{x})}^\dagger | \mathcal{U}_{\Phi(\vec{z})} | 0^n \rangle|^2$ . The quantum feature map  $\mathcal{U}_{\Phi(\vec{x})}$  encodes classical data in its parameters. In the training phase, the kernel entries are evaluated for all pairs of the training sample, and used to find a separating hyper-plane with a classical SVM. In the classification phase, the kernel entries are evaluated for new data  $\vec{x}'$  and data from the training sample, and used to classify data  $\vec{x}'$  with the separating hyper-plane.

## 7. Quantum kernel estimator result from IBM Quantum Computer Simulator

For the  $t\bar{t}H$  analysis, we employ the quantum kernel estimator method using 10 qubits on the *ibmq Statevector Simulator*. The *ibmq Statevector Simulator* emulates the executions of a quantum circuit and evaluates the resulting quantum state vector. It does not incorporate noises of the hardware device. The quantum kernel estimator method enables us to work with a larger number

of events. With the same datasets and the same training variables, we construct a classical SVM classifier using the scikit-learn package [9] and a BDT classifier using the XGBoost package [10].

In Figure 3, we show the area under the ROC curve (AUC) for each algorithm versus different dataset sizes, for the  $t\bar{t}H$  analysis. All results are averaged over 30 runs. We find that, the quantum kernel estimator method and the classical methods exhibit comparable performances.



**Figure 3:** Area under the ROC curve (AUC) versus different dataset sizes for the quantum kernel estimator method (blue), the classical SVM (yellow) and the BDT (green), for the  $t\bar{t}H$  analysis.

## 8. Quantum kernel estimator result from IBM Quantum Computer Hardware:

We employ the quantum kernel estimator method using 10 qubits on the “*ibmq\_paris*” quantum computer hardware for the  $t\bar{t}H$  analysis. Due to current limitation of access time on “*ibmq\_paris*”, we process only one dataset consisting of 100 training events and 100 test events. The quantum circuits and SVM hyperparameters of the hardware run are kept the same as for the simulator runs. We observe that, for the quantum kernel method, the quantum computer simulator and quantum computer hardware results are in good agreement. In the  $t\bar{t}H$  analysis, the AUC from the quantum computer hardware reaches 0.85, compared to 0.84 from the quantum computer simulator.

## 9. Summary

In this study, we have obtained early results in the application of quantum machine learning (the quantum variational classifier method and the quantum kernel estimator method) with 10 qubits on the *ibmq QasmSimulator* and the “*ibmq\_boeblingen*” and “*ibmq\_paris*” quantum hardware to two recent LHC flagship physics analyses:  $t\bar{t}H$  and  $H \rightarrow \mu^+\mu^-$ . The quantum machine learning methods on the *ibmq QasmSimulator* perform similarly to the classical SVM algorithm and the BDT algorithm. The quantum machine learning methods on the quantum hardware have shown promising discrimination power comparable to that on the quantum simulation. These results demonstrate that quantum machine learning on the hardware of the gate-model quantum computers has the ability to differentiate between signal and background in a realistic high energy

physics analysis at the LHC. Furthermore, with the rapid advance of the quantum computing technology, the use of quantum machine learning may offer a “speed up” advantage, which can be critical for the future of the high energy physics community.

## 10. Acknowledgement

We would like to acknowledge and thank our collaborators in IBM Research Zurich and IBM T.J. Watson Research Center - P. Barkoutsos, I. Tavernelli, S. Woerner, and J. Glick - for their contribution to this proceeding. In particular, we are indebted to their extensive knowledge of the IBM Quantum simulator and IBM Quantum hardware.

## References

- [1] Havlíček, V., Córcoles, A. D., Temme, K., Harrow, A. W., Kandala, A., Chow, J. M. and Gambetta, J. M. Supervised learning with quantum-enhanced feature spaces. *Nature* **567**, 209-212 (2019).
- [2] ATLAS Collaboration, Aaboud, M. et al. Observation of Higgs boson production in association with a top quark pair at the LHC with the ATLAS detector. *Phys. Lett. B* **784**, 173-191 (2018).
- [3] CMS Collaboration, Sirunyan, A. M. et al. Observation of ttH Production. *Phys. Rev. Lett.* **120**, 231801 (2018).
- [4] ATLAS Collaboration, Aaboud, M. et al. A search for the dimuon decay of the Standard Model Higgs boson with the ATLAS detector. *arXiv:2007.07830*.
- [5] CMS Collaboration, Sirunyan, A. M. et al. Evidence for Higgs boson decay to a pair of muons. *arXiv:2009.04363*.
- [6] Alwall, J., Frederix, R., Frixione, S., Hirschi, V., Maltoni, F., Mattelaer, O., Shao, H. -S., Stelzer, T., Torrielli, P. and Zaro, M. The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations. *JHEP* **07**, 079 (2014).
- [7] Sjostrand, T., Mrenna, S. and Skands, P. Z. PYTHIA 6.4 physics and manual. *JHEP* **05**, 026 (2006).
- [8] de Favereau, J., Delaere, C., Demin, P., Giammanco, A., Lemaître, V., Mertens, A. and Selvaggi, M. DELPHES 3, a modular framework for fast simulation of a generic collider experiment. *JHEP* **02**, 057 (2014).
- [9] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E. Scikit-learn: machine learning in python. *Journal of Machine Learning Research* **12**, 2825-2830 (2011).
- [10] Chen, T. and Guestrin, C. XGBoost: a scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785-794 (2016).