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

Speaker: Shaojun Sun

Jay Chan, Wen Guan, Shaojun Sun, Alex Wang, Sau Lan Wu, Chen Zhou
Physics Department, University of Wisconsin-Madison
and
Miron Livny
Computer Sciences Department and Wisconsin Institute for Discovery, University of Wisconsin-Madison
and
Federico Carminati, Alberto Di Meglio
CERN Openlab, IT Department, CERN
and
Panagiotis Barkoutsos, Ivano Tavernelli, Stefan Woerner
IBM Research Zurich
and
Andy Li, Joseph Lykken, Panagiotis Spentzouris
Quantum Institute, Fermilab

January 22, 2019 - Geneva, Switzerland
CERN openlab Technical Workshop

Machine learning and quantum computing

- Machine Learning has become one of the most popular and powerful techniques and tools for HEP data analysis
- Machine Learning: This is the field that gives computers “the ability to learn without explicitly programming them”.
- Issues raised by machine learning
  - Heavy CPU time is needed to train complex models
    - With the size of more data, the training time increases very quickly
  - May lead to local optimization, instead of global optimization
- Quantum computing
  - A way of parallel execution of multiple processes using Qubits
  - Can speed up certain types of problems effectively
  - It is possible that quantum computing can find a different, and perhaps better, way to perform machine learning.

Ref: “Global Optimization Inspired by Quantum Physics”, 10.1007/978-3-642-38703-6_41

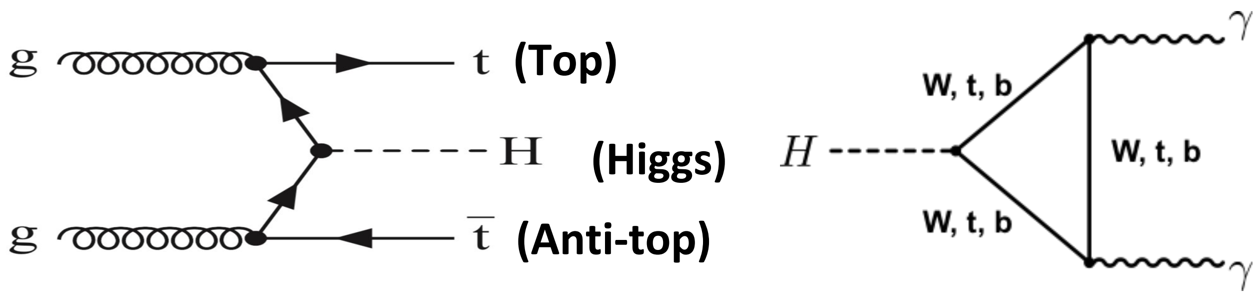
Our program with IBM Qiskit

Our Goal:

Perform LHC High Energy Physics analysis with Quantum computing

Our preliminary program is to:

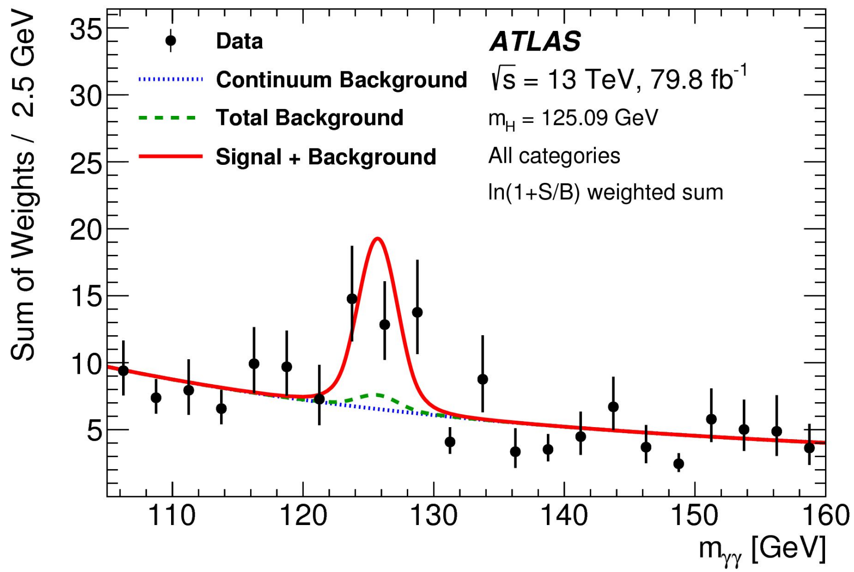
Employ the Quantum Support Vector Machine (QSVM) method for LHC High Energy Physics (HEP) analysis with the environment of IBM Qiskit, for example ttH (H $\rightarrow \gamma\gamma$), Higgs production in association with two top quarks analysis.



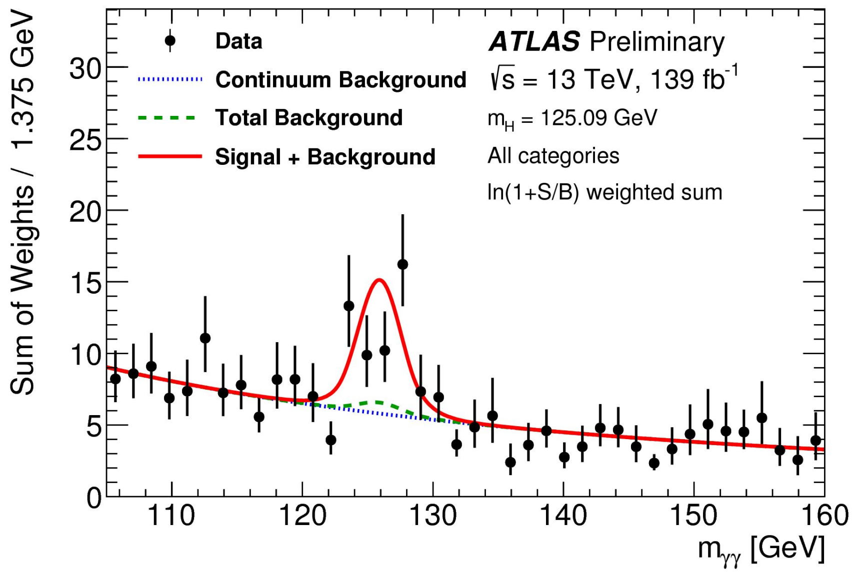
*** IBM Qiskit = IBM Quantum Information Science Kit**

ttH (H → $\gamma\gamma$) analysis by the ATLAS Collaboration

(ttH: Higgs production in association with two top quarks)



Phys. Lett. B 784 (2018) 173



ATLAS-CONF-2019-004

- By using Boosted Decision Tree (BDT, a classical machine learning technique) with XGBoost package, the ATLAS Collaboration observes the ttH (H→γγ) process
- This talk is to perform the machine learning step of the ATLAS ttH (H→γγ) analysis with delphes simulation events using quantum machine learning

Our program with IBM Qiskit

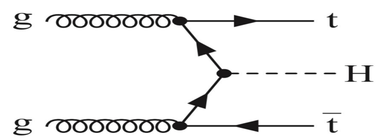
Our preliminary program can be divided into three parts with the Environment of IBM Qiskit:

Part 1. **Our workflow for quantum machine learning.**

Part 2. **Employing the quantum machine learning method for LHC High Energy Physics (HEP) analysis with quantum simulators, for example the IBM Qiskit qasm simulator.**

Part 3. **Employing the quantum machine learning method for LHC High Energy Physics (HEP) analysis with IBM quantum hardware, for example the IBM hardware (ibmq_boeblingen, a 20-qubit machine).**

Part 1: Our Workflow for Quantum Machine Learning



Start

Input Events

PCA
(convert features to less features based on available qubits)

Feature Map
(map classical features to large dimensional quantum features)

Quantum Machine Learning
(training and evaluating)

End

ttH: number of features(variables) = 45

- **PCA: Principal Component Analysis method is used to convert/combine features into fewer features to match the number of qubits.**
- **FeatureMap: encoding classical variables to quantum states. We have many more features for an event than available qubits (Number of qubits = 5, 10, 20 for example)**

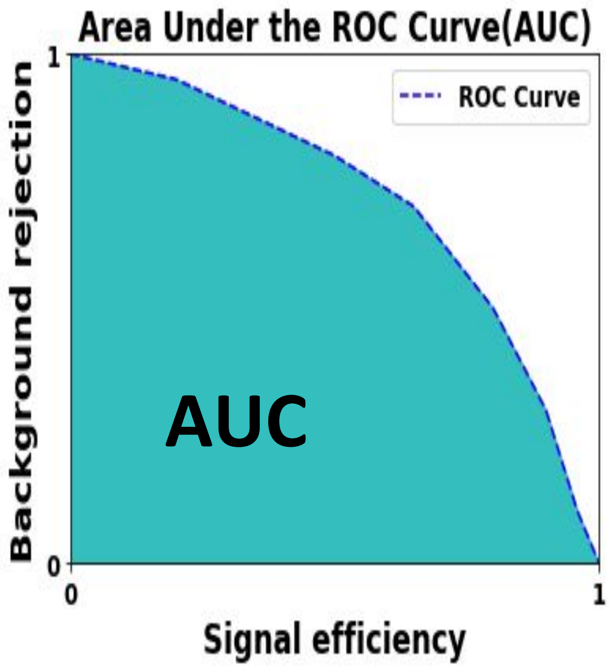
Support Vector Machine (SVM) quantum variational method, for example

Part 2: Employing QSVM Variational with Q simulators

- **Employing QSVM Variational for LHC HEP analyses**
  - **For example, a ttH ($H \rightarrow \gamma\gamma$), Higgs production in association with two top quarks analysis**
  - **Exploring different feature maps and entanglement methods**
  - **Training and evaluating quantum machine learning methods with different numbers of qubits, different numbers of events, different parameters and optimizers**

Part 2: Employing QSVM Variational with Q simulators

- **Definitions**
  - A **BDT**(Boosted Decision Tree) is a classical machine learning method. Here we are using XGBoost.
  - **Q simulator:** Quantum circuits simulator, such as Qasm simulator.
  - **Accuracy**: The ratio of correct predictions to total predictions.
  - **ROC Curve**: a graph showing background rejection vs signal efficiency.
  - **AUC**: Area Under the ROC Curve



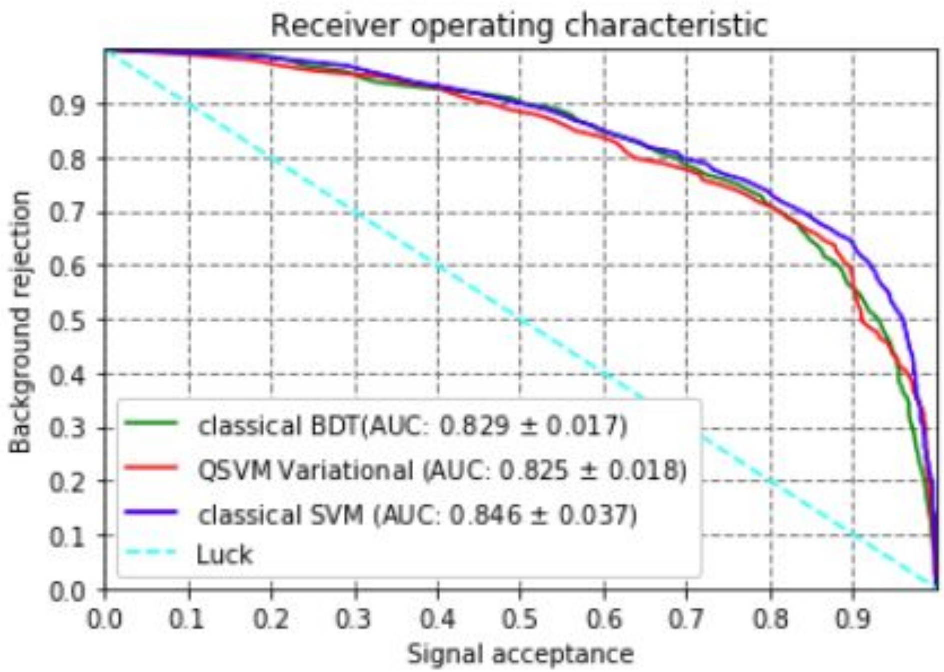
Ref: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

Part 2: Employing QSVM Variational with Q simulators

With 5 qubits, we successfully finished training and testing with 200 events with IBM Qiskit qasm simulator (where ‘200’ events means 200 training events and 200 test events. Events are simulated with Delphes).

- **For QSVM, COBYLA optimizer is used with 2000 iterations. Our simulation incorporates the hardware noise, although the qubit relaxation time (T1) and dephasing time (T2) are currently neglected**
- **Q simulator: Here Qiskit Qasm simulator is used.**

Part 2: Employing QSVM Variational with Q simulator



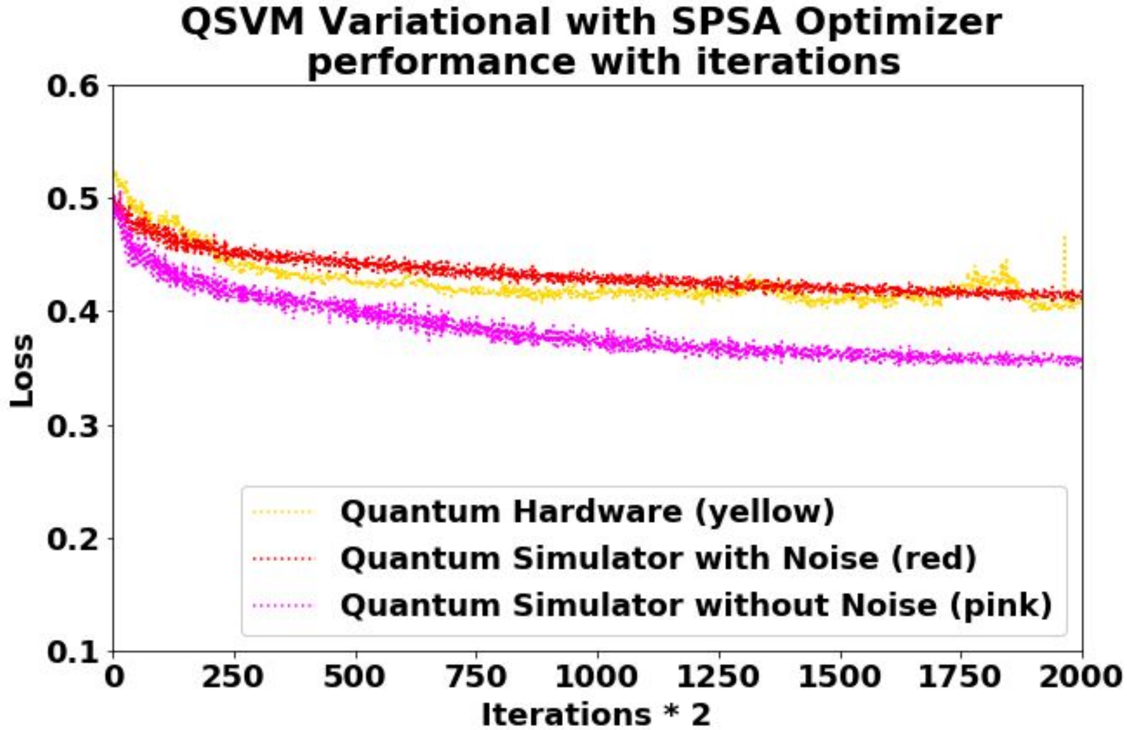
ttH(H->$\gamma\gamma$) AUC	AUC
Classical SVM	0.846 ± 0.037
Classical BDT	0.829 ± 0.017
QSVM Variational	0.825 ± 0.018

- Using ttH analysis dataset (200 events, 5 variables), we train classifiers with classical BDT, QSVM Variational on (5-qubit) quantum simulator, and classical SVM.
- QSVM variational on simulator (red) performs similarly with classical BDT (green) and classical SVM (blue).

Part 3: Employing QSVM Variational with IBM hardware (ibmq_boeblingen, a 20-qubit machine)

- **With the help of IBM Research Zurich and Fermilab, we have finished some jobs on the IBM hardware (ibmq_boeblingen) with 100 training events and 100 test events with 5 qubits.**

Part 3: Employing QSVM Variational with IBM hardware (ibmq_boeblingen, a 20-qubit machine)



PINK: Quantum Simulation without Noise

Red: Quantum Simulation with Noise

Yellow: Quantum Hardware

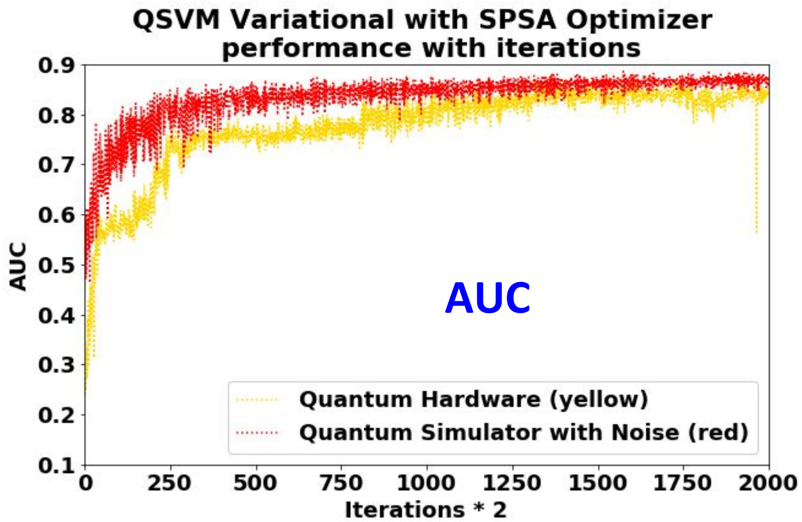
Loss: the mean of the squared differences between the output scores from the quantum algorithm and the ideal scores

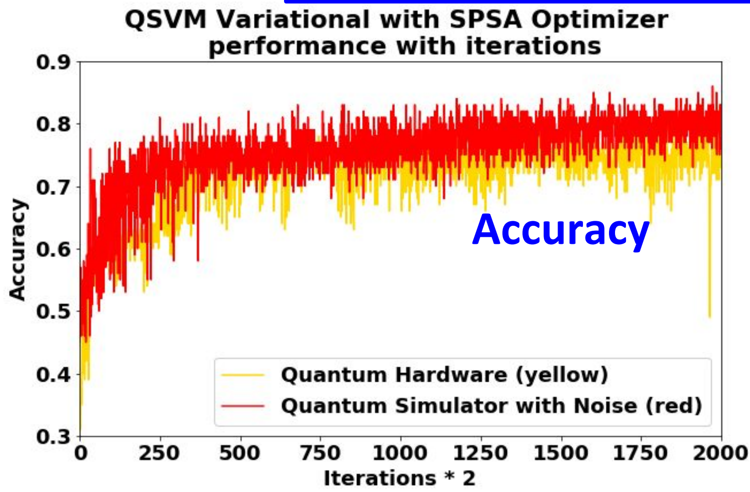
- We have finished 1000 iterations of training on the IBM hardware: ibmq_boeblingen. The hardware loss is decreasing with the increase of number of iterations. This indicates that the Quantum Computer has the ability to learn how to differentiate between the signal and the background for a high energy physics analysis.

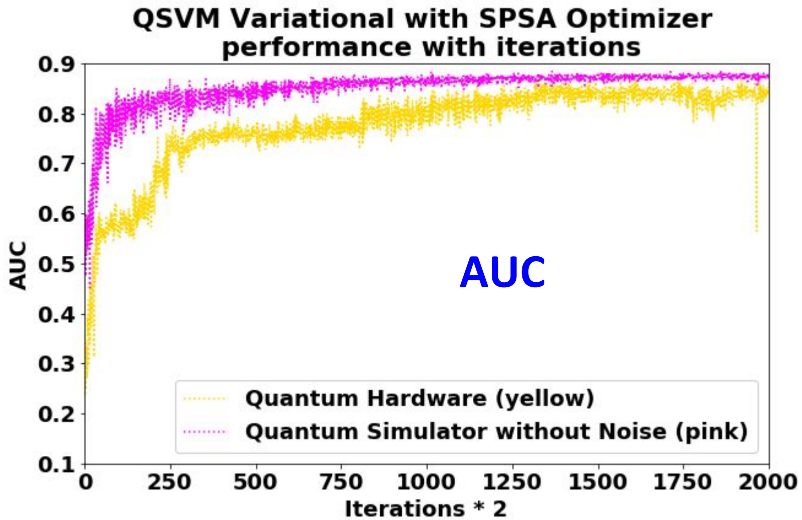
Note: the noise model for the noise simulation is from ibmq_boeblingen.

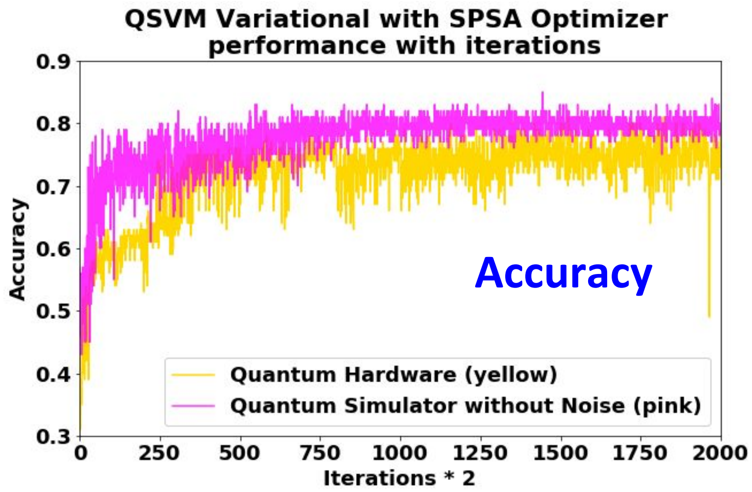
Part 3: Employing QSVM Variational with IBM hardware (ibmq_boeblingen, a 20-qubit machine)

PINK: Quantum Simulation without Noise
Red: Quantum Simulation with Noise;
Yellow: Quantum Hardware



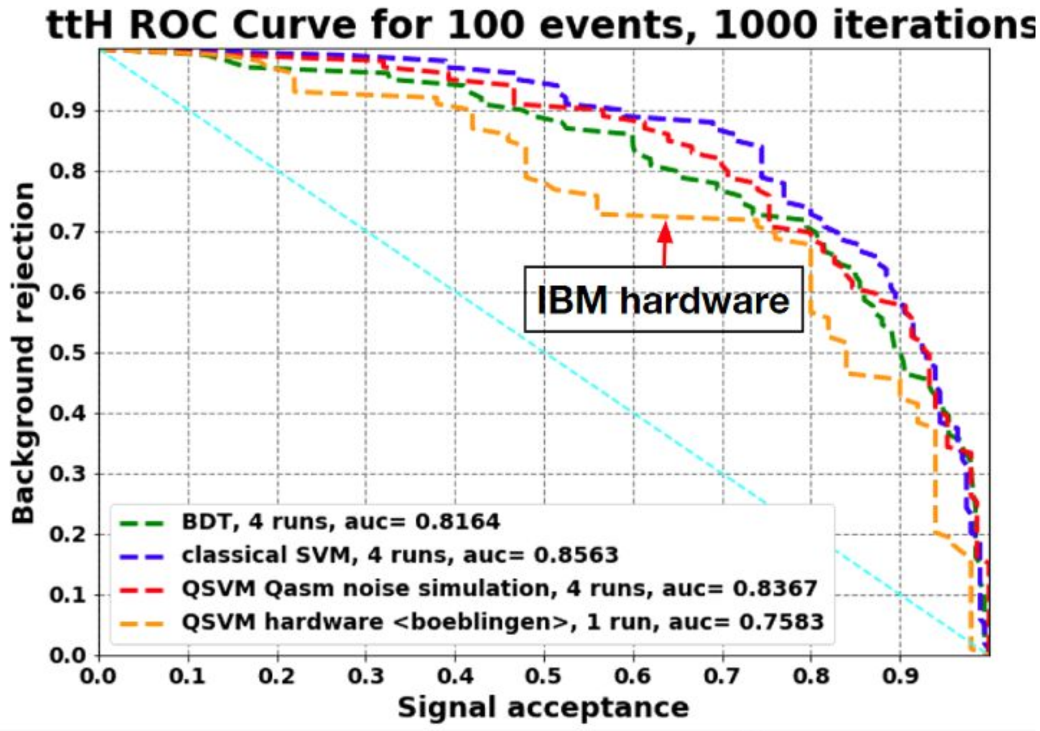






Both the hardware AUC and the hardware Accuracy are increasing with the increase of number of iterations. The performance is improving during the QSVM training process.

Part 3: Employing QSVM Variational with IBM hardware (ibmq_boeblingen, a 20-qubit machine)



ttH(H->$\gamma\gamma$) AUC	AUC
Classical SVM	0.856
XGBoost BDT	0.816
QSVM Simulation with Noise	0.837
QSVM Hardware	0.758

- Using ttH analysis dataset (100 events, 5 variables), we train classifiers with classical BDT, QSVM on (5-qubit) quantum simulator, classical SVM, QSVM on (5-qubit) quantum hardware.
- With 1000 iterations, the discrimination power of the QSVM on the Quantum Hardware is currently about 10% below that of the classical machine learning methods and the QSVM on quantum simulator.

Part 3: Employing QSVM Variational with IBM hardware (ibmq_boeblingen, a 20-qubit machine)

- **Temporary limitations with IBM hardware**
  - **Only one set of dataset is tested currently**
    - **Limited access time**
      - **Long queue time**
  - **Input preparation and output reading is not optimized**
  - **Circuit length and number of CNOT gates are limited**

Summary

Referring to Part 1 of this presentation:

- **We introduced our workflow to employ quantum machine learning methods for LHC High Energy Physics analyses.**

Summary

Referring to Part 2 of this presentation:

- **Using IBM Qiskit simulator, we have successfully employed the QSVM Variational method for a ttH (H $\rightarrow \gamma\gamma$), Higgs production in association with two top quarks analysis at the LHC with Delphes simulation events.**
- **With 200 events and 5 qubits, QSVM Variational perform similarly with classical BDT and classical SVM.**
- **At the same time, we are working on various ways (e.g. different optimizers, loss functions) to improve AUC.**

Summary

Referring to Part 3 of this presentation:

- **Using the IBM hardware (ibmq_boeblingen, a 20-qubit machine), we have successfully employed a Quantum Support Vector Machine method (5 qubits) for a ttH ($H \rightarrow \gamma\gamma$), Higgs production in association with two top quarks analysis at the LHC with Delphes simulation events.**

- **Using 100 events and 5 qubits, with 1000 optimization iterations, the discrimination power of the Quantum Support Vector Machine (QSVM) on the Quantum Hardware is currently about 10% below that of the classical machine learning methods and the QSVM on quantum simulator.**