The Joint Workshop of the CEPC Physics, Software and New Detector Concept

Application of quantum computing at Higgs measurements

Abdualazem Fadol, Zhao Yu, Ryuta Kiuchi, Fangyi Guo, Shuiting Xin
Yaquan Fang, Xin Shi, Xifeng Ruan

April 16, 2021

中國科學院高能物理研究所
Institute of High Energy Physics
Chinese Academy of Sciences

Institute for Collider Particle Physics
University of the Witwatersrand

Contents

- ☐ Machine learning has blossomed in the last decades and becomes essential in many fields.
- ☐ It played a significant role in solving High Energy physics problems, such as reconstruction, particle identification;
- ☐ and handling high dimensional and complex problems using deep learning.
- ☐ Quantum computing is a new idea for our workstations to process data faster than currently achievable.
- ☐ Machine learning & quantum computing may:
  - locating more computationally complex feature spaces
  - better data classification
  - smarter algorithms that can give us accurate prediction.
- ☐ Companies such as Google and IBM are committed to accelerating the development of quantum technology.

Objective

- ☐ Apply quantum machine learning in high energy physic.
- ☐ We compare quantum support-vector machine to classical support-vector machine.
- ☐ The comparison is demonstrated in terms of process from different CEPC and LHC:
  - CEPC: $e^+e^- \to Z(\to q\bar{q})H(\to \gamma\gamma)$ & $e^+e^- \to Z(\to \mu^-\mu+)H(\to \nu\nu qq)$
  - LHC: (VBF) $H \to \gamma\gamma$
- ☐ We use IBM quantum simulator " qasm_simulator "
- ☐ The simulator is build using Qiskit packages

Support-vector machines

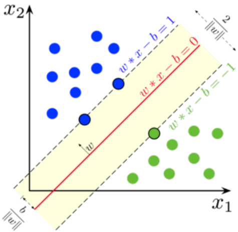
SVM

- ☐ SVMs are supervised machine learning algorithms for classifications.

$$(\vec{x}_i, y_i) \ldots (\vec{x}_n, y_n)$$

- ☐ $\vec{x}_i$ is n-dimensional vector and y_i is class label of each data point.

- ☐ SVM tries to maximize the margin between hyperplanes.
- ☐ Useful if the training dataset is linearly separable.
- ☐ It'll hard to separate non-linear datasets.
- ☐ So a trick called kernel machine is introduced.



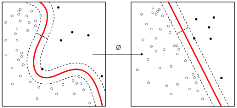
Support-vector machines

Kernel trick

- ☐ The dot product of a feature $\vec{x}_i$ and $\vec{x}_j$, after being transferred to a higher dimension via a function f, is called kernel.

$$k_{ij}(\vec{x}_i, \vec{x}_j) = \langle f(\vec{x}_i), f(\vec{x}_j) \rangle$$

- ☐ Non-leaner futures can then mapped to a liner ones.
- ☐ The function $f(\vec{x})$ could be:
  - linear
  - polynomial
  - Radial basis function
  - sigmoid



- ☐ In our case, we'll be using a linear function;
- ☐ and we call the SMV a classical SVM.

Quantum support-vector machines

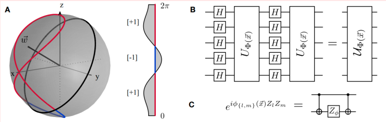
Quantum kernel

- ☐ In a quantum kernel, a classical feature $\vec{x}$ is mapped to higher dimension Hilbert space like $|\phi(\vec{x})\rangle\langle\phi(\vec{x})|$ in such a way that:

$$k_{ij}(\vec{x}_i, \vec{x}_j) = |\langle\phi(\vec{x}_i)|\phi(\vec{x}_j)\rangle|^2$$

- ☐ Feature map quantum circuits:
  - ZZFeatureMap
  - ZFeatureMap
  - PauliFeatureMap
- ☐ Many more in Qiskit packages.

arXiv:1804.11326v2

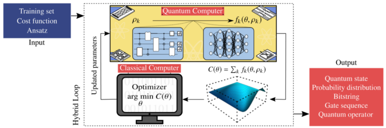


Variational quantum algorithm

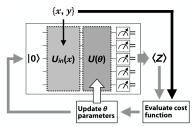
VQA

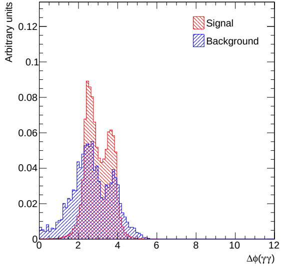
- ☐ VQA uses a classical optimizer to train a parametric quantum circuit.
- ☐ It takes advantage of quantum parallelism, the superposition of quantum states that allows a circuit to simultaneously process 2^n eigenstates.
- ☐ For the classical machine learning algorithm, the model is such as a neural network running on the classical computer.
- ☐ For the variational quantum algorithm, the model is a quantum circuit running on a quantum computer.

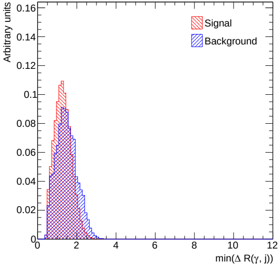
arXiv:2012.09265

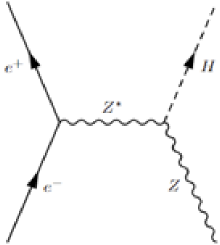


arXiv:2002.09935







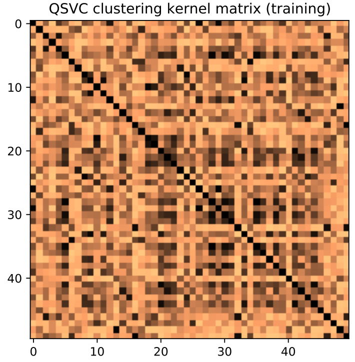


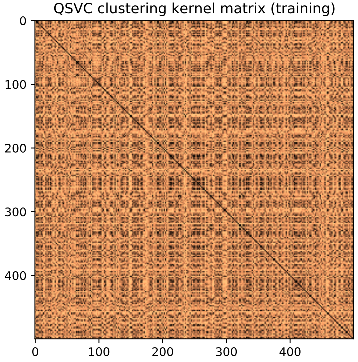
Training and testing strategy

- Two variables as inputs, so the number of qubit is 2 for the QSVM.
- Then training with different dataset size like 50, 100, 150, 200, 500

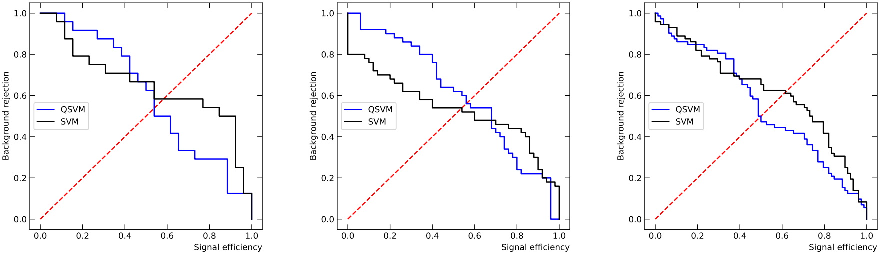
CEPC: $e^+e^- \to Z(\to q\bar{q})H(\to \gamma\gamma)$

Shape of the training kernel matrix in QSVM

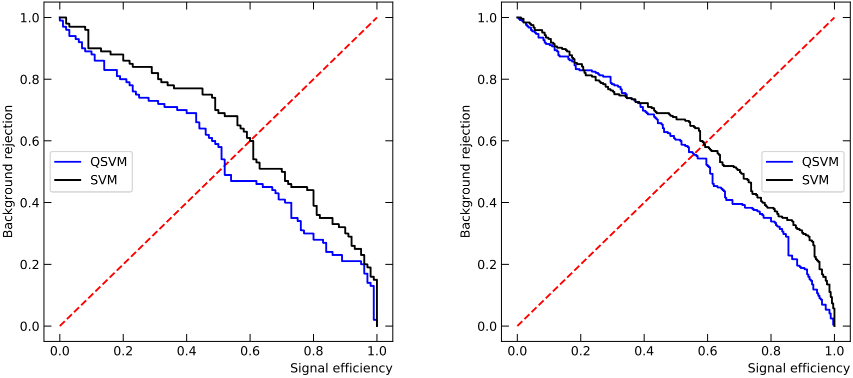




- ☐ The more clustered features the better the kernel.
- ☐ Quantum kernel, slide 6, from 50 to 500 training dataset.



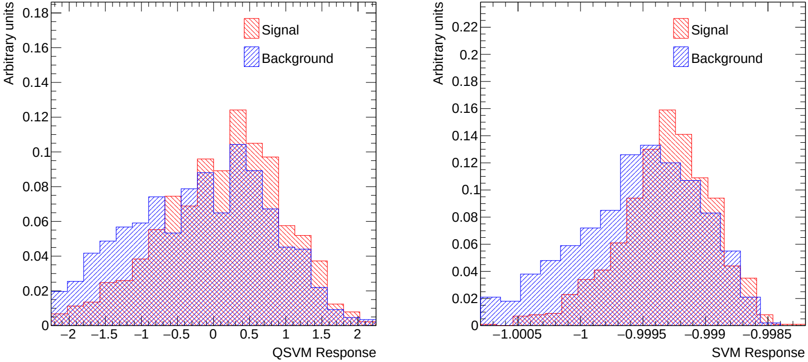
☐ ROC with 50, 100, 150 events fort both training and testing dataset.



☐ ROC for 200 and 500 events for both training and testing dataset.

CEPC: $e^+e^- \to Z(\to q\bar{q})H(\to \gamma\gamma)$

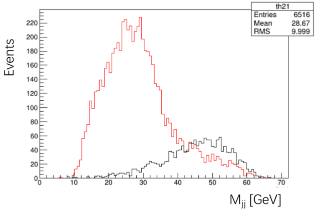
Checking the separation power

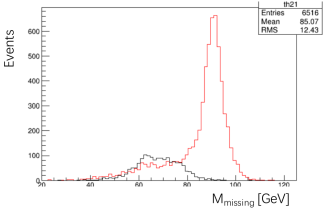


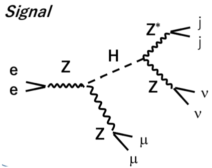
- ☐ 1000 events (training = 500, testing = 500)
- ☐ Unlike the TMVA, it's the separation margin between the classes.

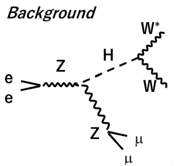
CEPC: $e^+e^- \to Z(\to \mu^-\mu+)H(\to \nu\nu qq)$

Input datasets





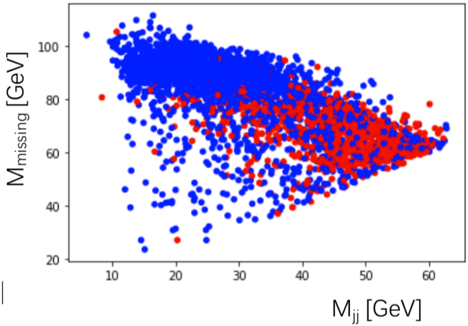




- ☐ Preliminary cut on di-muon invariant mass $80 < M_{\mu\mu} < 100$ GeV
- ☐ 837 background and 3263 signal events
- ☐ Taking the half from each data set for training and testing.

CEPC: $e^+e^- \to Z(\to \mu^-\mu+)H(\to \nu\nu qq)$

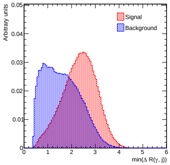
Training and testing output

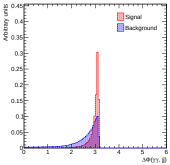


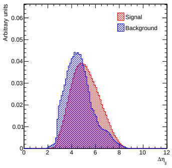
- ☐ VQA with a classical optimiser in IBM simulator.
- ☐ Result of the training and testing:
  - 67.21% signal efficiency
  - Reject 32.42% of the background

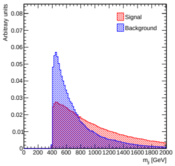
LHC: (VBF) $H \to \gamma\gamma$

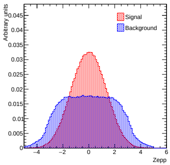
Training and testing strategy

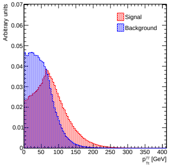


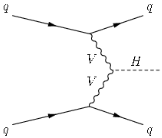










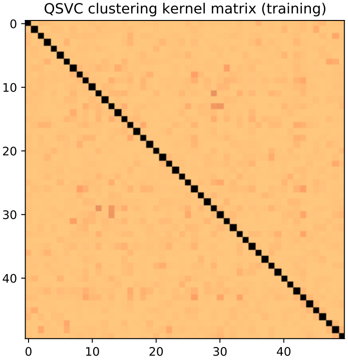


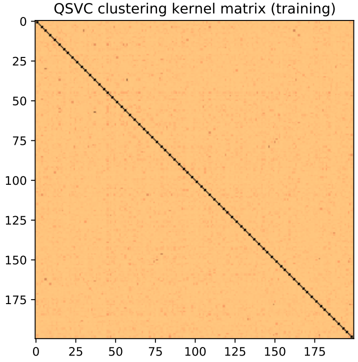
Training and testing strategy

- ☐ Six variables as inputs, so the number of qubit is 6 for the QSVM.
- ☐ Then training with different dataset size like 50, 100, 150, 200, 500

LHC: (VBF) $H \to \gamma\gamma$

Shape of the training kernel matrix in QSVM

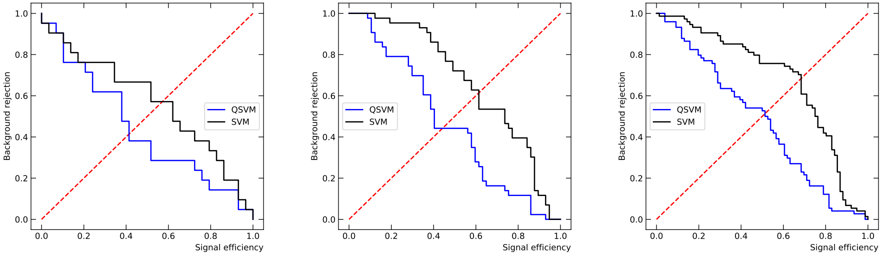




- ☐ The more clustered features the better the kernel.
- ☐ Quantum kernel, slide 6, from 50 to 500 training dataset.

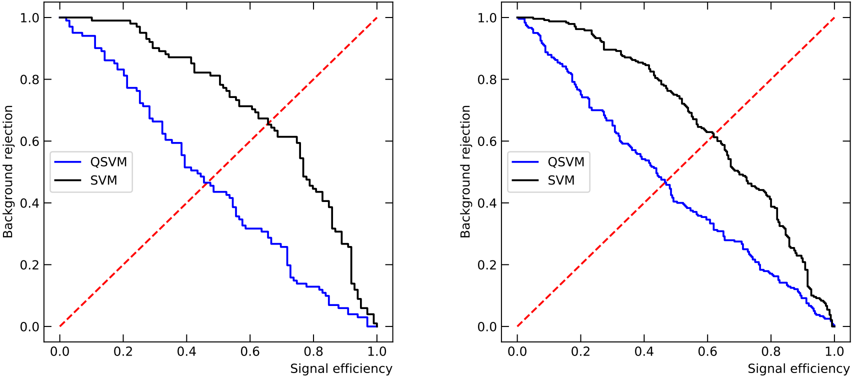
LHC: (VBF) $H \to \gamma\gamma$

ROC



- ☐ ROC with 50, 100, 150 events fort both training and testing dataset.

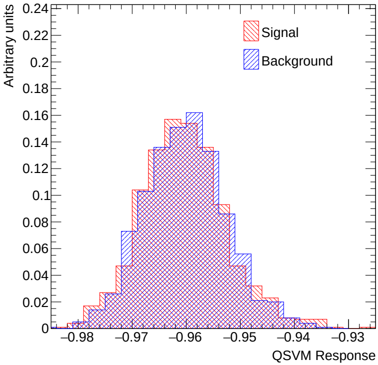
LHC: (VBF) $H \to \gamma\gamma$
ROC

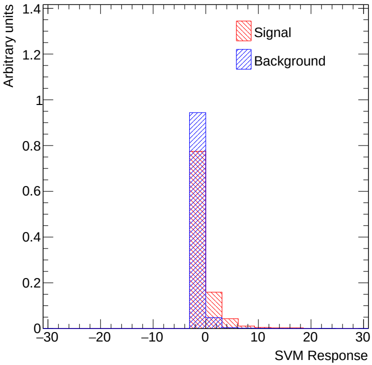


- ☐ ROC for 200 and 500 events for both training and testing dataset.

LHC: (VBF) $H \to \gamma\gamma$

Checking the separation power

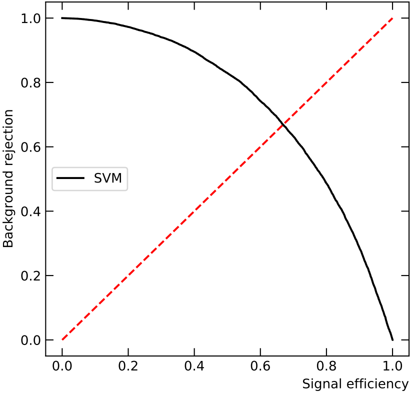




- ☐ 1000 events (training = 500, testing = 500)
- ☐ Unlike the TMVA, it’s the separation margin between the classes.

LHC: (VBF) $H \to \gamma\gamma$

SVM with more stats



- ☐ 20k events (training = 10k, testing = 10k)

Summary

- ☐ QSMV and SVM are compared using few events up to 500 using the qsam simulator:
  - Using 2 qubits $e^+e^- \rightarrow ZH \rightarrow \gamma\gamma q\bar{q}$ (CEPC)
  - Using 6 qubits $pp \rightarrow H \rightarrow \gamma\gamma$ (LHC)
- ☐ CEPC: $e^+e^- \rightarrow Z(\rightarrow \mu^-\mu+)H(\rightarrow \nu\nu qq)$ using VQA.
- ☐ We test the performance of QSVM and SVM using ROCs and SB separation.
- ☐ The computational is time expensive specially when running locally.

TO DO . . .

- Optimizing QSVM, SVM and QVA for better results.
- Train more events after setting the framework in the server.
- Train and test the same algorithm in real IBM quantum computer.

- ☐ We are open to any comments and suggestions as we still learning.

Thank you!

Additional slides

Hadamard $= H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	CNOT $= CX = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$
$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \ S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$	$CZ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$
$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$	Controlled-U $= CU = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & U_{00} & U_{01} \\ 0 & 0 & U_{10} & U_{11} \end{pmatrix}$

NOT $= X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	SWAP $= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	Toffoli (CCNOT) $= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$
$R(\theta) = P(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$	

- ☐ Commonly used single- and multi-qubit quantum gates