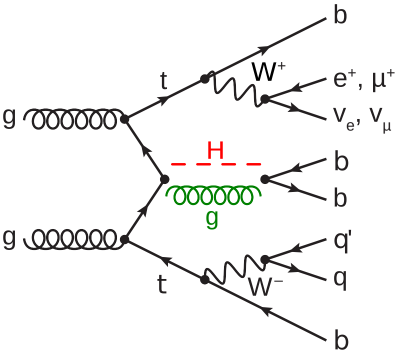
Identifying the Higgs boson production in the $t\bar{t}H(b\bar{b})$ channel using quantum classifier models

Vasilis Belis (ETH Zürich)

Joint Annual Meeting of the Austrian Physical Society and Swiss Physical Society 2021
30 August - 3 September 2021, Universität Innsbruck

September 1, 2021

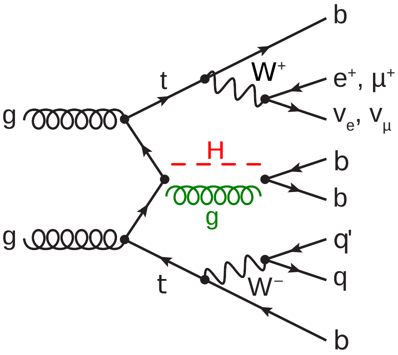
The $t\bar{t}H(b\bar{b})$ process



LO Feynman diagram of the signal and the dominant background processes in the semi-leptonic channel.

$$n^{\text{features}} = 8\times7(\text{jets})+7(\text{lepton})+4(\text{MET}) = 67$$

The $t\bar{t}H(b\bar{b})$ process



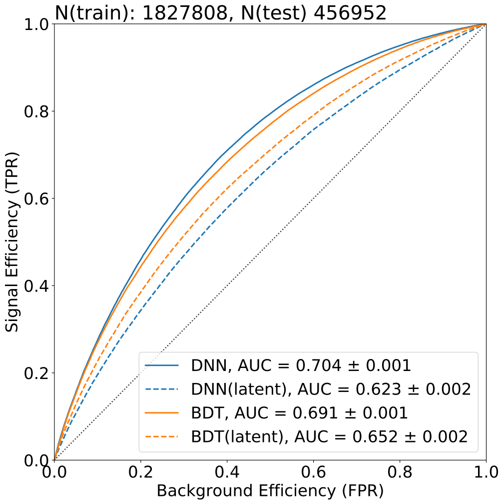
LO Feynman diagram of the signal and the dominant background processes in the semi-leptonic channel.

$n^{\text{features}} = 8\times 7(\text{jets}) + 7(\text{lepton}) + 4(\text{MET}) = 67$

Analysis methods for $t\bar{t}H(b\bar{b})$ utilizing most features:

- ML models: Boosted Decision Trees (BDT), Deep Neural Networks (NN) exploiting all input feature correlations [ATL20, CMS19].
- Define physics-inspired high-level variables (m^2, jet shape, angular differences, etc.).
- State-of the art approaches for $t\bar{t}H(b\bar{b})$: graph and attention networks, etc.

Classification with conventional methods



- Assess performance of realistic HEP approaches on our data set (Delphes simulation).
- Full CMS simulation yields higher classifier performance.
- Models trained on full set of input features (67) and a reduced set (16) $\rightarrow$ benchmark.
- Measure of information loss (discriminating power reduction).

Motivation

Why quantum machine learning for HEP?

- Heuristic answer: investigate the new set of ML techniques and methods available and assess advantages.

Why quantum machine learning for HEP?

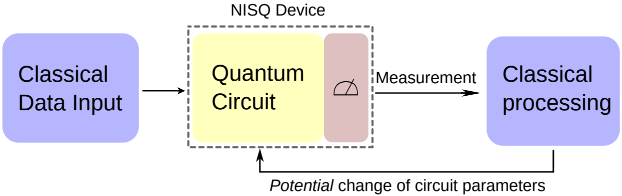
- Heuristic answer: investigate the new set of ML techniques and methods available and assess advantages.
- Fundamental *motivation*: can quantum models utilise the quantum correlations inherent in HEP data leading to performance advantages?
    - Goal in "ML jargon" [KBS21]: Find inductive bias based on prior knowledge on the data generation (*quantum* process for HEP data).
    - If this bias can be *constructed* and is *classically difficult* to simulate $\rightarrow$ *quantum advantage*.

Motivation

Why quantum machine learning for HEP?

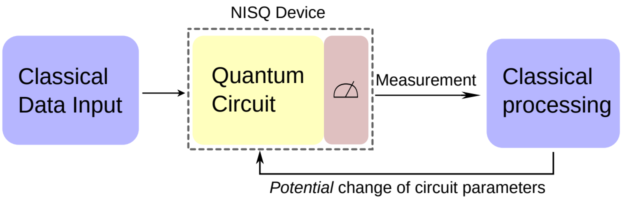
- Heuristic answer: investigate the new set of ML techniques and methods available and assess advantages.
- Fundamental *motivation*: can quantum models utilise the quantum correlations inherent in HEP data leading to performance advantages?
  - Goal in "ML jargon" [KBS21]: Find inductive bias based on prior knowledge on the data generation (*quantum* process for HEP data).
  - If this bias can be *constructed* and is *classically difficult* to simulate $\rightarrow$ *quantum advantage*.
- Example: quantum algorithm for HEP event shower simulation, produces accurate results [NPdJB21]. Can simulate naturally the interference diagram.

Hybrid Quantum-Classical machine learning models



- Noisy Intermediate Scale Quantum (NISQ) devices:
  - *Circuit width*: limited number of qubits (superconducting qubits at IBM ~ 50).
  - *Circuit depth*: limited number of operations per qubit (small decoherence times).

Hybrid Quantum-Classical machine learning models



- Noisy Intermediate Scale Quantum (NISQ) devices:
  - *Circuit width*: limited number of qubits (superconducting qubits at IBM ~ 50).
  - *Circuit depth*: limited number of operations per qubit (small decoherence times).

Quantum Machine learning models for classification:

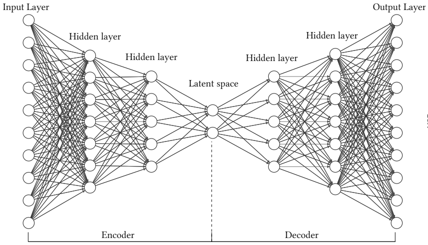
- Kernel methods $\rightarrow$ Quantum Support Vector Machine (QSVM)
- Quantum "Neural Networks" $\rightarrow$ Variational/Parametrized Quantum Circuits (VQC/PQC)

$\rightarrow$ To accommodate NISQ limitations feature reduction is needed.

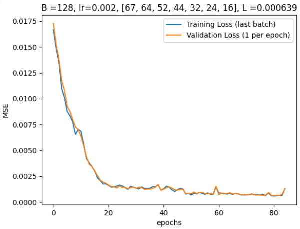
Feature Reduction

1. **AutoEncoders (AE)**

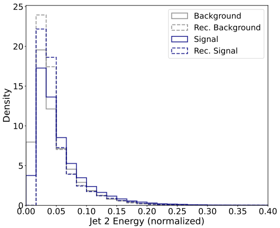
   - Two AutoEncoders: one with 16 latent space features and one with 8.



(a)



(b)

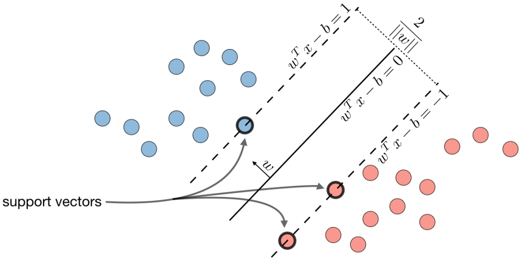


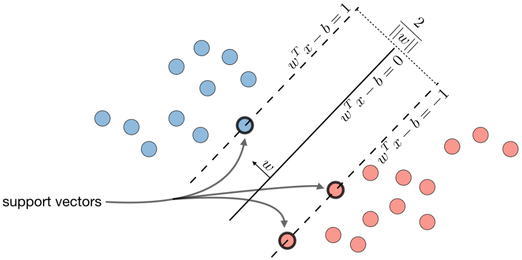
(c)

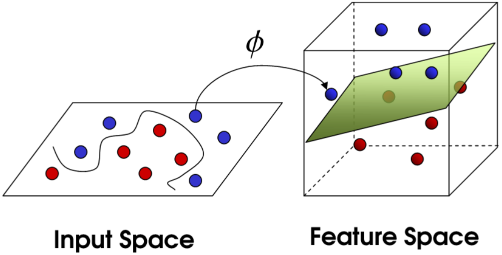
2. **Feature Selection**

   - Select 16 (8) input variables with the highest discriminative power according to their AUC score (Area Under Receiver Operating Characteristic curve).

Support Vector Machines







- SVM objective function is equivalent to (dual Lagrangian)

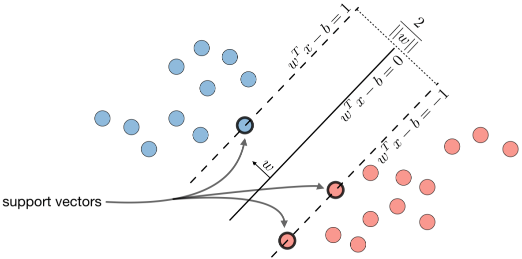
$$\text{maximize } L(c_1 \ldots c_n) = \sum_{i=1}^{n} c_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i c_i (\vec{x}_i \cdot \vec{x}_j) y_j c_j$$

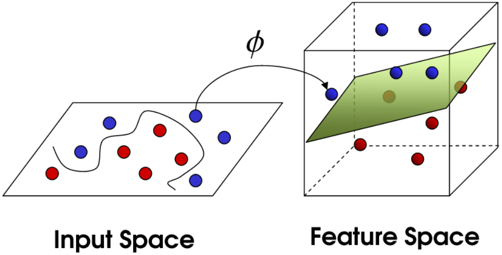
$$\text{subject to } \sum_{i=1}^{n} c_i y_i = 0, \text{ and } 0 \leq c_i \leq C \text{ for all } i$$

- Kernel trick:

$$(\vec{x}_i \cdot \vec{x}_j) \mapsto k(\vec{x}_i, \vec{x}_j) := \phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$$

Support Vector Machines





- SVM objective function is equivalent to (dual Lagrangian)

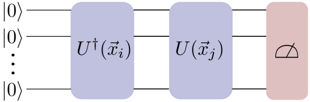
  maximize $L(c_1 \dots c_n) = \sum_{i=1}^{n} c_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i c_i (\vec{x}_i \cdot \vec{x}_j) y_j c_j$

  subject to $\sum_{i=1}^{n} c_i y_i = 0$, and $0 \leq c_i \leq C$ for all i

- Kernel trick:

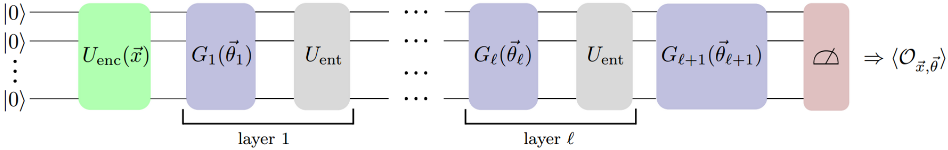
$$(\vec{x}_i \cdot \vec{x}_j) \mapsto k(\vec{x}_i, \vec{x}_j) := \phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$$

- Make the kernel *quantum*:



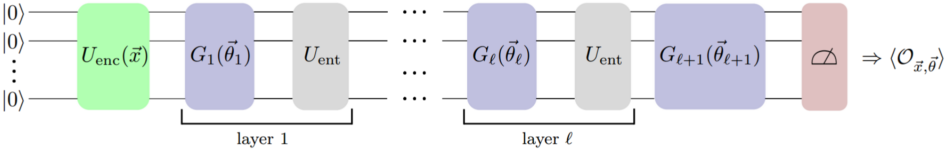
$$\Rightarrow K_{ij} = |\langle 0| U^\dagger(\vec{x}_i) U(\vec{x}_j) |0\rangle|^2$$

Variational Quantum Circuits



- Data embedding circuit (feature map) here is fixed.
- Layers of parametrised quantum gates → trainable parameters.

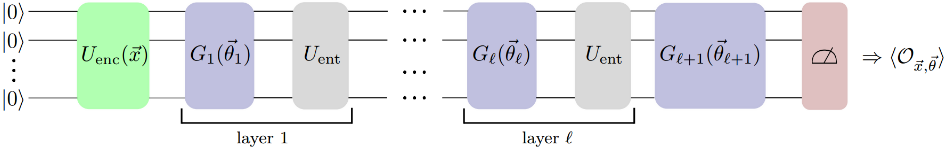
Variational Quantum Circuits



- Data embedding circuit (feature map) here is fixed.
- Layers of parametrised quantum gates → trainable parameters.
- Output of the model → expectation value of an observable on the prepared state $|\psi(\vec{x},\vec{\theta}\,)\rangle$ e.g. measure the first qubit on the computational basis

$$\mathcal{O} = \sigma_z \otimes \mathbb{1} \otimes \cdots \otimes \mathbb{1},$$

Variational Quantum Circuits



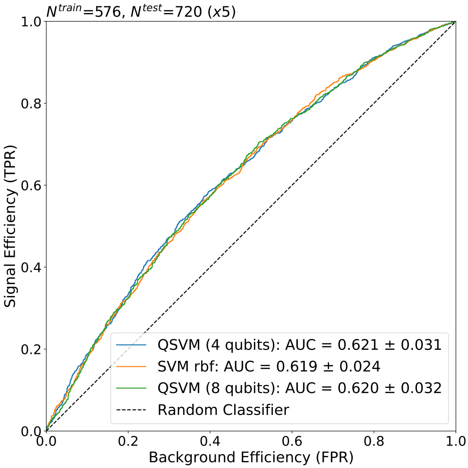
- Data embedding circuit (feature map) here is fixed.
- Layers of parametrised quantum gates → trainable parameters.
- Output of the model → expectation value of an observable on the prepared state $|\psi(\vec{x},\vec{\theta}\,)\rangle$ e.g. measure the first qubit on the computational basis

$$\mathcal{O} = \sigma_z \otimes \mathbb{1} \otimes \cdots \otimes \mathbb{1},$$

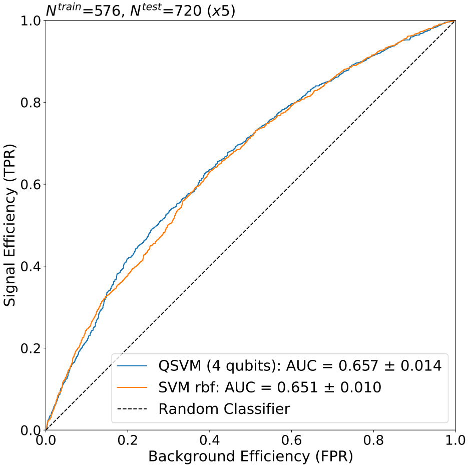
$$f(\vec{x},\vec{\theta}) := \langle \psi(\vec{x},\vec{\theta})| \, \mathcal{O} \, |\psi(\vec{x},\vec{\theta})\rangle \equiv \langle \psi(\vec{x})| \, G^\dagger(\vec{\theta}) \mathcal{O} G(\vec{\theta}) \, |\psi(\vec{x})\rangle \equiv \langle \mathcal{O} \rangle_{\vec{x},\vec{\theta}}.$$

- Classification: if $\langle \mathcal{O} \rangle_{\vec{x},\vec{\theta}} > 0.5$ → signal, otherwise background.

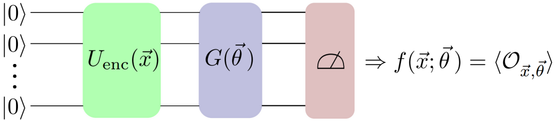
QSVM results with reduced features



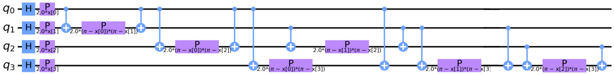
AE latent features (16)



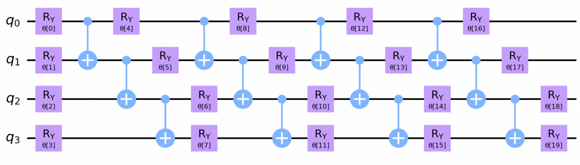
AUC-based input feature selection (16)

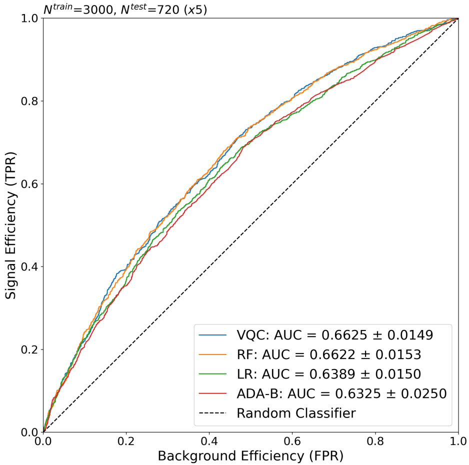


Data encoding for the VQC model [HCTea19]:



Parametrised quantum circuit ("QNN"):





AUC-based input feature selection (8)

Summary

Investigated:

- Different quantum algorithms QSVM and VQC.
- Data encoding circuits (amplitude encoding, direct encoding and data re-uploading).
- Feature dimensionality reduction methods.
- Classical benchmarks against state-of-the-art approaches in HEP and ML.

Our results [BGR+21]:

- Classical and quantum models have similar performance for the challenging $t\bar{t}H(b\bar{b})$ classification task (in agreement with previous studies [TKK+21, BS20, WCG+20, MJV+17]).
- The feature reduction procedure is extremely crucial (high impact on model performance).

Outlook & ongoing work

- Hybrid quantum-classical Autoencoder-based feature reduction.
    - Novel architectures: Preserve/enhance classification power in the latent space.
- Implementation of the algorithms on NISQ devices.
    - Assess the effect of the different noise components on model performance.
    - Error mitigation protocol if needed.
- Anomaly detection studies for model independent searches in HEP.

Thank you!

Backup

$t\bar{t}H(b\bar{b})$ process

Why is it important?

- Study the Yukawa couplings of the Higgs in a purely fermionic process
- $t\bar{t}H$ coupling carries direct information about the scale of new physics [BS15]

$\rightarrow$Both processes have identical final state

Analysis features

Monte Carlo simulation: generation with Powheg v.2, parton shower Pythia 8 and Delphes v.3.4.1 (CMS Run II settings)

- Nominally: $n^{\text{jets}} = 6$ and $n^{\text{b-jets}} = 4$
- Jet observables (8) : $(p_T,\, \eta,\, \phi,\, E,$ b-tag, $p_x,\, p_y,\, p_z)$
- Semi-leptonic channel to reduce QCD background
  $\rightarrow$ 1 lepton and 1 neutrino (MET) per event
- MET observables (4) : (p_T, p_x, p_y, ϕ)
- Lepton observables (7) : $(p_T,\, \eta,\, \phi,\, E,\, p_x,\, p_y,\, p_z)$
- Keep 7 most energetic jets per event allowing for 1 correction of final or initial state radiation

$$\Rightarrow \boxed{n^{\text{features}} = 8 \times 7(\text{jets}) + 7(\text{lepton}) + 4(\text{MET}) = 67}$$

Pre-processing and pre-selection

Object pre-selection:

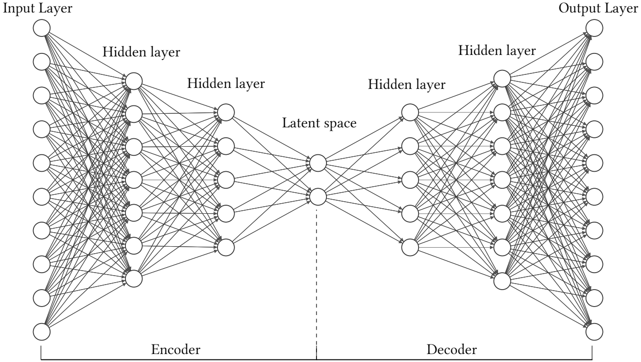
- $p_T > 30$ GeV, $|\eta| < 2.1$ and iso > 0.1 for the electrons
- $p_T > 26$ GeV, $|\eta| < 2.1$ and iso > 0.1 for the muons
- $p_T > 30$ GeV, $|\eta| < 2.4$ for the jets

Event selection:

$$n^{\text{jet}} \geq 4,\ n^{\text{b-tag}} \geq 2 \text{ and } n^{\text{leptons}} = 1$$

- b-tag $\in \{0, 1, \ldots, 7\}$, for different efficiencies
  $\rightarrow$redefinition: $\text{b-tag}' = \begin{cases} 1, & \text{if b-tag} > 1 \\ 0, & \text{otherwise} \end{cases}$

Auto-Encoder model



Goal: Preserve non-linear correlations in the latent representation space

- Developed two models: 8 and 16-dimensional latent space
- Input features normalised to $[0, 1]$ (min-max scaling)

$$x_i \to \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}$$

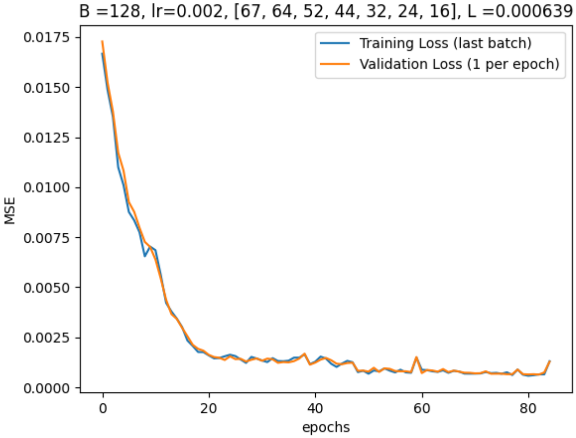
Model Architecture:

- Fully connected feed forward layers
- ELU activation functions. Sigmoid on latent and output layers

Auto-Encoder hyperparameters

	PyTorch AE	TensorFlow AE
Layer Type	Dense	
Encoder hidden layers	6	7
Latent space dim.	16	8
Loss	Mean Square Error (MSE)	
Optimizer	Adam	
Learning Rate	2×10^{-3}	$\sqrt{3} \times 10^{-3}$
Batch size	128	93
Number of epochs	80	30

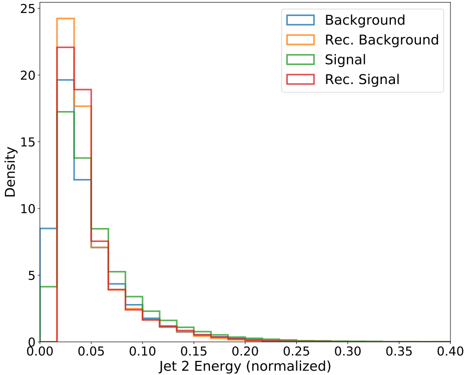
Auto-Encoder training



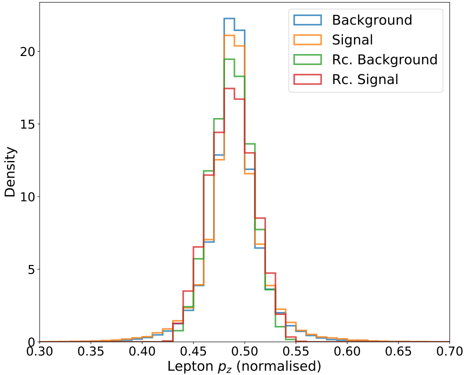
$$L(\vec{\theta}) = \frac{1}{N}\sum_{i=0}^{N} |\vec{x}^{\,i} - \vec{x}^{\,i}_{\vec{\theta}}|^2$$

- Data set split $80\%/10\%/10\%$ (train/validate/test):
  $N^{\text{train}} = 1.1 \times 10^6$
  $N^{\text{test}} = N^{\text{valid.}} = 1.44 \times 10^5$
- Compute validation loss after each epoch (probe for over-training)
- $L^{\text{test}} = 6.41 \times 10^{-4}$

Reconstruction of the features



(d) PyTorch Auto-Encoder (16)



(e) TensorFlow Auto-Encoder (8)

Basics of quantum information processing

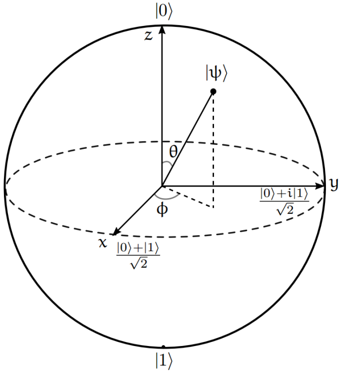
The qubit:

$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$

Basics of quantum information processing

The qubit:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \equiv \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle$$



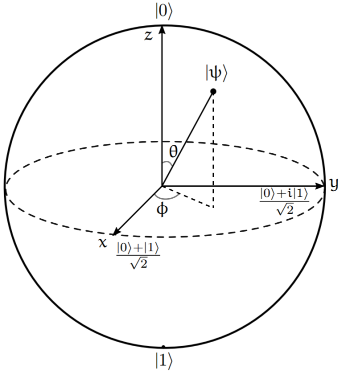
Generic qubit operations (quantum gates) $U = e^{-i\vec{\theta}\cdot\frac{\vec{\sigma}}{2}} \in \mathrm{SU}(2)$:

$$U(\theta,\phi,\lambda) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -e^{i\lambda}\sin\left(\frac{\theta}{2}\right) \\ e^{i\phi}\sin\left(\frac{\theta}{2}\right) & e^{i(\phi+\lambda)}\cos\left(\frac{\theta}{2}\right) \end{pmatrix}$$

Basics of quantum information processing

The qubit:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \equiv \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle$$

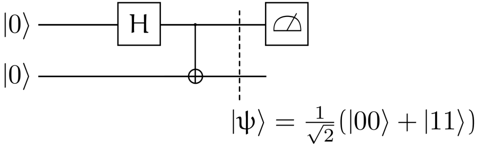


Generic qubit operations (quantum gates) $U = e^{-i\vec{\theta}\cdot\frac{\vec{\sigma}}{2}} \in \mathrm{SU}(2)$:

$$U(\theta,\phi,\lambda) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -e^{i\lambda}\sin\left(\frac{\theta}{2}\right) \\ e^{i\phi}\sin\left(\frac{\theta}{2}\right) & e^{i(\phi+\lambda)}\cos\left(\frac{\theta}{2}\right) \end{pmatrix}$$

Construct all possible gates from $U(\theta,\phi,\lambda)$

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \equiv U\left(\frac{\pi}{2}, 0, \pi\right)$$



Quantum gates and universality

Single qubit gates:

- A generic quantum gate can be decomposed in a series of R_y and R_z [BBC+95]

$$U(\theta, \phi, \lambda) = R_z(\lambda) R_y(\theta) R_z(\phi)$$

- For hardware implementation: more convenient to decompose to gates that have a direct physical operation analogue on the device.

Quantum gates and universality

Single qubit gates:

- A generic quantum gate can be decomposed in a series of R_y and R_z [BBC+95]

$$U(\theta, \phi, \lambda) = R_z(\lambda) R_y(\theta) R_z(\phi)$$

- For hardware implementation: more convenient to decompose to gates that have a direct physical operation analogue on the device.

Multi-qubit gates:

- 2-qubit SWAP and CNOT (Control-X) gates and the 3-qubit Toffolli gate

$$CX = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

- Any control-U gate can be written as a combination of CX, R_y and R_z gates.

Single qubit gates:

- A generic quantum gate can be decomposed in a series of R_y and R_z [BBC+95]

$$U(\theta, \phi, \lambda) = R_z(\lambda) R_y(\theta) R_z(\phi)$$

- For hardware implementation: more convenient to decompose to gates that have a direct physical operation analogue on the device.

Multi-qubit gates:

- 2-qubit SWAP and CNOT (Control-X) gates and the 3-qubit Toffolli gate

$$CX = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

- Any control-U gate can be written as a combination of CX, R_y and R_z gates.

Quantum Gate Universality [DiV95]: The above “building blocks” can construct any quantum circuit acting on n qubits, i.e. SU(2^n), operating on at most *two-qubits* at a time.

Quantum classifiers

Kernel-based models (Quantum Support Vector Machines):

- Convex optimization tasks
- $\mathcal{O}(n^2)$ complexity construction of the kernel matrix elements

Quantum Neural Networks (Variational Quantum Circuits):

- Non-convex optimization
- $\mathcal{O}(n)$ complexity

Quantum classifiers

Kernel-based models (Quantum Support Vector Machines):

- Convex optimization tasks
- $\mathcal{O}(n^2)$ complexity construction of the kernel matrix elements

Quantum Neural Networks (Variational Quantum Circuits):

- Non-convex optimization
- $\mathcal{O}(n)$ complexity

Encoding (embedding) the classical data in a quantum circuit [SP18]:

$$|\psi(x)\rangle = G(\vec{x})\,|0\rangle^{\otimes n_{\text{qubits}}}$$

Quantum classifiers

Kernel-based models (Quantum Support Vector Machines):

- Convex optimization tasks
- $\mathcal{O}(n^2)$ complexity construction of the kernel matrix elements

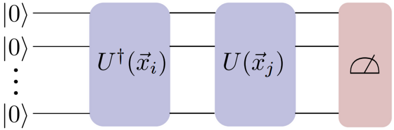
Quantum Neural Networks (Variational Quantum Circuits):

- Non-convex optimization
- $\mathcal{O}(n)$ complexity

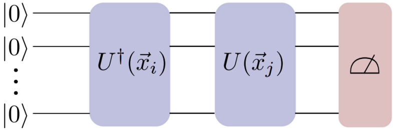
Encoding (embedding) the classical data in a quantum circuit [SP18]:

$$|\psi(x)\rangle = G(\vec{x})\,|0\rangle^{\otimes n_{\text{qubits}}}$$

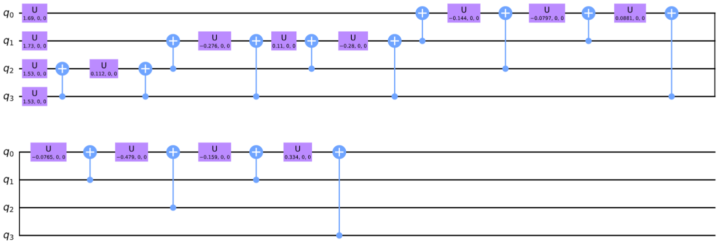
- Amplitude encoding: exponentially decrease the needed number of qubits *but* have deep circuits
- Angle (direct) encoding: map each feature to a separate qubit shallow but wider circuits
- Data re-uploading [PSCLGFL20]: repeat any data embedding circuit



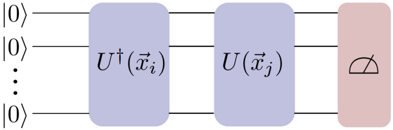
$$\Rightarrow K_{ij} = |\langle 0 | U^\dagger(\vec{x}_i) U(\vec{x}_j) | 0 \rangle|^2$$



$$\Rightarrow K_{ij} = |\langle 0 | U^\dagger(\vec{x}_i) U(\vec{x}_j) | 0 \rangle|^2$$

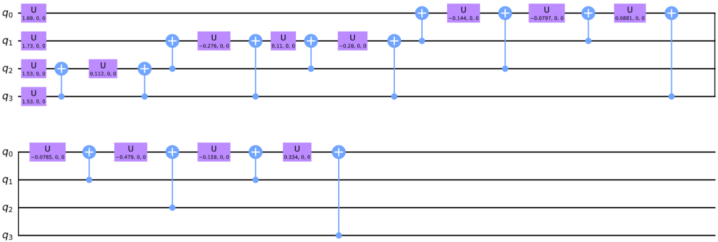


Quantum Support Vector Machines



$$\Rightarrow K_{ij} = |\langle 0|U^\dagger(\vec{x}_i)U(\vec{x}_j)|0\rangle|^2$$

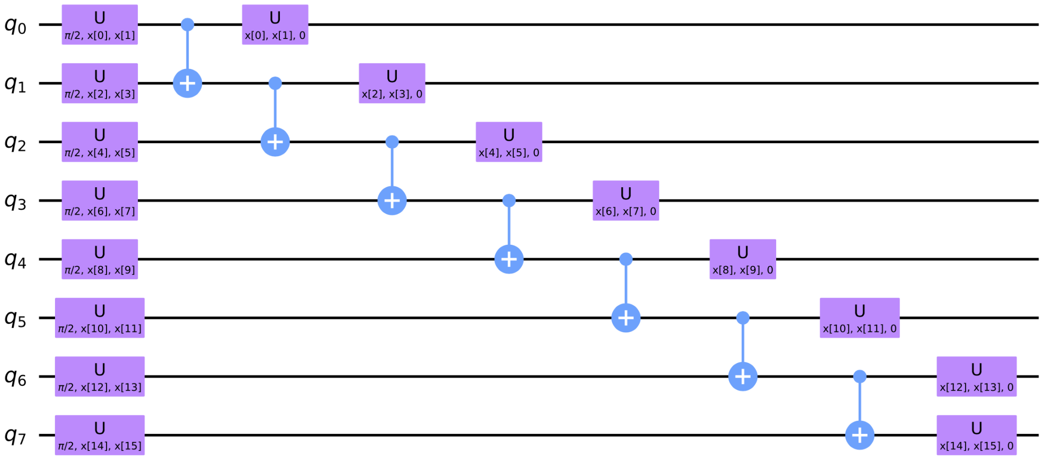
- Sample the kernel matrix on a quantum device (multiple measurements)
- Maximise the SVM objective function on a classical computer



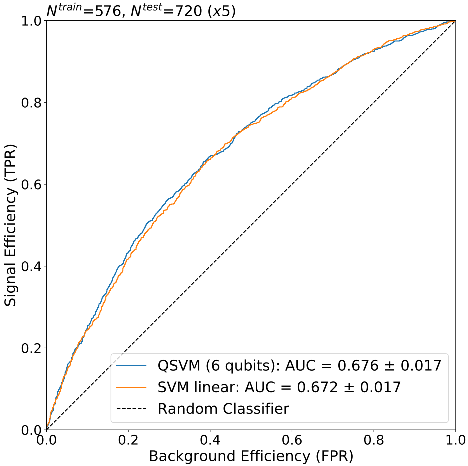
Amplitude encoding circuit

$$\phi : \mathbb{R}^N \rightarrow \mathcal{H}^{\otimes n^{\text{qubits}}} \Rightarrow \vec{x} \in \mathbb{R}^{16} \rightarrow |\psi_{\vec{x}}\rangle = \frac{1}{4}\sum_{i=0}^{15} m_i\,|i\rangle\,,\ m_i \text{ norm. inputs}$$

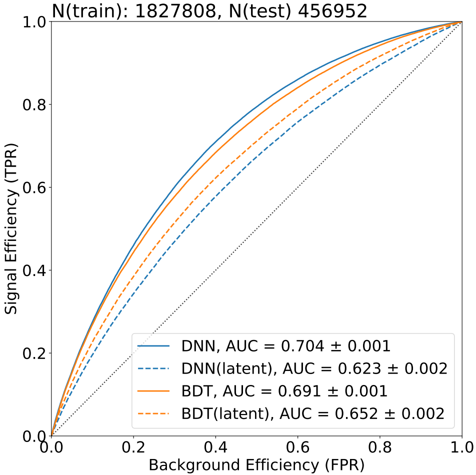
Alternative data encoding circuit (8-qubits)



QSVM results on the input space



64 out of the 67 input features



“Realistic” approach

QSVM feature reduction benchmarks

Feature selection + Model	AUC
INFO + QSVM	0.66 ± 0.01
PyTorch AE + QSVM	0.62 ± 0.03
INFO + SVM rbf	0.65 ± 0.01
PyTorch AE + SVM rbf	0.62 ± 0.02
KMeans + SVM rbf	0.61 ± 0.02

(a) 16 input variables

Feature selection + Model	AUC
INFO + QSVM	0.68 ± 0.02
INFO + Linear SVM	0.67 ± 0.02
Logistic Regression	0.68 ± 0.02

(b) 64 (QSVM, LSVM) and 67 (LR) input variables

- Trained and tested (same data set size) a collection of classical models (SVMs, Logistic Regression, BDT, Random Forests, Multilayer Perceptrons, kNN, Naive Bayes and QDA).
- Feature extraction techniques: PCA, K-means, Truncated SVD, Isomap and Locally Linear Embedding.

VQC benchmarks

Feature selection + Model	AUC
INFO + VQC	0.66 ± 0.01
INFO + Random Forest	0.66 ± 0.02
KMeans + Log. Regr.	0.64 ± 0.01
TensorFlow AE + AdaBoost	0.63 ± 0.03

- Needs more training data to achieve same performance as QSVM.
- VQC poor performance with amp. enc. 16 features and 8 AE features (AUC~ 0.55)
  → resort to feature selection of 8 input features.

- *Measurement of the Higgs boson decaying to b-quarks produced in association with a top-quark pair in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector*, Tech. Report ATLAS-CONF-2020-058, CERN, Geneva, Nov 2020.
- Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A. Smolin, and Harald Weinfurter, *Elementary gates for quantum computation*, Phys. Rev. A **52** (1995), 3457–3467.
- Belis Vasilis, González-Castillo Samuel, Reissel Christina, Vallecorsa Sofia, Combarro Elías F., Dissertori Günther, and Reite Florentin, *Higgs analysis with quantum classifiers*, EPJ Web Conf. **251** (2021), 03070.

- F. Bezrukov and M. Shaposhnikov, *Why should we care about the top quark yukawa coupling?*, Journal of Experimental and Theoretical Physics **120** (2015), no. 3, 335–343.
- Andrew Blance and Michael Spannowsky, *Quantum machine learning for particle physics using a variational quantum classifier*, arXiv preprint arXiv:2010.07335 (2020).
- *Measurement of* $\mathrm{t\bar{t}H}$ *production in the* $\mathrm{H} \rightarrow \mathrm{b\bar{b}}$ *decay channel in* $41.5\,\mathrm{fb}^{-1}$ *of proton-proton collision data at* $\sqrt{s} = 13\,\mathrm{TeV}$, Tech. Report CMS-PAS-HIG-18-030, CERN, Geneva, 2019.
- David P. DiVincenzo, *Two-bit gates are universal for quantum computation*, Phys. Rev. A **51** (1995), 1015–1022.

- V. Havlíček, A.D. Córcoles, K. Temme, and et al, *Supervised learning with quantum-enhanced feature spaces*, Nature **567** (2019), 209–212.
- Jonas M. Kübler, Simon Buchholz, and Bernhard Schölkopf, *The inductive bias of quantum kernels*, 2021.
- Alex Mott, Joshua Job, Jean-Roch Vlimant, Daniel Lidar, and Maria Spiropulu, *Solving a higgs optimization problem with quantum annealing for machine learning*, Nature **550** (2017), no. 7676, 375–379.
- Benjamin Nachman, Davide Provasoli, Wibe A. de Jong, and Christian W. Bauer, *Quantum algorithm for high energy physics simulations*, Phys. Rev. Lett. **126** (2021), 062001.
- Adrián Pérez-Salinas, Alba Cervera-Lierta, Elies Gil-Fuster, and José I Latorre, *Data re-uploading for a universal quantum classifier*, Quantum **4** (2020), 226.

- Maria Schuld and Francesco Petruccione, *Supervised learning with quantum computers*, Springer International Publishing, 2018.
- Koji Terashi, Michiru Kaneda, Tomoe Kishimoto, Masahiko Saito, Ryu Sawada, and Junichi Tanaka, *Event classification with quantum machine learning in high-energy physics*, Computing and Software for Big Science **5** (2021), no. 1, 1–11.
- Sau Lan Wu, Jay Chan, Wen Guan, Shaojun Sun, Alex Wang, Chen Zhou, Miron Livny, Federico Carminati, Alberto Di Meglio, Andy CY Li, et al., *Application of quantum machine learning using the quantum variational classifier method to high energy physics analysis at the lhc on ibm quantum computer simulator and hardware with 10 qubits*, arXiv preprint arXiv:2012.11560 (2020).