Aalto University
School of Science
and Technology

Group Invariance in Quantum Kernels for Vector Boson Scattering Identification at the LHC

Väinö Mehtola

Department of Applied Physics
Aalto University, School of Science
VTT Technical Research Centre of Finland
vaino.mehtola@vtt.fi

July 11, 2025

Advisors & Supervisor

Advisors: Michele Grossi (CERN QTI), Santeri Laurila (CERN, HIP), Massimiliano Incudini (Intel)
Supervisor: Alexandru Paler (Aalto)

Motivation

- Quantum machine learning (QML) offers theoretical speedups with certain fault-tolerant subroutines or in otherwise contrived cases[1,2,3]

[1] Molteni, Gyurik, and Dunjko, "Exponential quantum advantages in learning quantum observables from classical data".

[2] Liu, Arunachalam, and Temme, "A rigorous and robust quantum speed-up in supervised machine learning".

[3] Biamonte et al., "Quantum machine learning".

Motivation

- Quantum machine learning (QML) offers theoretical speedups with certain fault-tolerant subroutines or in otherwise contrived cases[1][2][3]
- The LHC produces great volumes of high-dimensional collision data, driving the need for new analysis methods

[1] Molteni, Gyurik, and Dunjko, "Exponential quantum advantages in learning quantum observables from classical data".

[2] Liu, Arunachalam, and Temme, "A rigorous and robust quantum speed-up in supervised machine learning".

[3] Biamonte et al., "Quantum machine learning".

Motivation

- Quantum machine learning (QML) offers theoretical speedups with certain fault-tolerant subroutines or in otherwise contrived cases[1][2][3]
- The LHC produces great volumes of high-dimensional collision data, driving the need for new analysis methods
- Actual implementation of QML models on real hardware may often face the common problem of exponential concentration (Barren plateaus)

---

[1] Molteni, Gyurik, and Dunjko, "Exponential quantum advantages in learning quantum observables from classical data".

[2] Liu, Arunachalam, and Temme, "A rigorous and robust quantum speed-up in supervised machine learning".

[3] Biamonte et al., "Quantum machine learning".

Motivation

- Quantum machine learning (QML) offers theoretical speedups with certain fault-tolerant subroutines or in otherwise contrived cases[1][2][3]
- The LHC produces great volumes of high-dimensional collision data, driving the need for new analysis methods
- Actual implementation of QML models on real hardware may often face the common problem of exponential concentration (Barren plateaus)
- Incorporating domain knowledge (e.g. symmetries of the problem) can 1) improve the inductive bias of the learning model & 2) reduce the risk of exponential concentration

[1] Molteni, Gyurik, and Dunjko, "Exponential quantum advantages in learning quantum observables from classical data".

[2] Liu, Arunachalam, and Temme, "A rigorous and robust quantum speed-up in supervised machine learning".

[3] Biamonte et al., "Quantum machine learning".

Research Questions

1. *What is the impact of incorporating permutation-group invariance on the i) classification performance and ii) scalability of the studied fidelity-based quantum kernels?*

Research Questions

1. *What is the impact of incorporating permutation-group invariance on the i) classification performance and ii) scalability of the studied fidelity-based quantum kernels?*
2. *How does kernel-bandwidth-tuning affect the two aforementioned factors, both in ideal simulations and under shot noise?*

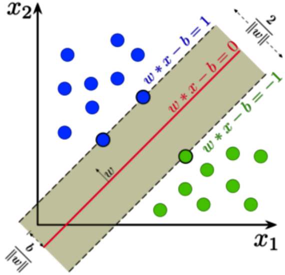
Research Questions

1. *What is the impact of incorporating permutation-group invariance on the i) classification performance and ii) scalability of the studied fidelity-based quantum kernels?*
2. *How does kernel-bandwidth-tuning affect the two aforementioned factors, both in ideal simulations and under shot noise?*
3. *To what extent do findings from simulated environments carry over to real hardware on VTT's Q50 processor?*

Support Vector Machines

$$y_i = sign(b + \sum_{j}^{N} w_j * x_i^T x_j)$$

$$y_i \in \{-1, +1\}$$



Intuition Behind Kernel Methods

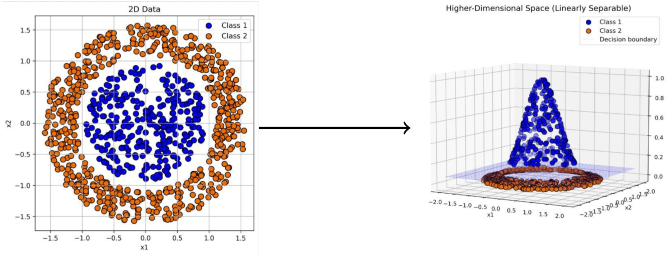


Figure: Left: original 2D view; Right: its 3D projection.

Kernel Function

Let a $\mathcal{D}$ be a dataset belonging to the data space $\mathcal{X}$ (usually, $\mathcal{X} \equiv \mathbb{R}^n$). A kernel function is

Kernel Function

Let a $\mathcal{D}$ be a dataset belonging to the data space $\mathcal{X}$ (usually, $\mathcal{X} \equiv \mathbb{R}^n$). A kernel function is

$$\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0} \tag{1}$$

Kernel Function

Let a $\mathcal{D}$ be a dataset belonging to the data space $\mathcal{X}$ (usually, $\mathcal{X} \equiv \mathbb{R}^n$). A kernel function is

$$\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0} \tag{1}$$

that implicitly defines a feature map

$$\phi : \mathcal{X} \to \mathcal{H} \tag{2}$$

Kernel Function

Let a $\mathcal{D}$ be a dataset belonging to the data space $\mathcal{X}$ (usually, $\mathcal{X} \equiv \mathbb{R}^n$). A kernel function is

$$\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0} \tag{1}$$

that implicitly defines a feature map

$$\phi : \mathcal{X} \to \mathcal{H} \tag{2}$$

into $\mathcal{H}$ (with $\dim(\mathcal{X}) < \dim(\mathcal{H})$ usually).

Kernel Function

Let a $\mathcal{D}$ be a dataset belonging to the data space $\mathcal{X}$ (usually, $\mathcal{X} \equiv \mathbb{R}^n$). A kernel function is

$$\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0} \tag{1}$$

that implicitly defines a feature map

$$\phi : \mathcal{X} \to \mathcal{H} \tag{2}$$

into $\mathcal{H}$ (with $\dim(\mathcal{X}) < \dim(\mathcal{H})$ usually). The kernel function expressed in this way becomes

$$\kappa(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle_{\mathcal{H}}. \tag{3}$$

Kernel Function

Let a $\mathcal{D}$ be a dataset belonging to the data space $\mathcal{X}$ (usually, $\mathcal{X} \equiv \mathbb{R}^n$). A kernel function is

$$\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0} \tag{1}$$

that implicitly defines a feature map

$$\phi : \mathcal{X} \to \mathcal{H} \tag{2}$$

into $\mathcal{H}$ (with $\dim(\mathcal{X}) < \dim(\mathcal{H})$ usually). The kernel function expressed in this way becomes

$$\kappa(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle_{\mathcal{H}}. \tag{3}$$

Properties:

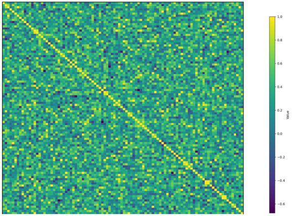
- symmetric: $\kappa(\mathbf{x}, \mathbf{x}') = \kappa(\mathbf{x}', \mathbf{x})$
- positive semi-definite: $\kappa(\mathbf{x}, \mathbf{x}') \geq 0$.

Examples of Kernel Functions

- **Linear:**
$$\kappa(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}'$$
- **Polynomial:**
$$\kappa(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^\top \mathbf{x}' + c)^d$$
where $c \in \mathrm{R}, d \in \mathrm{Z}$
- **Gaussian:**
$$\kappa(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2\right)$$
  - one of the most widely used kernels in all of machine learning
  - can approximate any continuous function on a compact set[4]
  - here, $\gamma > 0$ represents the bandwidth

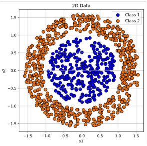
[4] Micchelli, Xu, and Zhang, "Universal Kernels".

The Gram Matrix in Kernel Methods

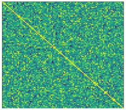


- Given training set $\mathcal{D}_{\text{train}} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ and kernel κ, we form the *Gram matrix*
$$K_{ij} = \kappa\Big(\mathbf{x}^{(i)},\, \mathbf{x}^{(j)}\Big), \quad i, j = 1, \ldots, N.$$
- Key properties:
  - K is symmetric: $K_{ij} = K_{ji}$
  - diagonal entries $K_{ii} = \kappa(\mathbf{x}^{(i)}, \mathbf{x}^{(i)})$ are constant under normalization
  - scales quadratically in the number of datapoints N

Kernel Method Pipeline

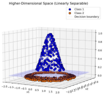


$\rightarrow$ $\kappa(x, x')$ Kernel function $\rightarrow$



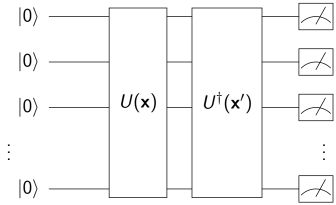
Gram matrix

$\xrightarrow{\text{solve SVM}}$



Solve (linear decision)

Quantum Fidelity Kernels



- **Fidelity kernel definition:**

$$\kappa_Q(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle = \mathrm{Tr}\big[\rho(\mathbf{x})\, \rho(\mathbf{x}')\big]$$

where the feature map is $\rho(\mathbf{x}) = U(\mathbf{x})\, (|0\rangle\langle 0|)^{\otimes n}\, U^\dagger(\mathbf{x})$.

- **Full trace notation:**

$$\kappa_Q(\mathbf{x}, \mathbf{x}') = \mathrm{Tr}\big[U^\dagger(\mathbf{x}')\, U(\mathbf{x})(|0\rangle\langle 0|)^{\otimes n} U^\dagger(\mathbf{x})\, U(\mathbf{x}')(|0\rangle\langle 0|)^{\otimes n}\big].$$

Exponential Concentration in Quantum Kernels

Deterministic Exponential Concentration

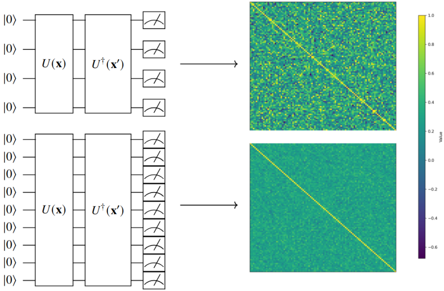
Definition from[5]: a quantity $X(\alpha)$ is *exponentially concentrated* in the number of qubits n toward μ if

$$\left|X(\alpha) - \mu\right| \;\leq\; \beta \in O\left(1/b^n\right) \quad \text{for some } b > 1 \text{ and all } \alpha.$$

- In quantum kernels, $X(\alpha) = \kappa_Q(\mathbf{x}, \mathbf{x}')$.

[5] Thanasilp et al., "Exponential concentration in quantum kernel methods".

Exponential Concentration in Quantum Kernels



- As n_{qubits} grows, off-diagonal kernel values collapse toward μ exponentially fast.

Kernel Bandwidth γ - an Important Hyperparameter

- The bandwidth γ rescales each data point linearly:

$$\tilde{\mathbf{x}} = \gamma\,\mathbf{x}, \quad \tilde{\mathbf{x}}' = \gamma\,\mathbf{x}'.$$

[6] Shaydulin and Wild, "Importance of kernel bandwidth in quantum machine learning".

Kernel Bandwidth γ - an Important Hyperparameter

- The bandwidth γ rescales each data point linearly:

$$\tilde{\mathbf{x}} = \gamma\,\mathbf{x}, \quad \tilde{\mathbf{x}}' = \gamma\,\mathbf{x}'.$$

- In the fidelity kernel, this enters explicitly in the feature map:

$$\kappa_Q(\mathbf{x}, \mathbf{x}') = \mathrm{Tr}\left[\rho(\tilde{\mathbf{x}})\,\rho(\tilde{\mathbf{x}}')\right], \quad \rho(\tilde{\mathbf{x}}) = U(\gamma\,\mathbf{x})\,|0\rangle\langle 0|\,U^\dagger(\gamma\,\mathbf{x}).$$

[6] Shaydulin and Wild, "Importance of kernel bandwidth in quantum machine learning".

Kernel Bandwidth γ - an Important Hyperparameter

- The bandwidth γ rescales each data point linearly:

$$\tilde{\mathbf{x}} = \gamma\,\mathbf{x}, \quad \tilde{\mathbf{x}}' = \gamma\,\mathbf{x}'.$$

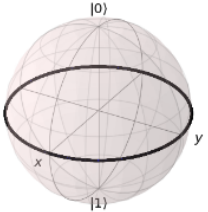
- In the fidelity kernel, this enters explicitly in the feature map:

$$\kappa_Q(\mathbf{x}, \mathbf{x}') = \mathrm{Tr}\left[\rho(\tilde{\mathbf{x}})\,\rho(\tilde{\mathbf{x}}')\right], \quad \rho(\tilde{\mathbf{x}}) = U(\gamma\,\mathbf{x})\,|0\rangle\langle 0|\,U^\dagger(\gamma\,\mathbf{x}).$$

- γ critically affects generalization[6].

[6] Shaydulin and Wild, "Importance of kernel bandwidth in quantum machine learning".

The Effect of γ on the Possible States of the QML Model



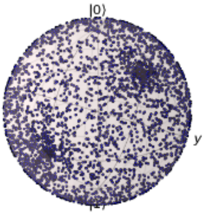
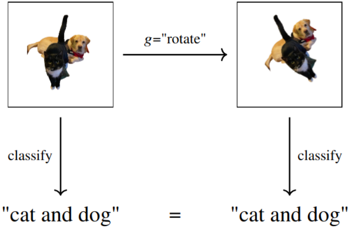


Figure: Left: $\gamma = 10^{-4}$. Right: $\gamma = 1.5$.

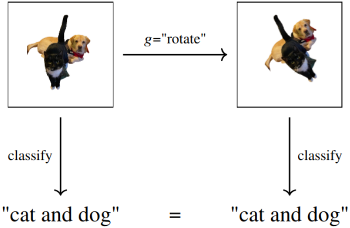
Invariance in ML



[7] Tahmasebi and Jegelka, “The Exact Sample Complexity Gain from Invariances for Kernel Regression”.

[8] Sokolic et al., “Generalization Error of Invariant Classifiers”.

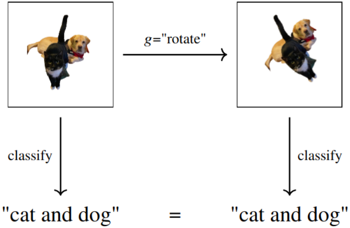
Invariance in ML



- **Embedding known symmetries** can boost data efficiency[7] and generalization[8].

[7] Tahmasebi and Jegelka, "The Exact Sample Complexity Gain from Invariances for Kernel Regression".
[8] Sokolic et al., "Generalization Error of Invariant Classifiers".

Invariance in ML

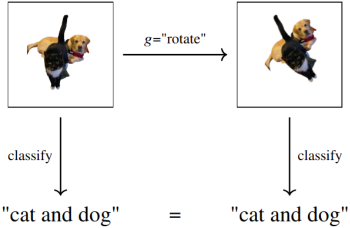


- **Embedding known symmetries** can boost data efficiency[7] and generalization[8].
- **Classic examples:**

[7] Tahmasebi and Jegelka, "The Exact Sample Complexity Gain from Invariances for Kernel Regression".

[8] Sokolic et al., "Generalization Error of Invariant Classifiers".

Invariance in ML

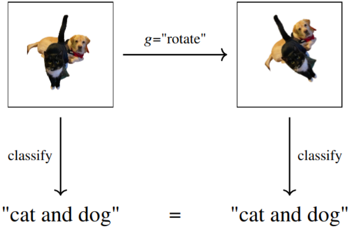


- **Embedding known symmetries** can boost data efficiency[7] and generalization[8].
- **Classic examples:**
  - CNNs – translation *invariance* in images.

[7] Tahmasebi and Jegelka, "The Exact Sample Complexity Gain from Invariances for Kernel Regression".

[8] Sokolic et al., "Generalization Error of Invariant Classifiers".

Invariance in ML

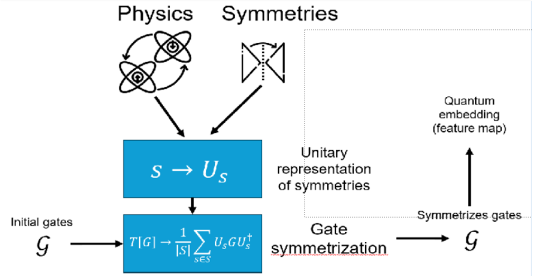


- **Embedding known symmetries** can boost data efficiency[7] and generalization[8].
- **Classic examples:**
  - CNNs – translation *invariance* in images.
  - GNNs – permutation *equivariance* on graph-structured data.

[7] Tahmasebi and Jegelka, "The Exact Sample Complexity Gain from Invariances for Kernel Regression".

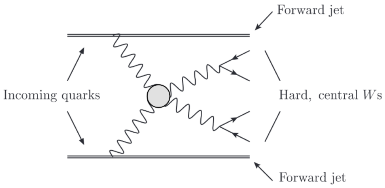
[8] Sokolic et al., "Generalization Error of Invariant Classifiers".

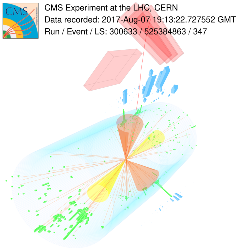
Pauli Twirling (Invariance in QML)



(+ given an initial embedding!)

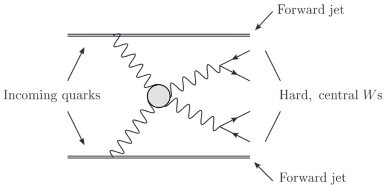
Vector Boson Scattering

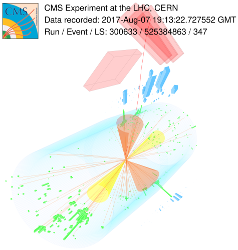




- Two quarks emit one electroweak vector boson (W/Z) each, that then scatter off one another and decay to the detector.

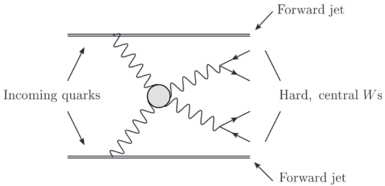
Vector Boson Scattering

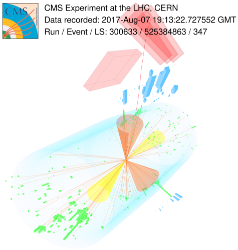




- Two quarks emit one electroweak vector boson (W/Z) each, that then scatter off one another and decay to the detector.
- VBS allows us to probe the EWSB and Higgs' mechanism through polarization measurements.

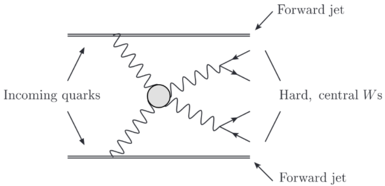
Vector Boson Scattering

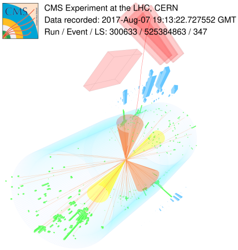




- Two quarks emit one electroweak vector boson (W/Z) each, that then scatter off one another and decay to the detector.
- VBS allows us to probe the EWSB and Higgs' mechanism through polarization measurements.
- Classification problem in this work: VBS WW all-hadronic vs. QCD background.

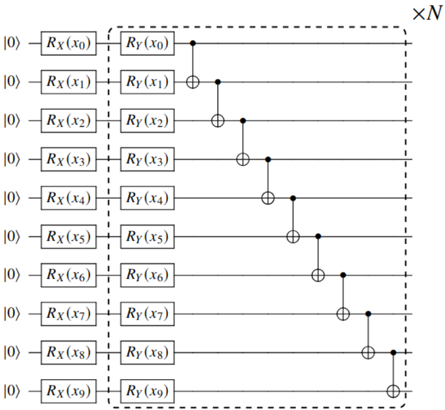
Vector Boson Scattering

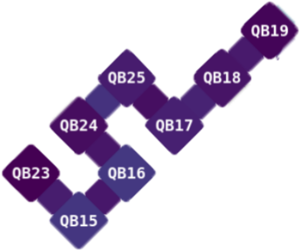




- Two quarks emit one electroweak vector boson (W/Z) each, that then scatter off one another and decay to the detector.
- VBS allows us to probe the EWSB and Higgs' mechanism through polarization measurements.
- Classification problem in this work: VBS WW all-hadronic vs. QCD background.
- Permutation symmetry of the two boson-decay jets (and the forward jets) as the underlying symmetry group.

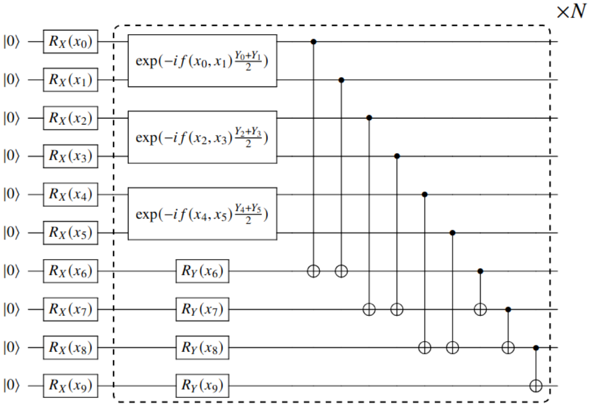
HEA – Baseline Model





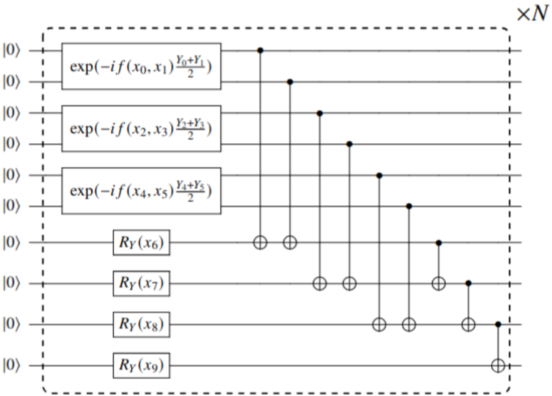
- Baseline model as starting point: the hardware-efficient ansatz (**HEA**)
- Use Pauli twirling to progressively symmetrise the HEA to obtain model variants

HEA and Its Variants



- **HEA-SYMM-ROT-ENT**: symmetrized rotations and entanglement for qubit pairs that carry jet-specific information: 1 & 2, 3 & 4, and 5 & 6.
- Rest of the encode carry features that are permutation invariant (such as invariant mass or $\Delta\eta$)

HEA and Its Variants



- **HEA-FULLSYMM**: same as HEA-SYMM-ROT-ENT (the one before) but wihtout the initial RX embedding layer.

Bivariate Permutation-Invariant Fidelity Quantum Kernel

Definition (**BPINVFQK**)

Let κ be an embedding QK on $\mathcal{X}$ and S_n a feature-permutation group. Then

$$\kappa(\mathbf{x}, \mathbf{x}') = \kappa(\pi(\mathbf{x}), \pi'(\mathbf{x}')), \quad \forall\, \pi, \pi' \in S_n$$

is fully permutation-invariant embedding QK.

Building on the overlap test, define

$$\xi(\mathbf{x}, \mathbf{x}') = \mathrm{Tr}\big[\, U^\dagger(\mathbf{x}')U(\mathbf{x})\, \rho_0\, U^\dagger(\mathbf{x})U(\mathbf{x}')\, (I \otimes |0\rangle\langle 0|)\, \big],$$

then symmetrize:

$$\kappa(\mathbf{x}, \mathbf{x}') = \tfrac{1}{2}\big[\xi(\mathbf{x}, \mathbf{x}') + \xi(\mathbf{x}', \mathbf{x})\big].$$

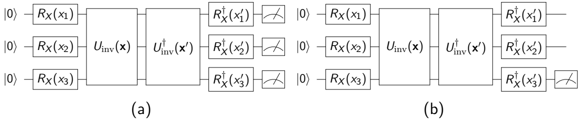
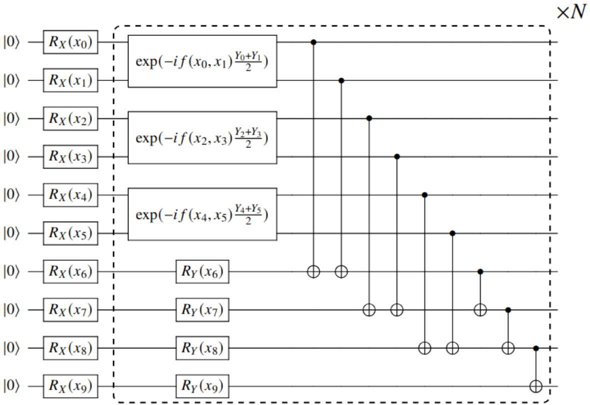


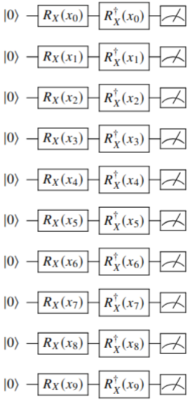
Figure: (a) Full overlap test; (b) BPINVFQK local measurement on qubit 3.

BPINVFQK



- **BPINVFQK** uses the same feature map as HEA-SYMM-ENT-ROT map but with a local measurement at the end of the circuit.

RX Kernel - Sanity Check Benchmark



- **RX**: Quantum baseline model without entanglement included as a sanity check as inspired by[9].

[9] Bowles, Ahmed, and Schuld, *Better than classical? The subtle art of benchmarking quantum machine learning models*.

Model Types

Model	**Description**
<u>HEA</u>	Baseline hardware-efficient ansatz.
<u>RX</u>	Pauli-X angle embedding (no entanglement) benchmark.
<u>RBF</u>	Classical Gaussian (RBF) kernel.
HEA-FULLSYMM	Fully permutation-invariant HEA (HEA-SYMM-ENT-ROT without the initial RX embedding). No direct functional dependence on all of the features.
BPINVFQK	Bivariate permutation-invariant fidelity quantum kernel via local measurement. Direct functional dependence on all features.
HEA-SYMM-ROT	HEA with symmetrized single-qubit rotations.
HEA-SYMM-ENT	HEA with symmetrized entanglement.
HEA-SYMM-ENT-ROT	HEA combining symmetrized entanglement and rotations.

- Underlined models are “<u>baseline</u>”.
- Bolded models are **fully, bivariately permutation-invariant**.
- The rest are partially symmetrized.

Experimental Setup (1/2)

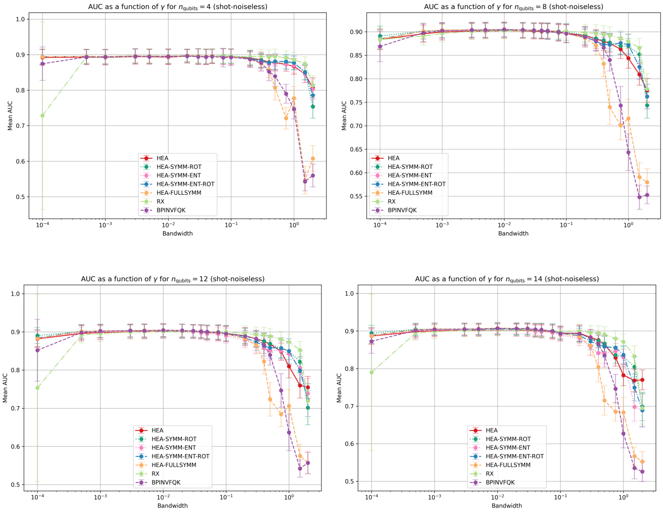
- Dataset: 2000 train + 2000 test samples, balanced signal/background, for $n_{\text{qubits}} = \{4, 6, 8, 10, 12, 14\}$
- Feature normalization: set mean to 0, scale to unit variance, multiply by γ
- Remove implicit "lead/trail" ordering by randomizing jet-pair order
- Hyperparameter sweep per qubit-count:
  - 20 values of γ
  - multiple λ for L_1 regularization
  - Average performance over 25 seeds

Experimental Setup (2/2)

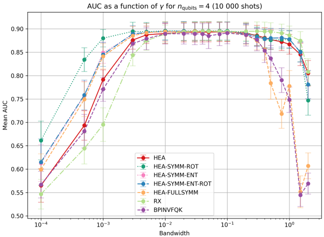
- Execution backend configurations:
  1. Ideal shot-noiseless simulation (PennyLane `lightning.qubit`)
  2. Ideal + shot noise (10000 shots; same backend)
  3. Q50 hardware (10000 shots)
- Added runs for statistical significance: additional 100 seeds at best γ for $n = 10, 12, 14$

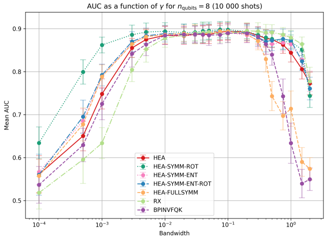
$\rightarrow$ 50B+ circuits

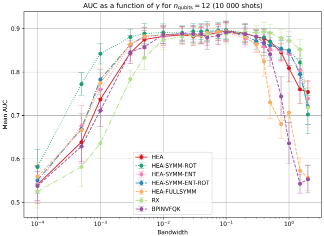
Results - Shot-noiseless Simulation (AUC)

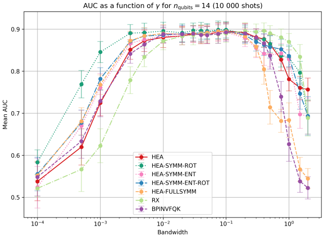


Results - Shot-noisy Simulation (AUC)

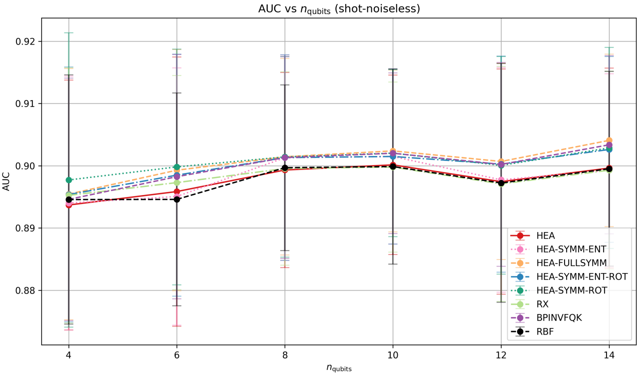






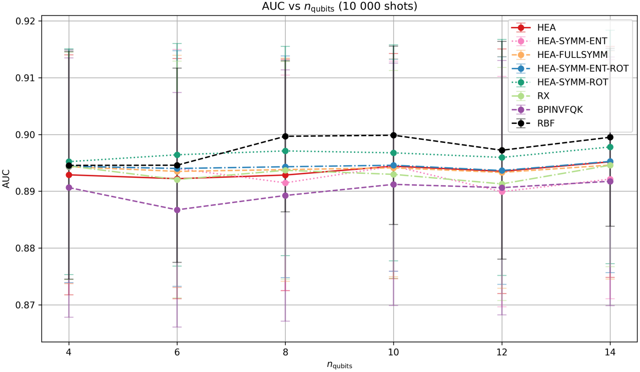


AUC vs. Number of Qubits (Shot-noiseless)



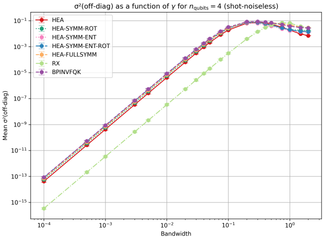
- Symmetrized models show statistically significant AUC gains over RBF, HEA, and RX baselines.
- Variance remains high, despite mean improvements.

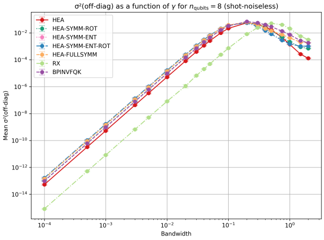
AUC vs. Number of Qubits (10 000 Shots)

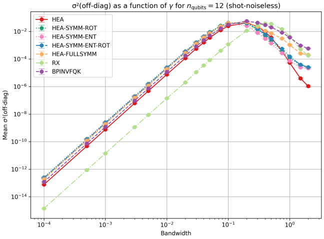


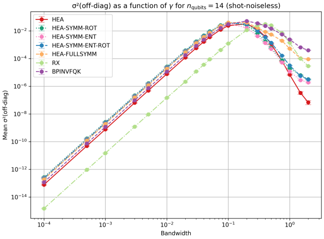
- Shot noise largely eliminates the AUC gains of symmetrized models.
- Classical RBF strives!

Results - Shot-noiseless Simulation (Variance)

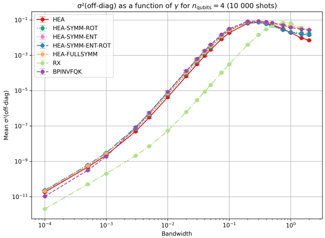


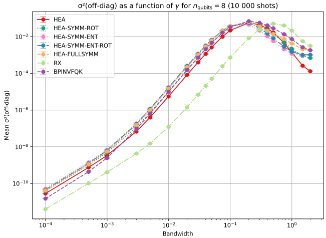


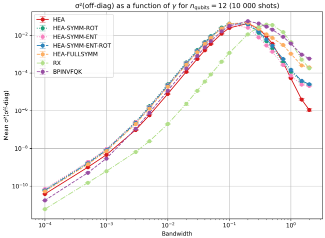


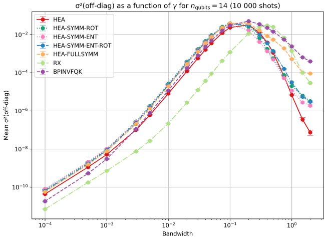


Results - Shot-noisy Simulation (Variance)







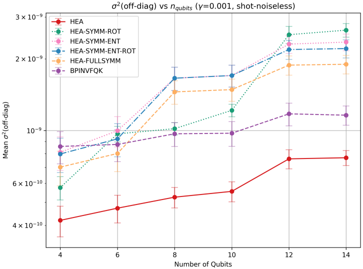


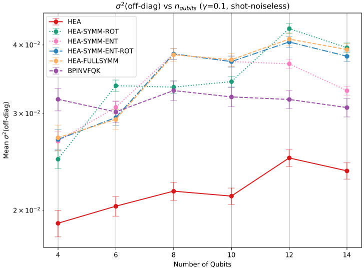
Effect of Shot Noise on γ

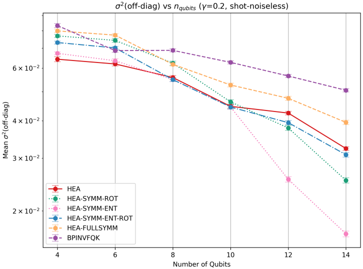
n_{qubits}	**Model**	γ (noiseless)	γ (10 000 shots)	**Relative Diff.**
10	HEA	0.030	0.100	**2.33**
	HEA-SYMM-ENT	0.030	0.100	**2.33**
	HEA-SYMM-ROT	0.030	0.075	**1.50**
	HEA-SYMM-ENT-ROT	0.030	0.100	**2.33**
	HEA-FULLSYMM	0.030	0.100	**2.33**
	BPINVFQK	0.030	0.100	**2.33**
12	HEA	0.030	0.100	**2.33**
	HEA-SYMM-ENT	0.0005	0.100	**199.00**
	HEA-SYMM-ROT	0.010	0.100	**9.00**
	HEA-SYMM-ENT-ROT	0.0005	0.100	**199.00**
	HEA-FULLSYMM	0.010	0.100	**9.00**
	BPINVFQK	0.010	0.100	**9.00**
14	HEA	0.030	0.100	**2.33**
	HEA-SYMM-ENT	0.010	0.075	**6.50**
	HEA-SYMM-ROT	0.0005	0.075	**149.00**
	HEA-SYMM-ENT-ROT	0.030	0.075	**1.50**
	HEA-FULLSYMM	0.030	0.075	**1.50**
	BPINVFQK	0.030	0.100	**2.33**

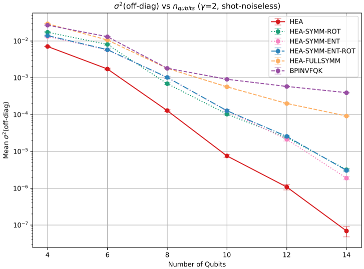
- Optimal kernel bandwidth (in terms of AUC) with and without shot noise as well as the relative difference.

Concentration Behavior for Some Values of γ

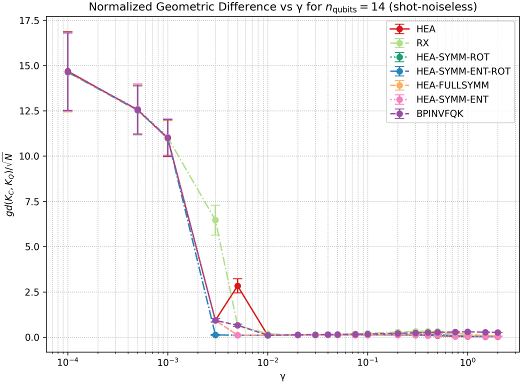








Geometric Difference vs. Bandwidth ($n_{\text{qubits}} = 14$)



- Geometric difference (normalized) as a function of γ at $n_{\text{qubits}} = 14$
- GD grows as γ decreases, i.e., as potential for classical simulability increases

Q50 Experimental Gram Matrix Results

- 30 data points, $n_{\mathrm{qubits}} = 10$, 2-layer HEA vs. HEA-SYMM-ENT-ROT
- 465 circuits per model (optimized pulses, dynamical decoupling, Bayesian readout mitigation[10])

Model	Condition	MSE	Median SE
HEA	Unmitigated	0.086	0.086
	Mitigated	**0.041**	0.038
HEA-SYMM-ENT-ROT	Unmitigated	0.106	0.101
	Mitigated	**0.058**	0.048

[10] Cosco, Plastina, and Gullo, *Bayesian mitigation of measurement errors in multi-qubit experiments*.

Discussion

- **Symmetry improvements (shot-noiseless):** BPINVFQK, HEA-FULLSYMM, HEA-SYMM-ROT achieve:

Discussion

- **Symmetry improvements (shot-noiseless):** BPINVFQK, HEA-FULLSYMM, HEA-SYMM-ROT achieve:
  - Better AUC compared to HEA, RBF, and the RX

Discussion

- **Symmetry improvements (shot-noiseless):** BPINVFQK, HEA-FULLSYMM, HEA-SYMM-ROT achieve:
  - Better AUC compared to HEA, RBF, and the RX
  - Better scalability compared to HEA

Discussion

- **Symmetry improvements (shot-noiseless):** BPINVFQK, HEA-FULLSYMM, HEA-SYMM-ROT achieve:
  - Better AUC compared to HEA, RBF, and the RX
  - Better scalability compared to HEA
  - BPINVFQK performs well in terms of AUC and scalability

Discussion

- **Symmetry improvements (shot-noiseless):** BPINVFQK, HEA-FULLSYMM, HEA-SYMM-ROT achieve:
  - Better AUC compared to HEA, RBF, and the RX
  - Better scalability compared to HEA
  - BPINVFQK performs well in terms of AUC and scalability
- **Effect of shot-noise:** At 10 000 shots, AUC gains vanish: γ increases from optimal, obscuring the benefits

Discussion

- **Symmetry improvements (shot-noiseless):** BPINVFQK, HEA-FULLSYMM, HEA-SYMM-ROT achieve:
  - Better AUC compared to HEA, RBF, and the RX
  - Better scalability compared to HEA
  - BPINVFQK performs well in terms of AUC and scalability
- **Effect of shot-noise:** At 10 000 shots, AUC gains vanish: γ increases from optimal, obscuring the benefits
- γ **and concentration:** $\gamma \lesssim 0.1$ anti-concentrates; $\gamma \gtrsim 0.2$ HEA variants concentrate exponentially, BPINVFQK scales the best at higher γ

Discussion

- **Symmetry improvements (shot-noiseless):** BPINVFQK, HEA-FULLSYMM, HEA-SYMM-ROT achieve:
  - Better AUC compared to HEA, RBF, and the RX
  - Better scalability compared to HEA
  - BPINVFQK performs well in terms of AUC and scalability
- **Effect of shot-noise:** At 10 000 shots, AUC gains vanish: γ increases from optimal, obscuring the benefits
- γ **and concentration:** $\gamma \lesssim 0.1$ anti-concentrates; $\gamma \gtrsim 0.2$ HEA variants concentrate exponentially, BPINVFQK scales the best at higher γ
- **Geometric difference:** Smaller γ increases separation from RBF while increasing classical simulability

Discussion

- **Symmetry improvements (shot-noiseless):** BPINVFQK, HEA-FULLSYMM, HEA-SYMM-ROT achieve:
  - Better AUC compared to HEA, RBF, and the RX
  - Better scalability compared to HEA
  - BPINVFQK performs well in terms of AUC and scalability
- **Effect of shot-noise:** At 10 000 shots, AUC gains vanish: γ increases from optimal, obscuring the benefits
- γ **and concentration:** $\gamma \lesssim 0.1$ anti-concentrates; $\gamma \gtrsim 0.2$ HEA variants concentrate exponentially, BPINVFQK scales the best at higher γ
- **Geometric difference:** Smaller γ increases separation from RBF while increasing classical simulability
- **Hardware results:** Q50 experiments show noticeably higher MSE for symmetrized circuits

Discussion

- **Symmetry improvements (shot-noiseless):** BPINVFQK, HEA-FULLSYMM, HEA-SYMM-ROT achieve:
  - Better AUC compared to HEA, RBF, and the RX
  - Better scalability compared to HEA
  - BPINVFQK performs well in terms of AUC and scalability
- **Effect of shot-noise:** At 10 000 shots, AUC gains vanish: γ increases from optimal, obscuring the benefits
- γ **and concentration:** $\gamma \lesssim 0.1$ anti-concentrates; $\gamma \gtrsim 0.2$ HEA variants concentrate exponentially, BPINVFQK scales the best at higher γ
- **Geometric difference:** Smaller γ increases separation from RBF while increasing classical simulability
- **Hardware results:** Q50 experiments show noticeably higher MSE for symmetrized circuits
- **Outlook:** Classical simulability of the models, noise-resilient symmetry methods, shot-count scaling; symmetries remain a viable method of introducing inductive bias and improving scalability of the models

Take-home message

- Shot count matters in kernels

Take-home message

- **Shot count matters in kernels**
- **Optimizing γ matters a lot in terms of model performance; implications for classical simulability and minimum required shot count**

Take-home message

- **Shot count matters in kernels**
- **Optimizing γ matters a lot in terms of model performance; implications for classical simulability and minimum required shot count**
- **Incorporating symmetries (and/or using the BPINVFQK scheme) may help fight the concentration problem at least at higher γ**

Take-home message

- Shot count matters in kernels
- Optimizing γ matters a lot in terms of model performance; implications for classical simulability and minimum required shot count
- Incorporating symmetries (and/or using the BPINVFQK scheme) may help fight the concentration problem at least at higher γ

Take-home message

- **Shot count matters in kernels**
- **Optimizing γ matters a lot in terms of model performance; implications for classical simulability and minimum required shot count**
- **Incorporating symmetries (and/or using the BPINVFQK scheme) may help fight the concentration problem at least at higher γ**

https://urn.fi/URN:NBN:fi:aalto-202506174894

References

[1] Riccardo Molteni, Casper Gyurik, and Vedran Dunjko. "Exponential quantum advantages in learning quantum observables from classical data". In: *arXiv preprint arXiv:2405.02027v2* (2024). DOI: 10.48550/arXiv.2405.02027. arXiv: 2405.02027v2 [quant-ph].

[2] Yunchao Liu, Srinivasan Arunachalam, and Kristan Temme. "A rigorous and robust quantum speed-up in supervised machine learning". In: *Nature Physics* 17.9 (Sept. 2021), pp. 1013–1017. ISSN: 1745-2481. DOI: 10.1038/s41567-021-01287-z. URL: https://doi.org/10.1038/s41567-021-01287-z.

References (cont.)

[3] Jacob Biamonte et al. "Quantum machine learning". In: *Nature* 549.7671 (Sept. 2017), pp. 195–202. ISSN: 1476-4687. DOI: 10.1038/nature23474. URL: https://doi.org/10.1038/nature23474.

[4] Charles A. Micchelli, Yuesheng Xu, and Haizhang Zhang. "Universal Kernels". In: *Journal of Machine Learning Research* 7.95 (2006), pp. 2651–2667. URL: http://jmlr.org/papers/v7/micchelli06a.html.

[5] Supanut Thanasilp et al. "Exponential concentration in quantum kernel methods". In: *Nature Communications* 15.1 (June 2024), p. 5200. ISSN: 2041-1723. DOI: 10.1038/s41467-024-49287-w. URL: https://doi.org/10.1038/s41467-024-49287-w.

References (cont.)

[6] Ruslan Shaydulin and Stefan M. Wild. "Importance of kernel bandwidth in quantum machine learning". In: *Phys. Rev. A* 106 (4 Oct. 2022), p. 042407. DOI: 10.1103/PhysRevA.106.042407. URL: https://link.aps.org/doi/10.1103/PhysRevA.106.042407.

[7] Behrooz Tahmasebi and Stefanie Jegelka. "The Exact Sample Complexity Gain from Invariances for Kernel Regression". In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh et al. Vol. 36. Curran Associates, Inc., 2023, pp. 55616–55646. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/adf5a38a2e2e7606fbfc3eff72998afa-Paper-Conference.pdf.

References (cont.)

[8] Jure Sokolic et al. "Generalization Error of Invariant Classifiers". In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Ed. by Aarti Singh and Jerry Zhu. Vol. 54. Proceedings of Machine Learning Research. PMLR, 20–22 Apr 2017, pp. 1094–1103. URL: https://proceedings.mlr.press/v54/sokolic17a.html.

[9] Joseph Bowles, Shahnawaz Ahmed, and Maria Schuld. *Better than classical? The subtle art of benchmarking quantum machine learning models*. 2024. arXiv: 2403.07059 [quant-ph]. URL: https://arxiv.org/abs/2403.07059.

References (cont.)

[10] F. Cosco, F. Plastina, and N. Lo Gullo. *Bayesian mitigation of measurement errors in multi-qubit experiments*. 2024. arXiv: 2408.00869 [quant-ph]. URL: https://arxiv.org/abs/2408.00869.

[11] Hsin-Yuan Huang et al. “Power of data in quantum machine learning”. In: *Nature Communications* 12.1 (May 2021), p. 2631. ISSN: 2041-1723. DOI: 10.1038/s41467-021-22539-9. URL: https://doi.org/10.1038/s41467-021-22539-9.

Backup Slides

Geometric Difference

Definition (Geometric Difference)

For two Gram matrices K_C (classical) and K_Q (quantum), and regularization λ,

$$\mathrm{gd}(K_C, K_Q) = \sqrt{\left\| \sqrt{K_C}\,\sqrt{K_Q}\,(K_Q + \lambda I)^{-2}\,\sqrt{K_Q}\,\sqrt{K_C} \right\|_\infty}.$$

[11] Here $\|M\|_\infty = \max_i \sum_j |M_{ij}|$.

- If $\mathrm{gd}(K_C, K_Q) \gtrsim \sqrt{N_{\mathrm{train}}}$, a necessary (though not sufficient) condition for quantum advantage holds.
- Quantifies how "far apart" the quantum and classical feature spaces are.
- Widely used to assess a quantum kernel's potential to outperform classical counterparts.

[11] Huang et al., "Power of data in quantum machine learning".

Statistical Significance ($n_{\text{qubits}} = 10$)

Comparison	t	p	Signif.
HEA vs HEA-SYMM-ROT	2.48	0.022	*
HEA vs HEA-FULLSYMM	8.62	3.7×10^{-14}	***
HEA vs BPINVFQK	6.80	4.8×10^{-10}	***
RBF vs HEA-SYMM-ROT	2.00	0.047	*
RBF vs HEA-FULLSYMM	2.30	0.023	*
RBF vs BPINVFQK	2.02	0.044	*
RX vs HEA-SYMM-ROT	22.63	1.5×10^{-44}	***
RX vs HEA-FULLSYMM	22.73	9.9×10^{-45}	***
RX vs BPINVFQK	22.67	1.3×10^{-44}	***

Table: Paired t-tests and p-values for $n = 10$. Significance codes: * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$.

Statistical Significance ($n_{\mathrm{qubits}} = 12$)

Comparison	t	p	Signif.
HEA vs HEA-SYMM-ROT	2.48	0.022	*
HEA vs HEA-FULLSYMM	2.05	0.043	*
HEA vs BPINVFQK	2.05	0.046	*
RBF vs HEA-SYMM-ROT	2.85	5.2×10^{-4}	***
RBF vs HEA-FULLSYMM	3.95	1.4×10^{-4}	***
RBF vs BPINVFQK	3.58	5.1×10^{-4}	***
RX vs HEA-SYMM-ROT	19.45	3.1×10^{-37}	***
RX vs HEA-FULLSYMM	19.48	2.7×10^{-37}	***
RX vs BPINVFQK	19.44	3.2×10^{-37}	***

Table: Paired t-tests and p-values for $n = 12$. Significance codes: * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$.

Statistical Significance ($n_{\text{qubits}} = 14$)

Comparison	t	p	Signif.
HEA vs HEA-SYMM-ROT	5.05	7.1×10^{-6}	***
HEA vs HEA-FULLSYMM	5.05	7.1×10^{-6}	***
HEA vs BPINVFQK	8.72	1.1×10^{-15}	***
RBF vs HEA-SYMM-ROT	1.99	0.049	*
RBF vs HEA-FULLSYMM	2.50	0.043	*
RBF vs BPINVFQK	3.58	5.1×10^{-4}	***
RX vs HEA-SYMM-ROT	36.43	1.2×10^{-87}	***
RX vs HEA-FULLSYMM	37.37	4.9×10^{-90}	***
RX vs BPINVFQK	37.27	7.5×10^{-90}	***

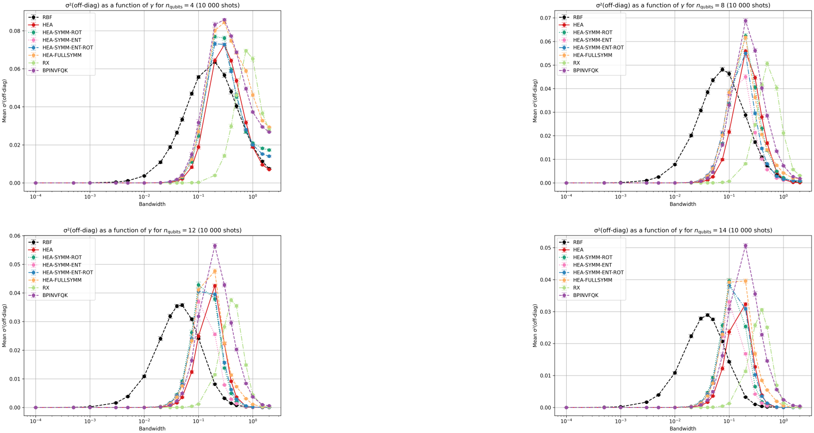
Table: Paired t-tests and p-values for $n = 14$. Significance codes: * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$.

Hyperparameter Ranges

Hyperparameter	Values
$C = 1/\lambda$	10^7, 10^6, 10^5, 10^4, 10^3, 750, 500, 300, 150, 100, 75, 50, 25, 15, 10, 5, 1, 0.75, 0.5, 0.4, 0.3, 0.2, 0.15, 0.1, 0.05, 0.01
γ	0.0001, 0.0005, 0.001, 0.003, 0.005, 0.01, 0.02, 0.03, 0.04, 0.05, 0.075, 0.1, 0.2, 0.3, 0.4, 0.5, 0.75, 1, 1.5, 2

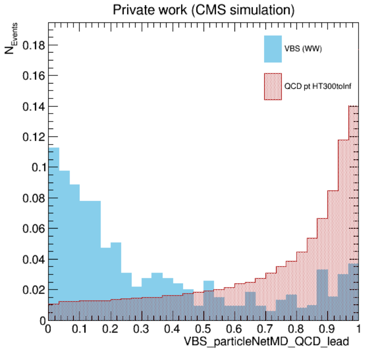
Table: Hyperparameter values used for the simulated results in this work.

Off-Diagonal Variance of the Gram Matrix Elements (Linear Scale)

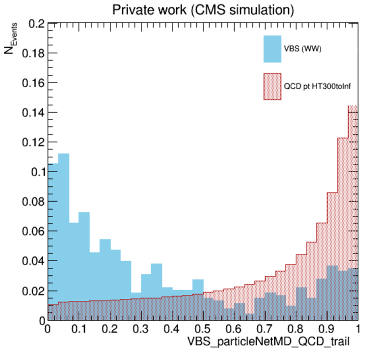


Off-diagonal variance of the models for qubit counts $n_q = 4, 8, 12, 14$ from experiment configuration (2). RBF included for reference.

ParticleNetMD QCD Score Distributions

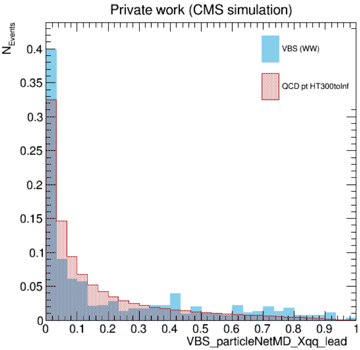


*QCD (lead)

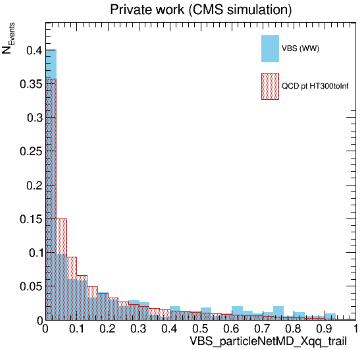


*QCD (trail)

ParticleNetMD Xqq Score Distributions

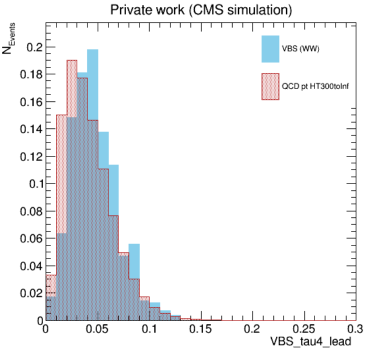


*Xqq (lead)

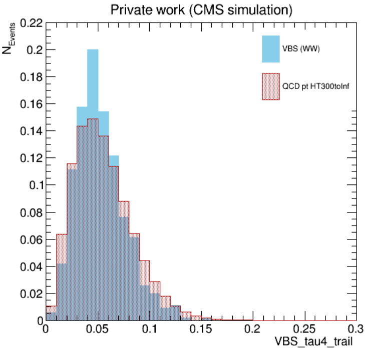


*Xqq (trail)

τ_4 (N-subjettiness)

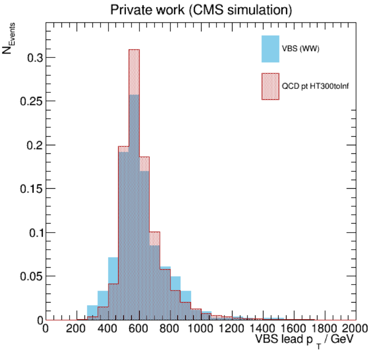


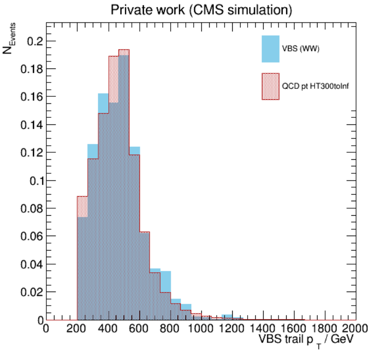
*τ_4 (lead)



*τ_4 (trail)

p_T Distributions

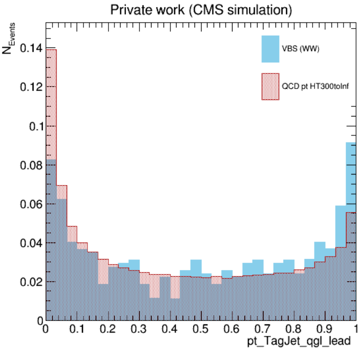




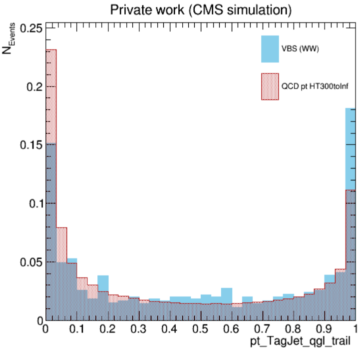
*p_T (lead)

*p_T (trail)

Tag Jet Quark/Gluon Likelihood

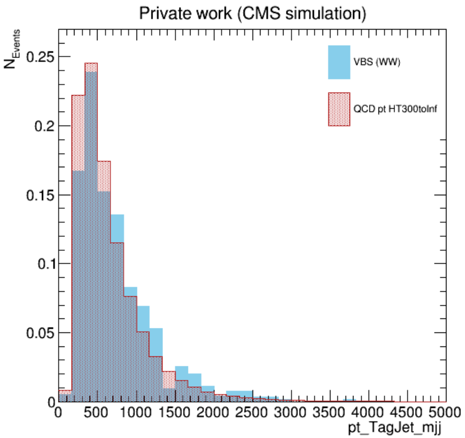


*q/g (lead)



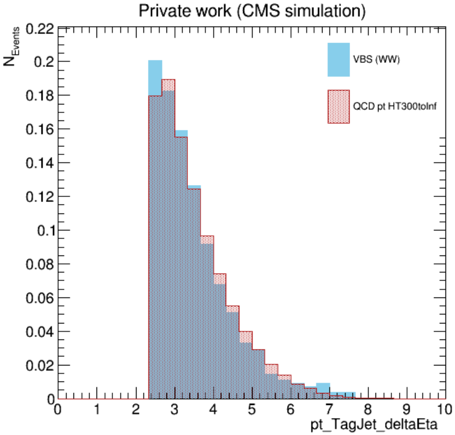
*q/g (trail)

Tag Jet Dijet Mass



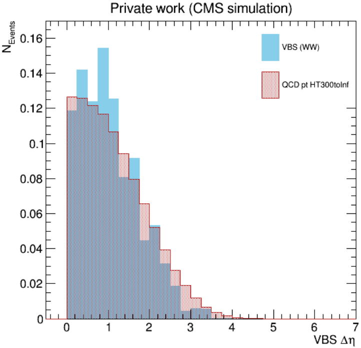
$*m_{jj}$

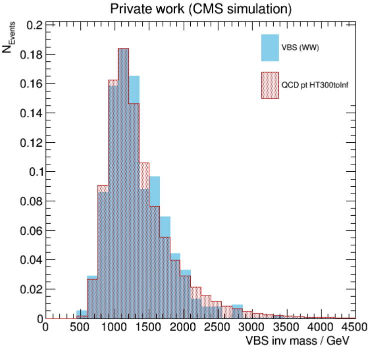
Tag Jet Rapidity Gap



$*\Delta\eta_{jj}$

VBS System Observables





$*\Delta\eta_{\mathrm{VBS}}$ $*m_{VV}$

Model Feature Sets by Numer of Features = n_{qubits}

Model	Features
4 features	VBS_particleNetMD_QCD_lead, VBS_particleNetMD_QCD_trail, pt_TagJet_mjj, VBS_mVV
6 features	VBS_particleNetMD_QCD_lead, VBS_particleNetMD_QCD_trail, VBS_particleNetMD_Xqq_lead, VBS_particleNetMD_Xqq_trail, pt_TagJet_mjj, VBS_mVV
8 features	VBS_particleNetMD_QCD_lead, VBS_particleNetMD_QCD_trail, VBS_particleNetMD_Xqq_lead, VBS_particleNetMD_Xqq_trail, VBS_deltaEta, pt_TagJet_mjj, pt_TagJet_deltaEta, VBS_mVV
10 features	VBS_particleNetMD_QCD_lead, VBS_particleNetMD_QCD_trail, VBS_particleNetMD_Xqq_lead, VBS_particleNetMD_Xqq_trail, VBS_tau4_lead, VBS_tau4_trail, VBS_deltaEta, pt_TagJet_mjj, pt_TagJet_deltaEta, VBS_mVV
12 features	VBS_particleNetMD_QCD_lead, VBS_particleNetMD_QCD_trail, VBS_particleNetMD_Xqq_lead, VBS_particleNetMD_Xqq_trail, VBS_tau4_lead, VBS_tau4_trail, VBS_pt_lead, VBS_pt_trail, VBS_deltaEta, pt_TagJet_mjj, pt_TagJet_deltaEta, VBS_mVV
14 features	VBS_particleNetMD_QCD_lead, VBS_particleNetMD_QCD_trail, VBS_particleNetMD_Xqq_lead, VBS_particleNetMD_Xqq_trail, VBS_tau4_lead, VBS_tau4_trail, VBS_pt_lead, VBS_pt_trail, pt_TagJet_qgl_trail, pt_TagJet_qgl_lead, VBS_deltaEta, pt_TagJet_mjj, pt_TagJet_deltaEta, VBS_mVV