أكاديمية كاوست
KAUST ACADEMY

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

# Quantum Graph Learning for Jet Identification

**Muhannad s Allahibi, Komal Tauqeer**

muhannad.allahibi@kaust.edu.sa, komal.taugeer@cern.ch

## Abstract

At the Large Hadron Collider (LHC), proton collisions produce many short-lived particles that quickly decay. These decays often appear in the detector as collimated sprays of particles called jets. Determining which particle created a jet is crucial for understanding the underlying physics of the collision. Graph Neural Networks (GNNs) represent each particle in a jet as a node and connect nodes with edges based on the kinematic and geometric relationships of nearby particles, allowing the network to capture the jet's detailed substructure. This approach has significantly improved jet identification compared to traditional methods and is now widely used in LHC experiments. With the emergence of quantum algorithms and computing techniques, this project explores whether incorporating quantum-enhanced features into GNNs can further improve jet identification or reveal new insights into jet substructure.

**Date:** August 24, 2026

# Contents


**1 Introduction** **2**

1.1 Related Work . . . 2

**2 Methodology** **2**

2.1 ParticleNet . . . 2

2.2 ParticleNet as a Graph Latent Space . . . 2

2.3 Quantum Evolution Kernel (QEK) . . . 3

2.4 Dataset . . . 4

2.5 Evaluation Procedure . . . 4

**3 Results** **4**

3.1 Baseline: Top-Tagging vs. JetClass-II . . . 4

3.2 Quantum Feature Separation . . . 5

3.3 Isolating the Bottleneck: Latent Space vs. Physics vs. QEK . . . 6

**4 Discussion** **7**

**5 Limitations and Future Work** **8**

**6 Conclusion** **8**

# 1 Introduction

Identifying jets from LHC collisions is critical for new physics searches and precision measurements. Classical Graph Neural Networks lead in this task. This project explores whether quantum-enhanced features can improve jet tagging or reveal new substructure insights.

## 1.1 Related Work

**Chen and Chen** [**chen2025qcgnn**] introduce the Quantum Complete Graph Neural Network (QCGNN), a quantum GNN that models jets as fully connected graphs, reducing aggregation complexity from $O(N^2)$ to $O(N)$. On the Top and JetNet datasets, it reaches AUCs of 0.932 and 0.822 in simulation — competitive with similarly-sized classical models but below top classical taggers. On real IBM Quantum hardware, performance drops to AUC $\approx$ 0.5 due to noise, showing the approach is feasible in simulation but not yet practical on current devices.

**Bal et al.** [**bal2025oneparticle**] propose 1P1Q, which encodes each jet particle's kinematics directly onto its own qubit via the Bloch sphere, skipping classical compression. Using this encoding, a quantum autoencoder beats a classical one with ~80× fewer parameters, and a variational quantum classifier reaches AUC 0.885 on top tagging (vs. 0.898 for Particle Transformer) with only 32 parameters. The QAE is also validated on real CMS data, showing consistent performance with simulation.

# 2 Methodology

## 2.1 ParticleNet

We use ParticleNet qu2020particlenet as the base architecture to build our model, extending it with quantum-enhanced features for jet tagging. Each jet is represented as a point cloud of its constituent particles; ParticleNet builds a dynamic $k$-nearest-neighbor graph over these particles at every layer and aggregates local structure via EdgeConv, enabling it to learn a jet's substructure directly from its constituents rather than from hand-engineered high-level observables.

## 2.2 ParticleNet as a Graph Latent Space

To connect ParticleNet's learned jet representation to the quantum encoding step, we modify its architecture to expose a compact, trained latent space rather than only a final classification score. The standard ParticleNet head — three EdgeConv blocks, global pooling, and a Linear(384→256) projection — is unchanged. We then replace the final classification layer with a narrow bottleneck, Linear(256→16), followed by a small classifier head, Linear(16→2), used only during training.

The network is trained end-to-end as an ordinary binary classifier on jet labels (QCD vs. signal): the classification loss backpropagates through the classifier head, the 16-dimensional bottleneck, and the full EdgeConv backbone. This is a deliberate design choice over the simpler alternative of taking an already-trained 2-class model and replacing its final layer post hoc, which would leave the new layer randomly initialized and produce, at best, a random linear projection of an already-trained representation rather than a representation optimized for the 16-dimensional space itself.

After training, the classifier head is discarded and the network is run in *embedding mode*: the forward pass stops at the 16-dimensional bottleneck, returning the latent vector $z \in \mathbb{R}^{16}$ directly rather than passing it through the classifier. This vector is the graph latent space passed to the quantum encoding step described next.

## 2.3 Quantum Evolution Kernel (QEK)

We use QEK qek, a library developed by Pasqal that extracts machine-learning features from graphs using a neutral-atom quantum device (or emulator). We use Pasqal's platform specifically because it is a neutral-atom architecture: qubit positions are physically programmable, which lets us encode the ParticleNet latent vector directly as physical qubit positions rather than requiring a fixed circuit topology.

### 2.3.1 Angle Encoding: A Custom Compiler

A quantum device only accepts two things: a set of physical atom positions, and a pulse sequence. It has no built-in notion of a 16-dimensional vector, so a custom compiler, `JetclassCompiler`, was built to bridge this gap, implementing QEK's `BaseGraphCompiler` interface.

Each of the 16 latent dimensions is mapped to one atom on a shared circle. Rather than using each latent value directly as an atom's angle, every atom is first given a safe, evenly-spaced base angle (analogous to positions on a clock face), which is then perturbed by a small amount proportional to that dimension's own value:

$$\theta_i = \frac{2\pi i}{16} + \alpha\, z_i, \qquad i = 0, \ldots, 15 \tag{1}$$

where $z_i \in [-\pi, \pi]$ is the $i$-th latent value and $\alpha$ (`NUDGE_SCALE`) controls how strongly the data perturbs each atom's position. We refer to this as *angle encoding*, or equivalently a *perturbed-clock encoding*. All 16 atoms are placed on a circle of one shared radius, scaled so that the closest pair of atoms after perturbation meets the device's minimum atom spacing.

Every atom is connected to every other (full connectivity). This is not merely a simplifying choice: it is the one connectivity rule confirmed, directly from QEK's source code, to always satisfy the library's internal consistency check regardless of how the atoms are arranged, since a fully connected graph has no non-adjacent pairs left to violate it.

### 2.3.2 Calibrating the Perturbation Strength

The parameter $\alpha$ in Equation 1 was tuned empirically, since it directly trades off two competing effects: too small, and the latent values barely influence the physical geometry; too large, and adjacent atoms risk being placed closer together than the device physically allows. An initial value of $\alpha = 0.08$, chosen without accounting for the true range of $z_i$, produced 0 embeddable graphs out of 200 tested jets. After correcting for the actual $[-\pi, \pi]$ range, $\alpha = 0.02$ restored full embeddability (200/200) but showed almost no separation between the quantum features of signal and background jets. The absolute geometric ceiling for $\alpha$, derived directly from the device's minimum atom distance and maximum radial distance, is $\alpha \leq 0.0405$. Table 2 (Section 3.2) summarizes the class separation obtained at three tested values.

We adopt $\alpha = 0.035$ for subsequent work: it retains 200/200 embeddability and recovers ~99.6% of the class separation achieved at the geometric ceiling, while leaving a genuine safety margin against the device's hard constraints.

### 2.3.3 Feature Extraction

The compiled atomic register is evolved under the device's Hamiltonian and measured repeatedly. Each jet's measurement outcomes are summarized as a *distribution of excitation*: the probability, across many simulated shots, of observing $0, 1, 2, \ldots$ excited atoms. From this distribution we extract either (i) a *quantum kernel* value between pairs of jets, $K(\mathcal{G}, \mathcal{G}') = \exp[-JS(\mathcal{P}, \mathcal{P}')]$, using the Jensen–Shannon divergence between their excitation distributions, or (ii) the excitation distribution itself as a quantum feature vector for a single jet.

These outputs are intended to feed into classification in one of two ways: the quantum kernel passed to a Support Vector Machine, or the quantum feature vector fused into ParticleNet's classical inputs. Implementing and comparing both paths is ongoing work.

### 2.4 Dataset

We use JetClass-II jetclass2, a public jet-tagging dataset of approximately 134 million simulated jets across 188 classes. It is split into three parts based on jet origin and substructure:

1. **Res2P**: generic resonance $X \to$ 2-prong decays (quark-antiquark, gluon-gluon, lepton-lepton, or tau-tau), 20M jets.
2. **Res34P**: generic resonance $X \to$ 3- or 4-prong decays, 86M jets, further split into 146 sub-categories by parton/lepton flavor.
3. **QCD**: Standard Model QCD multijet background, 28M jets.

Each jet is stored as a variable-length list of constituent particles (kinematics, particle-flow ID, impact-parameter information) along with jet-level observables such as $p_T$, soft-drop mass, and $N$-subjettiness.

For our signal class we select the qqb sub-category (jet_label 70): a 3-prong decay into two light quarks and one $b$-quark, chosen as the closest structural analog available in JetClass-II to a real top-quark decay $t \to Wb \to (q\bar{q}')b$, though its parent resonance $X$ is generic rather than an authentic top quark. Background is drawn from the QCD partition (jet_label $\geq 161$).

### 2.5 Evaluation Procedure

- **Tools, libraries, and frameworks.** `weaver-benchmark` for ParticleNet training and evaluation (PyTorch backend); Pasqal's `quantum-evolution-kernel` (QEK) and `pulser` for quantum feature extraction, run on the `QutipExtractor` emulator backend; scikit-learn for the SVM kernel classification path.
- **Dataset construction.** JetClass-II jets are downloaded, filtered to the qqb signal category and QCD background, balanced 50/50, and split 60/20/20 into train/validation/test sets. The held-out test set is used only for final reported metrics, never for training or checkpoint selection.
- **Metrics.** Classification accuracy and ROC-AUC on the held-out test set, computed identically for the top-tagging baseline and the JetClass-II analog to ensure a fair comparison.

## 3 Results

### 3.1 Baseline: Top-Tagging vs. JetClass-II

We first establish whether ParticleNet trained on our JetClass-II qqb-vs-QCD substructure analog reaches performance comparable to the same architecture trained on the real top-tagging benchmark. Both models were trained with identical hyperparameters (Ranger optimizer, batch size 256, 20 epochs) and evaluated on a held-out test set never used for training or checkpoint selection.

**Table 1** ParticleNet performance: real top-tagging benchmark vs. the JetClass-II qqb-vs-QCD analog. Top-tagging (real top vs. QCD) serves as the baseline against which the JetClass-II result is judged.

| Metric | Top-tagging (baseline) | JetClass-II |
|---|---|---|
| ROC-AUC | 0.984 | 0.979 |
| Best val. accuracy | 93.6% | 92.98% |

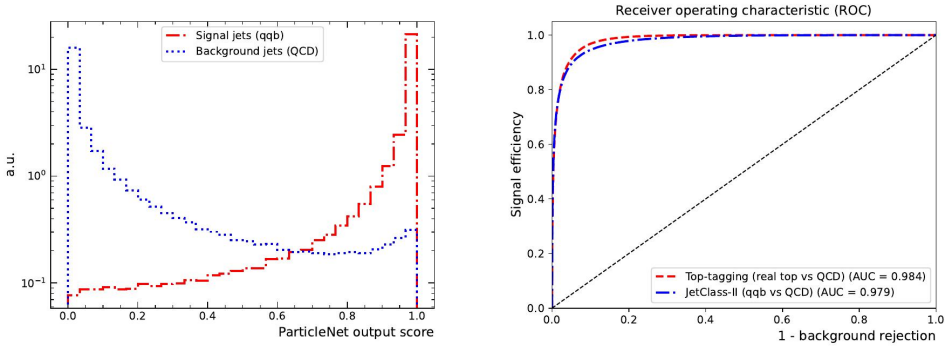



**Figure 1** Left: ParticleNet output score distribution for JetClass-II signal (qqb) and background (QCD) jets. Right: ROC curves comparing the real top-tagging benchmark against the JetClass-II analog.

### 3.2 Quantum Feature Separation

Having established a working, embeddable encoding (Section 2.3), we tested whether the resulting quantum feature actually distinguishes signal from background jets. For 200 real jets (97 signal, 103 QCD), we computed each jet's excitation distribution and averaged it within each class, then measured the Jensen–Shannon divergence between the two class-averaged distributions — the same metric used by the quantum kernel itself.

**Table 2** Class separation (signal vs. QCD) in the quantum feature, as a function of the perturbation strength $\alpha$ in Equation 1. JS divergence is reported as a fraction of its theoretical maximum, $\ln 2$.

| $\alpha$ | Embeddable | JS divergence | % of max |
|---|---|---|---|
| 0.02 | 200/200 | 0.0014 | 0.2% |
| 0.035 (adopted) | 200/200 | 0.0720 | 10.4% |
| 0.0405 (geometric ceiling) | 200/200 | 0.0723 | 10.4% |

Increasing $\alpha$ from 0.02 to 0.035 produced a 53-fold increase in class separation, but pushing further to the absolute geometric ceiling (0.0405) gained almost nothing more — separation plateaus well before the device's hard physical limit is reached. This is a genuine finding about the encoding itself, not merely a tuning artifact: at $\alpha = 0.035$, signal jets show a higher mean excitation count (6.33 excited atoms, averaged) than QCD jets (4.70), a systematic difference rather than random per-jet noise (Figure 2).

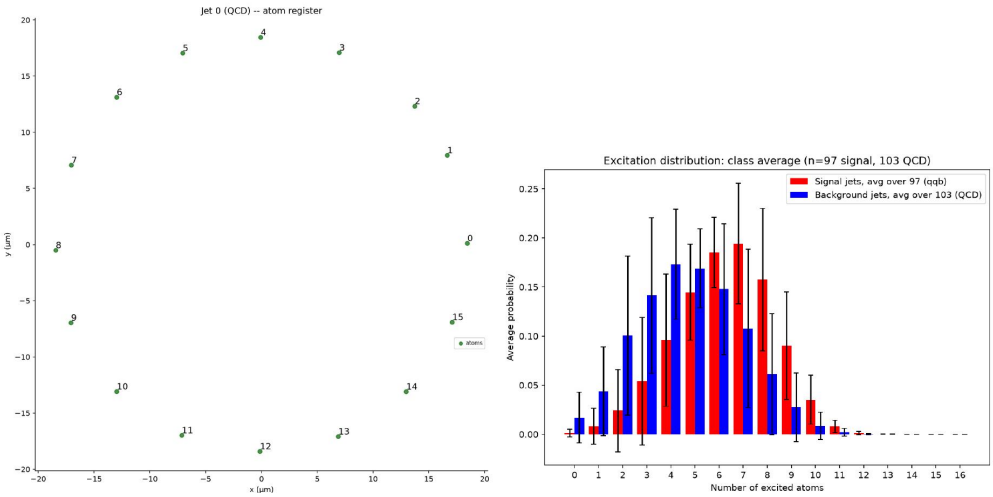



**Figure 2** Left: the compiled atomic register for one example jet (QCD class) under angle encoding at $\alpha = 0.035$ — 16 atoms placed on a shared circle, each perturbed slightly from its evenly-spaced base position by its own latent value. Right: the excitation distribution averaged within each class (97 signal, 103 QCD jets), with error bars showing the standard deviation across jets within each class. Signal jets show a visibly higher average excitation count than QCD jets.

### 3.3 Isolating the Bottleneck: Latent Space vs. Physics vs. QEK

The quantum feature separation in Section 3.2 is real but modest. To find out whether this modesty comes from ParticleNet's classical representation or from the angle-encoding step that turns it into a quantum feature, three feature sets were compared with an identical downstream classifier (logistic regression, 5-fold stratified cross-validation, same 200 jets): the raw 16-D ParticleNet latent vector, four hand-crafted physics variables (soft-drop mass, $N$-subjettiness ratios, particle multiplicity), and the raw QEK excitation output alone.

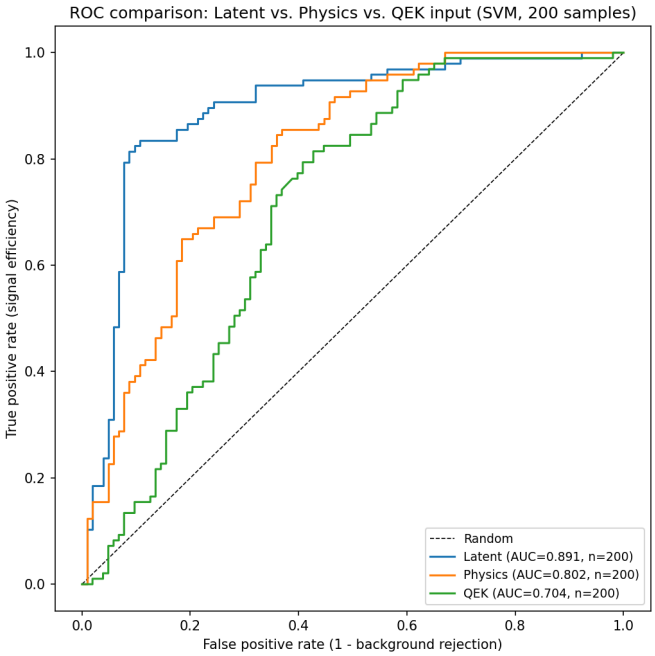



**Figure 3** ROC comparison, 5-fold CV, same 200 jets: the ParticleNet latent space (AUC 0.881) clearly outperforms both hand-crafted physics variables (0.795) and the raw QEK output alone (0.702).

The latent space alone reaches AUC 0.881 — comfortably ahead of the hand-crafted physics variables, and well ahead of the raw quantum feature. Figure 4 shows why directly: projecting the 16-D latent space of 1000 held-out jets (464 signal, 536 background) to 2D via t-SNE reveals visible, systematic class separation already present in ParticleNet's classical output, before any quantum encoding is applied.

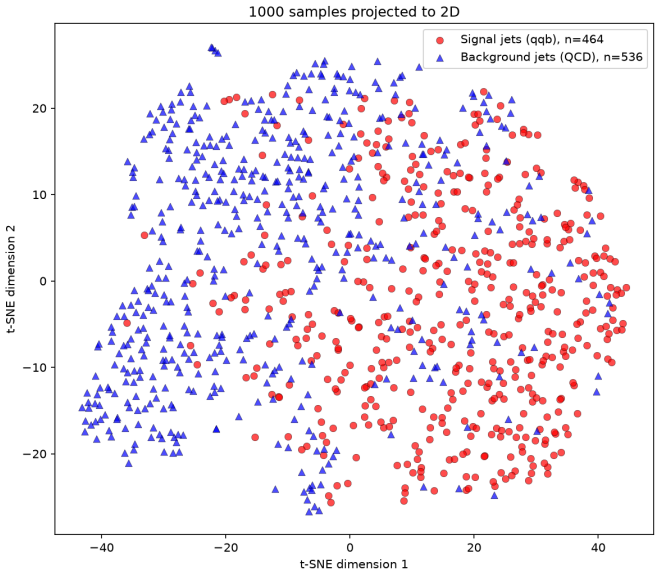



**Figure 4** Per-jet t-SNE of the 16-D ParticleNet latent space, 1000 held-out jets (464 signal, 536 background). Clear, though imperfect, class separation is visible directly in the classical representation.

Taken together, these two results isolate where the pipeline's limitation actually sits. ParticleNet's latent space is well separated on its own (AUC 0.881, visible t-SNE clustering); it is only after the angle-encoding compiler translates that space into physical qubit positions that separation drops to AUC 0.702 for the quantum feature alone, and to 10.4% of the theoretical maximum for the JS-divergence metric in Section 3.2. The discriminative information is demonstrably present in the classical representation — the bottleneck is the encoding step, not ParticleNet or the dataset.

## 4 Discussion

The three results in Section 3 point in the same direction: the classical and quantum halves of the pipeline each behave as a physically reasonable model should, but with very different headroom.

The classical baseline is close to a best case. ParticleNet trained on the JetClass-II qqb-vs-QCD analog reaches 0.979 ROC-AUC against 0.984 for the real top-tagging benchmark, and 92.98% validation accuracy against 93.6% — a gap of well under one point on both metrics (Table 1). Since qqb is a 3-prong decay through a generic resonance rather than an authentic top quark, this closeness was not guaranteed; it indicates the qqb substructure is a fair stand-in for top-like decays, and that routing the classifier through a narrow 16-dimensional bottleneck (Section 2.2) costs essentially no classical performance. This validates the bottleneck as a low-risk interface between the classical and quantum stages.

The quantum feature separation tells a more modest story. At the adopted operating point ($\alpha = 0.035$), the Jensen–Shannon divergence between signal and background excitation distributions reaches only 10.4% of its theoretical maximum (Table 2) — real and systematic (Figure 2 shows signal jets excite more atoms on average than background jets, not just noisier), but

far from saturating the metric. Section 3.3 shows this modesty is not a property of the underlying representation: the same latent space that produces only a 10.4% JS-divergence separation once encoded reaches AUC 0.881 on its own, well ahead of the raw quantum feature's 0.702. The gap opens specifically at the encoding step — 16 latent dimensions mapped onto fixed positions on a single circle, with only a small perturbation $\alpha z_i$ available to carry class-discriminative information before the geometry becomes invalid. The encoding was never designed to extract every bit of separation the latent space contains, only to translate it faithfully into physical qubit positions, and Section 3.3 confirms it is currently the limiting factor rather than ParticleNet or the qqb/QCD split itself.

The most unexpected finding is how the separation responds to $\alpha$. Going from $\alpha = 0.02$ to $\alpha = 0.035$ produced a 53-fold increase in JS divergence, yet pushing further to the geometric ceiling ($\alpha = 0.0405$) gained almost nothing more (10.4% → 10.4%, to three significant figures). This is not a smooth, proportional response to $\alpha$: separation is suppressed almost entirely below some threshold and then plateaus almost immediately once the perturbation is large enough to matter, well inside the device's hard physical limit. That plateau is itself useful information about this specific encoding's expressivity, independent of how the underlying latent space was trained.

## 5 Limitations and Future Work

- **Small evaluation sample.** Quantum feature separation was measured on only 200 jets. 16-qubit simulation is memory-intensive, and an initial attempt on a shared interactive node required moving to dedicated batch computation. *Next step:* scale to a larger sample now that per-jet runtime and memory requirements are characterized.
- **SVM kernel and fusion paths are not yet implemented.** Section 2.3.3 describes two intended downstream uses of the quantum feature — a kernel passed to an SVM, or a feature vector fused into ParticleNet's classical inputs. The logistic-regression comparison in Section 3.3 was used only as a diagnostic to isolate where the pipeline's bottleneck sits, not as either of these two paths. *Next step:* implement and compare both paths against the classical baseline in Table 1.
- **Only one encoding scheme tested.** Angle encoding has a narrow safe margin ($\alpha \leq 0.0405$) before risking an invalid atom configuration. *Next step:* implement and compare a radial-encoding alternative (fixed angle, latent value on the radial axis instead), which has a structurally different, and potentially larger, safe parameter range.
- **Bottleneck trained on classification loss alone.** The 16-dimensional latent space is optimized only to help ParticleNet classify jets, with no explicit incentive to preserve information that classification does not need but the quantum encoding might benefit from. *Next step:* explore an autoencoder-style bottleneck with an added reconstruction loss, to test whether a more general-purpose latent space improves downstream quantum separation.

## 6 Conclusion

This project built a pipeline from raw LHC jet data through a classical Graph Neural Network, a trained 16-dimensional latent bottleneck, and a custom compiler translating that bottleneck into physical input for a neutral-atom quantum device. The classical half of the pipeline is validated: ParticleNet on the JetClass-II qqb-vs-QCD analog matches the real top-tagging benchmark to within a point on both ROC-AUC and accuracy, confirming that the bottleneck does not cost meaningful classical performance. The quantum half is working and calibrated: after correcting

an early calibration error and tuning the perturbation strength $\alpha$ against the device's physical limits, the resulting quantum feature shows a real, systematic, quantified separation between signal and background jets, though this separation is modest relative to the metric's theoretical maximum and plateaus well before the device's geometric ceiling. Critically, a direct comparison (Section 3.3) shows this modesty is not inherited from ParticleNet or the dataset split: the same latent space reaches AUC 0.881 on its own, well ahead of the raw quantum feature's 0.702, so the bottleneck is demonstrably the angle-encoding step rather than the classical representation feeding it. The main takeaway is that the pipeline itself is sound end-to-end, and effort is best spent on a richer encoding scheme; the clear next step is to close the loop by feeding the quantum feature into a full classifier (SVM kernel or fusion with ParticleNet) and testing whether it adds discriminative power beyond the already-strong classical baseline.