

Quantum Vision Transformers for Quark–Gluon Classification

Marçal Comajoan Cara ¹, Gopal Ramesh Dahale ^{2,†}, Zhongtian Dong ^{3,†}, Roy T. Forestano ^{4,†},
Sergei Gleyzer ^{5,†}, Daniel Justice ^{6,†}, Kyoungchul Kong ^{3,†}, Tom Magorsch ^{7,†}, Konstantin T. Matchev ^{4,*,†},
Katia Matcheva ^{4,†} and Eyup B. Unlu ^{4,†}

¹ Department of Signal Theory and Communications, Polytechnic University of Catalonia, 08034 Barcelona, Spain; marcal.comajoan@estudiantat.upc.edu

² Indian Institute of Technology Bhilai, Bhilai 491001, Chhattisgarh, India; gopald@iitbhillai.ac.in

³ Department of Physics and Astronomy, University of Kansas, Lawrence, KS 66045, USA; cdong@ku.edu (Z.D.); kckong@ku.edu (K.K.)

⁴ Institute for Fundamental Theory, Physics Department, University of Florida, Gainesville, FL 32611, USA; roy.forestano@ufl.edu (R.T.F.); matcheva@ufl.edu (K.M.); eyup.unlu@ufl.edu (E.B.U.)

⁵ Department of Physics and Astronomy, University of Alabama, Tuscaloosa, AL 35401, USA; sgleyzer@ua.edu

⁶ Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA; dljustice@sei.cmu.edu

⁷ Physik-Department, Technische Universität München, 85748 Garching, Germany; tom.magorsch@tum.de

* Correspondence: matchev@ufl.edu

† These authors contributed equally to this work.

Abstract: We introduce a hybrid quantum-classical vision transformer architecture, notable for its integration of variational quantum circuits within both the attention mechanism and the multi-layer perceptrons. The research addresses the critical challenge of computational efficiency and resource constraints in analyzing data from the upcoming High Luminosity Large Hadron Collider, presenting the architecture as a potential solution. In particular, we evaluate our method by applying the model to multi-detector jet images from CMS Open Data. The goal is to distinguish quark-initiated from gluon-initiated jets. We successfully train the quantum model and evaluate it via numerical simulations. Using this approach, we achieve classification performance almost on par with the one obtained with the completely classical architecture, considering a similar number of parameters.

Keywords: quantum computing; deep learning; quantum machine learning; classical-quantum neural networks; vision transformers; supervised learning; classification; Large Hadron Collider

MSC: 68Q12; 81P68



Citation: Comajoan Cara, M.; Dahale, G.R.; Dong, Z.; Forestano, R.T.; Gleyzer, S.; Justice, D.; Kong, K.; Magorsch, T.; Matchev, K.T.; Matcheva, K.; et al. Quantum Vision Transformers for Quark–Gluon Classification. *Axioms* **2024**, *13*, 323. <https://doi.org/10.3390/axioms13050323>

Academic Editor: Angel Ricardo Plastino

Received: 25 January 2024

Revised: 3 May 2024

Accepted: 9 May 2024

Published: 13 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The imminent operation of the High Luminosity Large Hadron Collider (HL-LHC) [1] by the end of this decade signals an era of unprecedented data generation, necessitating vast computing resources and advanced computational strategies to effectively manage and analyze the resulting datasets [2]. A promising approach to deal with this huge amount of data could be the application of quantum machine learning (QML), which could reduce the time complexity of classical algorithms by running on quantum computers and obtain better accuracies thanks to the access to the exponentially large Hilbert space [3–11].

The innovative core of our research lies in the development of a novel quantum-classical hybrid vision transformer architecture that integrates variational quantum circuits into the attention mechanisms and multi-layer perceptions of the classical vision transformer (ViT) architecture [12]. More specifically, we adapt the classical ViT architecture to the quantum realm by replacing the classical linear projection layers used in the multi-head attention subroutines by variational quantum circuits (VQCs), as well as by using VQCs in the multi-layer perceptrons too. This approach is based on previous work [13], which proposed the same idea for the original transformer architecture for text [14]. Other works

have explored other possible quantum adaptations of the original transformer [13,15], as well as adaptations of the vision transformer [16,17] and the graph transformer [18]. Our work differs from [16] in the architecture that we propose, which explores the use of other quantum ansatzes. The model in [17] was developed in parallel with this study and differs in several respects—the use of classical multi-layer perceptrons (MLPs) instead of quantum MLPs and the use of different ansatzes for the key, value, and query operations.

We train and evaluate our proposed quantum vision transformer (QViT) on multi-detector jet images from data from the CMS Open Data Portal [19]. The goal is to discriminate between quark-initiated (quark) and gluon-initiated (gluon) jets. This task has broad applicability to searches and measurements at the Large Hadron Collider (LHC) [20]. Consequently, ways of solving this task have already been extensively examined with classical machine learning techniques [20–24].

The motivation behind the application of QML to this particular task stems from the inherent limitations of current classical deep learning models, which, despite their efficacy, are increasingly constrained by escalating computational demands and resource requirements inherent in processing and analyzing large datasets, such as those anticipated from the HL-LHC. Our research endeavors to address these challenges by leveraging the unique capabilities of quantum computing to enhance the efficiency and performance of machine learning models in the context of high-energy physics.

2. Background

2.1. (Classical) Deep Learning, the Transformer, and the Vision Transformer

The field of artificial intelligence aims to replicate in computers the remarkable capabilities of the human brain, such as identifying objects in images, writing text, transcribing and recognizing speech, offering personalized recommendations, and much more. The application of machine learning systems is becoming ubiquitous in many domains of science, technology, business, and government, gradually replacing the use of traditional hand-crafted algorithms. This shift has not only enhanced the efficacy of existing technologies but has also paved the way for an array of novel capabilities that would have been inconceivable otherwise.

Deep learning is a subfield of artificial intelligence that deals with neural networks, a type of computational model that has emerged as an exceptionally powerful and versatile approach to learning from data. The most straightforward realization of a neural network is in a “feedforward” configuration, also known as a multi-layer perceptron (MLP), which can be mathematically described as a composition of elementwise non-linearities with affine transformations of the data [25–27].

In this context, an affine transformation refers to a linear transformation followed by a translation. Given an input vector $x \in \mathbb{R}^{D_1}$, a weight matrix $W \in \mathbb{R}^{D_2 \times D_1}$, and a bias vector $b \in \mathbb{R}^{D_2}$, the affine transformation is defined as

$$a(x) = Wx + b, \quad (1)$$

where $a(x) \in \mathbb{R}^{D_2}$ is the output of the affine transformation.

The elementwise non-linearity, also known as an activation function, is then applied to each component of the output vector a :

$$f(x) = \sigma(a(x)), \quad (2)$$

where σ denotes the activation function. Traditional choices for the activation function include the sigmoid function and the hyperbolic tangent (tanh) function, but these have largely fallen out of favor in modern deep learning architectures. The rectified linear unit (ReLU) [28], defined as

$$\text{ReLU}(x) = \max(0, x), \quad (3)$$

has gained popularity due to its simplicity and effectiveness [29]. More recently, variations of the ReLU have been proposed to further improve the performance and stability of deep learning models, such as the Gaussian Error Linear Unit (GELU) [30], which is defined as

$$\text{GELU}(x) = x\Phi(x), \quad (4)$$

where $\Phi(x)$ is the cumulative distribution function of the standard normal distribution. Another important activation function in deep learning is the softmax function, which is commonly used in the output layer of a neural network for multi-class classification tasks. The softmax function takes a vector of real numbers and transforms it into a probability distribution over the classes. Given an input vector $z \in \mathbb{R}^K$, the softmax function is defined as

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}. \quad (5)$$

The output of the softmax function represents the predicted probabilities for each class, with the highest probability indicating the most likely class.

Deep learning networks are constructed by stacking multiple layers of these transformations:

$$\hat{y} = f_L \circ f_{L-1} \circ \cdots \circ f_1(x), \quad (6)$$

where

$$f_i(x) = \sigma_i(W_i x + b_i). \quad (7)$$

This stacking allows the network to learn increasingly complex representations of the input data. The output of one layer serves as the input to the subsequent layer, forming a hierarchical structure. The final layer of the network produces the desired output, which can be a classification label, a regression value, or any other task-specific output.

The learning process in deep learning involves adjusting the weights and biases of the network to minimize a loss function, which quantifies the discrepancy between the predicted outputs and the expected ones. For classification tasks, a commonly used loss function is the cross-entropy loss, which measures the dissimilarity between the predicted class probabilities and the true class labels. The cross-entropy loss is defined as

$$L = - \sum_{n=1}^N \sum_{k=1}^K y_{nk} \log(\hat{y}_{nk}), \quad (8)$$

where N is the number of samples, K is the number of classes, y_{nk} is the true label (0 or 1) for sample n and class k , and \hat{y}_{nk} is the predicted probability for sample n and class k .

The optimization of the loss function is typically performed using stochastic gradient descent (SGD) or its variants. SGD updates the model parameters using a randomly selected subset of the training data, called a mini-batch, at each iteration. The update rule for SGD is given by

$$\theta_{t+1} = \theta_t - \eta \nabla_{\theta} L_B(\theta_t), \quad (9)$$

where θ_t represents the model parameters at iteration t , η is the learning rate, and $\nabla_{\theta} L_B(\theta_t)$ is the gradient of the loss function with respect to the parameters, estimated using the mini-batch B .

The backpropagation algorithm is typically used to efficiently compute the gradients of the loss function with respect to the model parameters in a neural network. It relies on the chain rule of calculus to propagate the gradients from the output layer to the input layer, enabling the computation of the gradients for each layer in the network.

Apart from the MLP, more advanced neural network architectures have been devised. Among these, the Transformer architecture [14] stands out as a seminal breakthrough in the field of deep learning. The main building block of the Transformer is a layer that takes as input a matrix $X \in \mathbb{R}^{N \times D}$ and outputs a transformed matrix $X' \in \mathbb{R}^{N \times D}$ of the same dimensionality. Each of these layers has two sub-layers: first, a multi-head self-

attention mechanism, the core architectural component of the Transformer, and second, a simple MLP. Moreover, to improve training efficiency, layer normalization [31] and residual connections [32] around each sub-layer are employed. Thus, the resulting transformation is

$$Z = X + \text{LayerNorm}(\text{MHA}(X, X, X)), \tag{10}$$

$$X' = Z + \text{LayerNorm}(\text{MLP}(Z)). \tag{11}$$

The attention mechanism is a key component of the Transformer architecture. It allows the model to focus on specific parts of the input sequence when generating each output element. Given a query matrix $Q \in \mathbb{R}^{N \times D_k}$, a key matrix $K \in \mathbb{R}^{M \times D_k}$, and a value matrix $V \in \mathbb{R}^{M \times D_v}$, the attention function is defined in [14] as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{D_k}}\right)V, \tag{12}$$

where D_k is the dimension of the keys, used as a scaling factor to prevent the dot products from growing too large.

Self-attention is a special case of attention where the query, key, and value matrices are all derived from the same input matrix X . In the Transformer, self-attention allows each position in the input sequence to attend to all positions in the previous layer.

Multi-head attention is an extension of the attention mechanism that allows the model to jointly attend to information from different representation subspaces at different positions. Instead of performing a single attention function, multi-head attention linearly projects the queries, keys, and values h times with different learned linear projections, performs the attention function in parallel, concatenates the results, and projects the concatenated output using another learned linear projection. Mathematically, multi-head attention is defined as

$$\begin{aligned} \text{MHA}(Q, K, V) &= \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O, \\ \text{where head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{13}$$

where $W_i^Q \in \mathbb{R}^{D \times D_k}$, $W_i^K \in \mathbb{R}^{D \times D_k}$, $W_i^V \in \mathbb{R}^{D \times D_v}$, and $W^O \in \mathbb{R}^{hD_v \times D}$ are learnable parameter matrices.

The Transformer architecture, originally designed for natural language processing, has also been adapted to other domains. For instance, its adaptation for computer vision has given rise to the Vision Transformer (ViT) [12]. In ViTs, an image is split into a sequence of patches, which are then linearly embedded and treated as input tokens for a stack of Transformer layers, collectively referred to as the Transformer encoder. The ViT has achieved state-of-the-art performance on various image classification benchmarks, demonstrating the versatility and effectiveness of the Transformer architecture across different domains [33].

2.2. Quantum Computing and Quantum Machine Learning

In quantum computing, the fundamental unit of information is the qubit, which, unlike its classical counterpart, the bit, can exist in a state of superposition to represent non-binary states. The quantum state of n qubits can be represented with a unit vector $|\psi\rangle$ in the Hilbert space \mathbb{C}^{2^n} (in bra-ket notation, the ket $|\ \rangle$ denotes a column vector and the bra $\langle \ |$ a row vector).

A quantum circuit is a series of quantum logic operations (or gates) applied to qubits to change their state. This can be represented mathematically by matrix multiplication, $U|\psi\rangle$, where U is a $2^n \times 2^n$ unitary matrix. Typically, a quantum circuit ends with a measurement of all the qubits, which provides important information about the final state of the circuit.

In this paper, we make use of the R_X gate, which performs a single-qubit rotation about the X axis, and the CNOT gate, which operates over two qubits, by flipping the

second one (the target qubit) if and only if the first one (the control qubit) is $|1\rangle$. They can be represented with the following matrices:

$$R_X(\theta) = \begin{bmatrix} \cos(\theta/2) & -i \sin(\theta/2) \\ -i \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}, \quad (14)$$

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (15)$$

The main idea behind quantum machine learning (QML) is to use models that are partially or fully executed on a quantum computer by replacing some subroutines of the models with quantum circuits in order to exploit the unique properties of quantum mechanics to enhance the capabilities of classical machine learning algorithms. Some notable examples are quantum support vector machines [34], quantum nearest-neighbor algorithms [35], quantum nearest centroid classifiers [36], and quantum artificial neural networks [6,10], including quantum graph neural networks [11]. In the last case, some layers are typically executed on a quantum circuit that has rotation angles that are free parameters of the whole model. These parameters are optimized together with the parameters of the classical layers. Such parametrized quantum circuits are also called variational quantum circuits (VQCs).

2.3. High-Energy Physics and Jets

High-energy physics research aims to understand how our universe works at its most fundamental level. We do this by discovering the most elementary constituents of matter and energy, exploring the basic nature of space and time itself and probing the interactions between them. These fundamental ideas are at the heart of physics and hence all of the physical sciences. Among many other experiments, the LHC provides ubiquitous opportunities for precision measurement of particle properties in the standard model of elementary particle physics, as well as for searching for new physics beyond the standard model. It is not only the largest human-made experiment on Earth but also the most prolific producer of scientific data. The HL-LHC will produce 100-fold to about 1 exabyte per year, bringing quantitatively and qualitatively new challenges due to its event size, data volume, and complexity, therefore straining the available computational resources [37].

In collider experiments, jets arise as a result of the hadronization of the fundamental elementary particles, which carry color charge, namely, the quarks and the gluons. The color confinement phenomenon in quantum chromodynamics implies that quarks and gluons cannot exist in free form but must be converted into a collection of colorless objects (called hadrons) [38]. In high-energy particle collisions like those taking place at the LHC, the initial quarks and gluons are produced with significant boosts (i.e., with large momenta), and therefore, the resulting collections of hadrons appear as narrow collimated bunches, which are generically called jets. There are standard and well-tested jet reconstruction algorithms that identify candidate jets among the myriad of particles observed in the detector [39]. However, the question of the precise origin of a given jet—whether it came from a quark (and which type of quark) or a gluon—is highly non-trivial and to this day continues to be the subject of active investigations in the literature [40–42].

In this paper, we shall focus on the classification task of distinguishing between a jet arising from a light quark and a jet arising from a gluon progenitor particle (see [40–52] and references therein for various classical machine learning methods).

3. Method

3.1. Data

We use the dataset described in Andrews et al. [24], which was derived from simulated data for QCD dijet production available on the CERN CMS Open Data Portal [19]. Events

were generated and hadronized with the PYTHIA6 Monte Carlo event generator using the Z2* tune, which accounts for the difference in the hadronization patterns of quarks and gluons. The dataset consists of 933,206 3-channel 125×125 images, with half representing quarks and the other half gluons. Each of the three channels in the images corresponds to a specific component of the Compact Muon Solenoid (CMS) detector [53]: the inner tracking system (Tracks), which identifies charged particle tracks [54]; the electromagnetic calorimeter (ECAL), which captures energy deposits from electromagnetic particles [55]; and the hadronic calorimeter (HCAL), which detects energy deposits from hadrons [56,57].

In the CMS experiment, the components of the measured momenta of individual particles are represented in a coordinate system oriented as shown in Figure 1 [53]. The origin of the coordinate system is centered at the nominal collision point inside the experiment, the y -axis points vertically up, and the x -axis points radially inward toward the center of the LHC. In order to form a right-handed coordinate system, the z -axis then points along the beam direction toward the Jura mountains from LHC Point 5 (the location of the CMS experiment). The azimuthal angle φ is measured from the x -axis in the (x, y) plane, while the polar angle θ is measured from the z -axis. In particle physics, one often trades the polar angle θ for related kinematic variables like the rapidity y or the closely related pseudorapidity η , which are defined as [37]

$$y \equiv \frac{1}{2} \ln \frac{E + p_z}{E - p_z} \quad (16)$$

and

$$\eta \equiv -\ln \left[\tan \left(\frac{\theta}{2} \right) \right]. \quad (17)$$

Furthermore, the magnitude of the momentum \vec{p}_T transverse to the beam direction is computed from the respective p_x and p_y components as

$$p_T \equiv \sqrt{p_x^2 + p_y^2}. \quad (18)$$

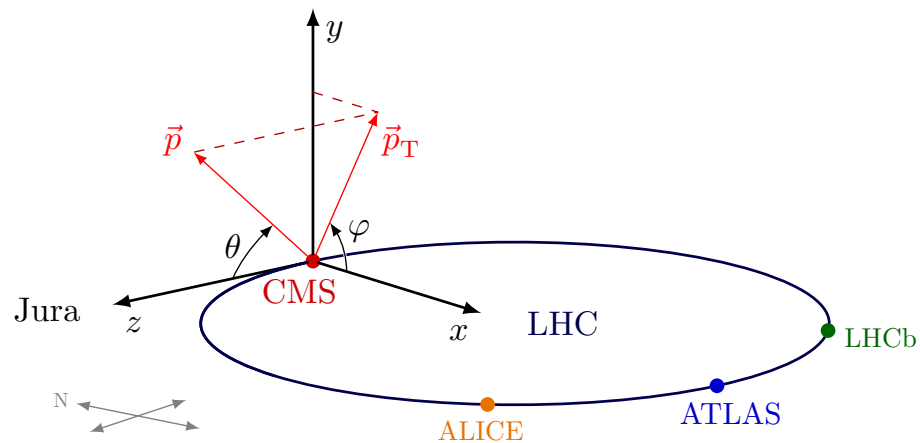


Figure 1. The CMS coordinates the system against the backdrop of the LHC, with the location of the four main experiments (CMS, ALICE, ATLAS, and LHCb). The z axis points to the Jura mountains, while the y -axis points toward the sky. In spherical coordinates, the components of a particle momentum \vec{p} are its magnitude $|\vec{p}|$, the polar angle θ (measured from the z -axis), and the azimuthal angle φ (measured from the x -axis). The transverse momentum \vec{p}_T is the projection of \vec{p} on the transverse (xy) plane. This figure was generated with TikZ code adapted from Ref. [58].

For a more intuitive understanding of the jet images in our dataset, we show several visualizations in Figures 2 and 3. Figure 2 shows the various subdetector images for a single jet: a representative quark jet in the upper row and a representative gluon jet in the bottom row. Then, Figure 3 shows the corresponding subdetector images averaged over the full

dataset. The ECAL images have 125×125 resolution in the plane of the azimuthal angle ϕ' and the pseudorapidity η' , while the HCAL resolution is only 25×25 in the (ϕ', η') plane.

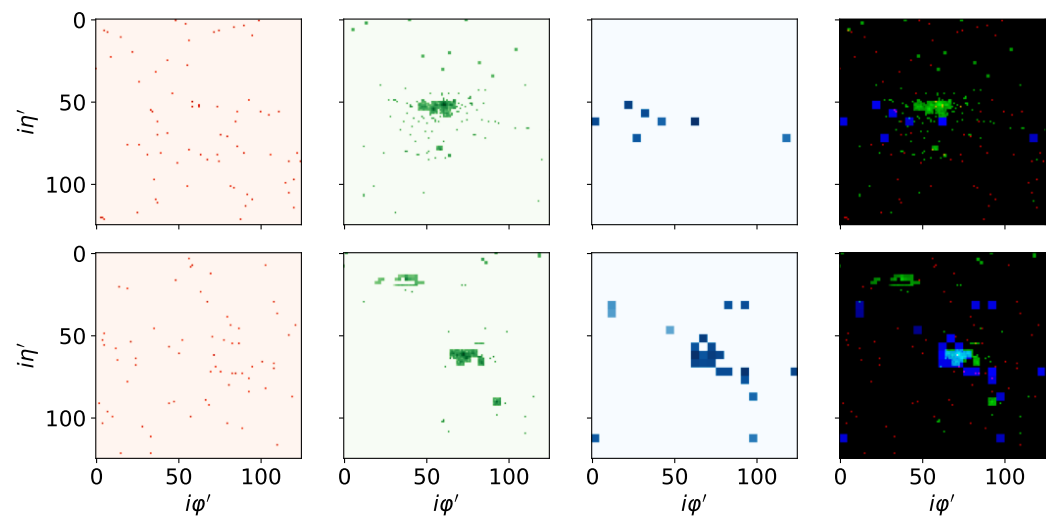


Figure 2. Representative images of jets for both quarks (**top**) and gluons (**bottom**). The columns show the distinct sub-detectors: Tracks, ECAL, HCAL, and a composite image combining all three. All images are in log scale. Note that the ECAL and HCAL were upscaled to match the Tracks resolution.

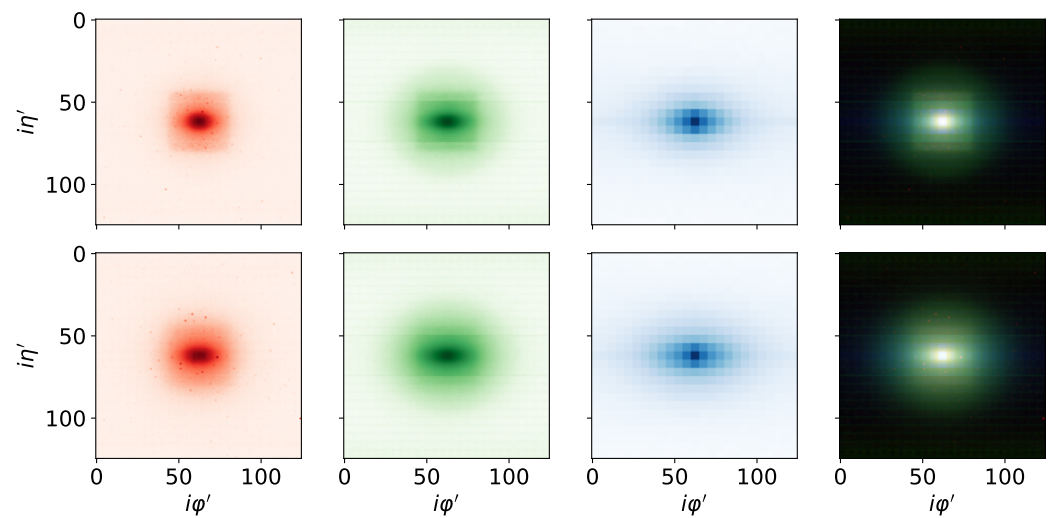


Figure 3. Average images of quarks (**top**) and gluons (**bottom**) across the entire dataset. The columns show the distinct sub-detectors: Tracks, ECAL, HCAL, and a composite image combining all three. All images are in log scale. Note the more dispersed nature of the gluon jets across channels.

3.2. Model

As in the original classical ViT [12], the image is split into patches that are linearly embedded together with position embeddings. Nonetheless, the change we introduce is that these patches are instead fed to the Quantum Transformer Encoder, which employs VQCs in the multi-head attention (MHA) and multi-layer perceptron (MLP) components. An overview of the model is shown in Figure 4.

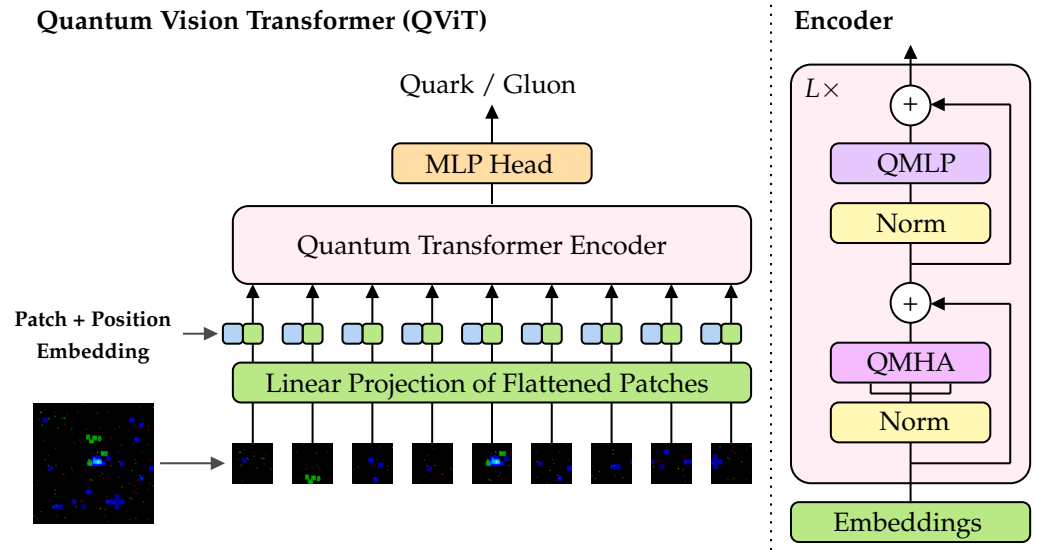


Figure 4. Model overview. QMHA stands for quantum multi-head attention and QMLP for quantum multi-layer perceptron. The drawing style of the illustration was inspired by Dosovitskiy et al. [12], the major difference being that here we use a quantum transformer encoder as depicted in the right side of the figure.

More concretely, the output of the classical multi-head attention layer is computed by using VQCs to compute all four linear projections in the MHA computations (Equation (13)) instead of classical feedforward layers. Similarly, in the MLP component of the encoder, we also employ VQCs to replace classical fully connected layers. Nonetheless, note that the activation functions in the MLP, which are GELU [30], are executed classically.

In particular, the VQC configuration we use is the one shown in Figure 5. First, each feature of the vector $x = (x_0, \dots, x_{n-1})$ is embedded into the qubits by encoding them into their rotation angles. Next, a layer of one-parameter single-qubit rotations acts on each wire. These parameters, $\theta = (\theta_0, \dots, \theta_{n-1})$, are learned together with the rest of the parameters of the model. Then, a ring of CNOT gates follows to entangle the qubit states. Thus, the obtained behavior is similar to a matrix multiplication. Finally, each qubit is measured, and the output is fed to the next corresponding component of the encoder.

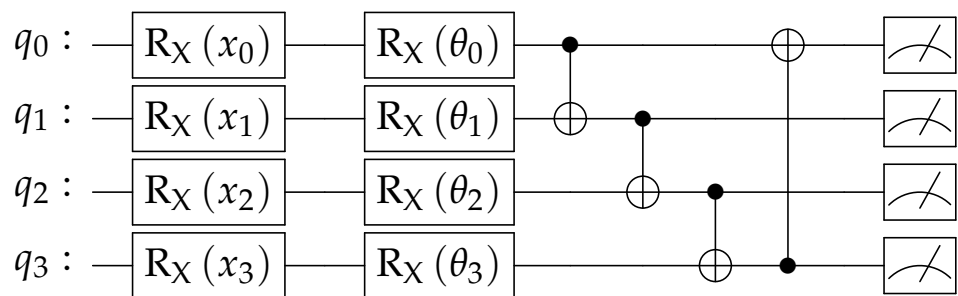


Figure 5. Variational quantum circuits used in the proposed QViT.

We train both the proposed QViT and a classical ViT with the same hyperparameters to have a meaningful baseline for comparison. We use a patch size of ten, a hidden size of eight, and four transformer blocks with four attention heads each and a hidden MLP size of four.

As suggested by recent work on benchmarking quantum utility [59], we choose the classical and the quantum architectures to have a similar number of trainable parameters. Note that since the input and output states of a VQC have the same dimension, the number of qubits has to coincide with the size of the corresponding layers in the neural network.

This results in the use of four circuits made up of four qubits for the QMHA layer of each transformer block, and, likewise, four circuits made up of four qubits for the QMLs. In total, the classical ViT has 5178 parameters, while the QViT has 4170 parameters. The smaller number in the QViT is due to the fact that the proposed VQC has only n free parameters, while a classical fully connected layer with bias has $n^2 + n$ parameters.

The dimensions used are small so that the circuits do not require many qubits. Consequently, the simulation time is not very long, and the model can be executed in already existing quantum hardware.

We use a batch size of 256 and train for 25 epochs with the AdamW optimizer [60] with gradient clipping at norm 1, and a learning rate scheduler that first performs a linear warmup for 5000 steps from 0 to 10^{-3} , followed by cosine decay [61]. We execute a random hyperparameter search to find good parameters in the classical baseline and apply them to the QViT.

We use the same training–validation–test split as in Andrews et al. [24]. In particular, of the whole dataset, 714,510 images are allocated for training, 79,390 for validation, and 139,306 for the final test set. To assess the classifier’s performance, we employ the Receiver Operating Characteristic (ROC) curve. In the context of high-energy physics, this curve can be interpreted in terms of signal efficiency (true positive rate) versus background rejection (true negative rate). The area under the ROC curve (AUC) is computed for each epoch of each model configuration. After all the epochs, we select the parameters from the epoch that achieves the highest validation AUC and reevaluate them on the separate hold-out test set to obtain the final test AUC.

We use JAX [62] and Flax [63] to implement the classical parts of the model and the classical baseline, as well as to train both models. We use TensorCircuit [64] to implement, train, and execute the VQCs by numerical simulation on a classical computer. By using TensorCircuit, we are able to train the quantum model for several epochs in a relatively short amount of time (around 39 minutes per epoch). This is an improvement over previous works, such as Di Sipio et al. [13], which required about 100 h to train a similar hybrid transformer model for just one epoch, even though we have many more samples.

4. Results

The evolution of the loss and AUC score during training, computed at the end of each epoch, is shown in Figures 6 and 7, respectively. We do not observe signs of overfitting in any case, as the training and validation curves are almost the same.

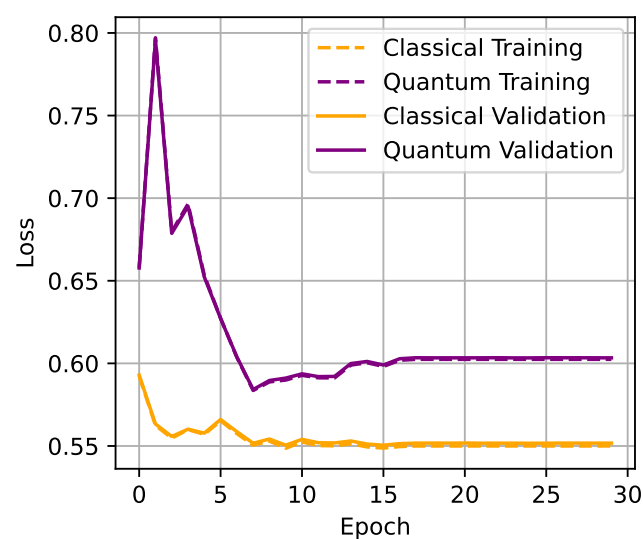


Figure 6. Binary cross-entropy loss evolution during training, computed at the end of each epoch on the training (dashed lines) and validation (solid lines) sets for both the baseline classical ViT (orange lines) and our hybrid QViT (purple lines).

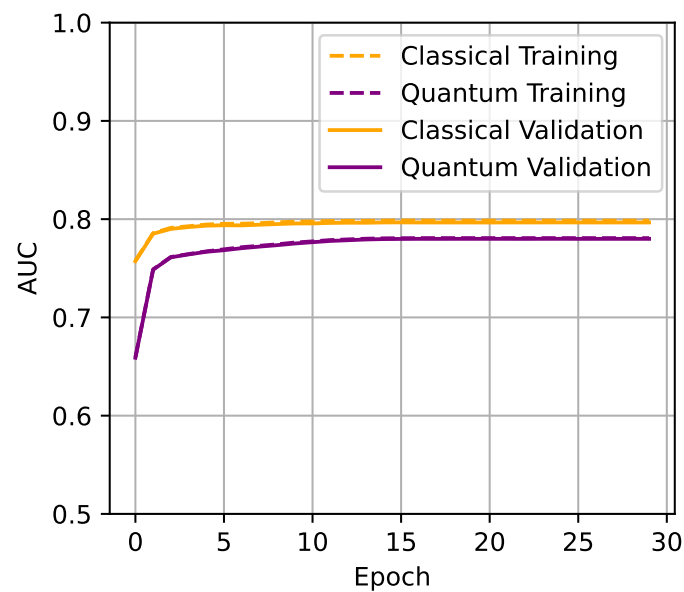


Figure 7. AUC score evolution during training computed at the end of each epoch on the training (dashed lines) and validation (solid lines) sets for both the baseline classical ViT (orange lines) and our hybrid QViT (purple lines).

The epoch that obtains the highest validation AUC is the 16th in the case of the classical ViT and 25th in the case of our hybrid QViT. Although the ViT converges faster, we observe that the QViT converges quite fast too, but keeps improving slightly for a few more epochs.

With the parameters from the best epoch of each model, we compute the ROC curve and AUC score on the separate hold-out test set. We show the achieved test ROC curve and its AUC scores in Figure 8.

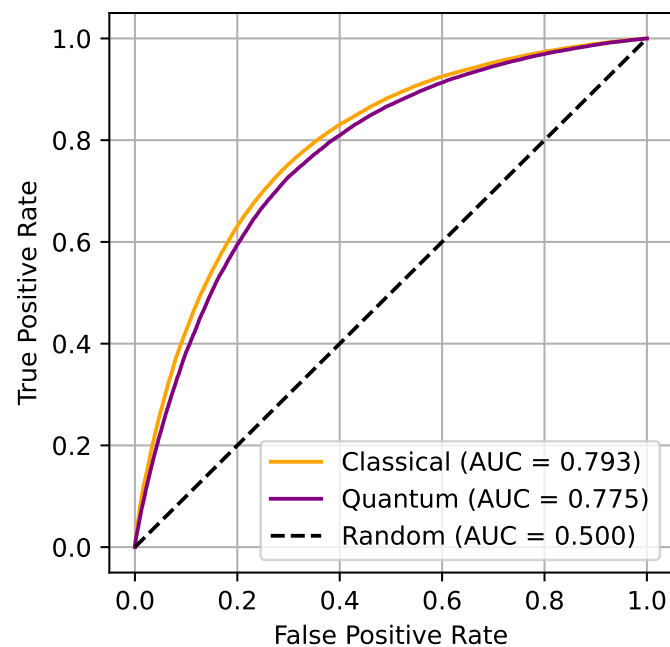


Figure 8. Receiver Operating Characteristic (ROC) curves for the baseline classical ViT (orange line) and our hybrid QViT (purple line). The black dashed line represents the performance of a random classifier.

We observe that the proposed QViT results in almost the same ROC curve and obtains almost the same AUC score as the classical baseline, though it still lags by approximately two percentage points. We hypothesize that one potential reason for the slightly inferior performance of the quantum model is that it is harder for the optimizer to find good parameters within the numerically simulated VQCs. Alternatively, the proposed VQCs might lack the expressiveness required to match or exceed the performance of the classical model. Still, we note that the difference between both obtained metrics is quite small.

5. Conclusions

In this work, we introduced a quantum-classical hybrid approach to vision transformers and applied it to the task of quark-gluon classification of sub-detector images from the CMS Open Data. The novel element is the integration of variational quantum circuits within both the attention mechanism and the multi-layer perceptrons. The trained model was benchmarked against a classical vision transformer with the same hyperparameters and a similar number of trainable parameters and was found to have comparable performance. The results achieved so far are encouraging and warrant future investigations.

Moving forward, our plans include evaluating more hyperparameter configurations, assessing the impact of the number of training samples, and experimenting with data augmentation techniques that have been shown to improve classical ViTs [33,65], such as RandAugment [66] and Mixup [67]. We also aim to explore different configurations for the VQCs, as well as evaluate the usage of data re-uploading [68] to check if we obtain a quantum advantage. Finally, we would also like to execute the VQCs on real quantum hardware to measure the performance of the proposed QViT in it, as well as to assess the robustness to quantum noise.

Ideally, the progress in improving the performance of the ML and QML algorithms should be accompanied by progress in understanding the fundamental physics behind the hadronization of quarks and gluons. As a first step in this direction, one could use symbolic learning to obtain interpretable analytical formulas that capture the decision-making of our trained classifiers [69].

Author Contributions: Conceptualization, M.C.C.; methodology, M.C.C., G.R.D., Z.D., R.T.F., S.G., D.J., K.K., T.M., K.T.M., K.M. and E.B.U.; software, M.C.C.; validation, M.C.C., G.R.D., Z.D., R.T.F., T.M. and E.B.U.; formal analysis, M.C.C.; investigation, M.C.C., G.R.D., Z.D., R.T.F., T.M. and E.B.U.; resources, M.C.C. and S.G.; data curation, G.R.D., S.G. and T.M.; writing—original draft preparation, M.C.C.; writing—review and editing, S.G., D.J., K.K., K.T.M. and K.M.; visualization, M.C.C.; supervision, S.G., D.J., K.K., K.T.M. and K.M.; project administration, S.G., D.J., K.K., K.T.M. and K.M.; funding acquisition, S.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research used resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231 using NERSC award NERSC DDR-ERCAP0025759. SG is supported in part by the U.S. Department of Energy (DOE) under Award No. DE-SC0012447. KM is supported in part by the U.S. DOE award number DE-SC0022148. KK is supported in part by US DOE DE-SC0024407. CD is supported in part by the College of Liberal Arts and Sciences Research Fund at the University of Kansas. CD, RF, EU, MCC, and TM were participants in the 2023 Google Summer of Code.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The code and data we used to train and evaluate our models are available at https://github.com/ML4SCI/QMLHEP/tree/main/Quantum_Transformers_Mar%2024/A7a1_Comajoan_Cara (accessed on 14 March 2024).

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

ALICE	A Large Ion Collider Experiment
ATLAS	A Toroidal LHC ApparatuS
AUC	Area Under the Curve
CERN	Conseil Européen pour la Recherche Nucléaire
CMS	Compact Muon Solenoid (experiment)
CNOT	Controlled NOT
ECAL	electromagnetic calorimeter
GELU	Gaussian Error Linear Unit
HCAL	hadronic calorimeter
HL-LHC	High Luminosity Large Hadron Collider
LHC	Large Hadron Collider
LHCb	Large Hadron Collider beauty (experiment)
MDPI	Multidisciplinary Digital Publishing Institute
MHA	multi-head attention
MLP	multi-layer perceptron
QCD	Quantum Chromodynamics
QMHA	quantum multi-head attention
QML	quantum machine learning
QMLP	quantum multi-layer perceptron
QViT	quantum vision transformer
ReLU	Rectified Linear Unit
ROC	receiver operating characteristic
SGD	stochastic gradient descent
ViT	vision transformer
VQC	variational quantum circuit

References

1. CERN. The HL-LHC Project. 2022. Available online: <https://hilumilhc.web.cern.ch/content/hl-lhc-project> (accessed on 24 September 2023).
2. HSF Physics Event Generator WG; Valassi, A.; Yazgan, E.; McFayden, J.; Amoroso, S.; Bendavid, J.; Buckley, A.; Cacciari, M.; Childers, T.; Ciulli, V.; et al. Challenges in Monte Carlo Event Generator Software for High-Luminosity LHC. *Comput. Softw. Big Sci.* **2021**, *5*, 12. [\[CrossRef\]](#)
3. Arunachalam, S.; de Wolf, R. A Survey of Quantum Learning Theory. *arXiv* **2017**, arXiv:1701.06806. [\[CrossRef\]](#)
4. Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum machine learning. *Nature* **2017**, *549*, 195–202. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Schuld, M.; Killoran, N. Quantum Machine Learning in Feature Hilbert Spaces. *Phys. Rev. Lett.* **2019**, *122*, 040504. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Mangini, S.; Tacchino, F.; Gerace, D.; Bajoni, D.; Macchiavello, C. Quantum computing models for artificial neural networks. *Europhys. Lett.* **2021**, *134*, 10002. [\[CrossRef\]](#)
7. Liu, Y.; Arunachalam, S.; Temme, K. A rigorous and robust quantum speed-up in supervised machine learning. *Nat. Phys.* **2021**, *17*, 1013–1017. [\[CrossRef\]](#)
8. Huang, H.Y.; Broughton, M.; Cotler, J.; Chen, S.; Li, J.; Mohseni, M.; Neven, H.; Babbush, R.; Kueng, R.; Preskill, J.; et al. Quantum advantage in learning from experiments. *Science* **2022**, *376*, 1182–1186. [\[CrossRef\]](#)
9. Caro, M.C.; Huang, H.Y.; Cerezo, M.; Sharma, K.; Sornborger, A.; Cincio, L.; Coles, P.J. Generalization in quantum machine learning from few training data. *Nat. Commun.* **2022**, *13*, 4919. [\[CrossRef\]](#) [\[PubMed\]](#)
10. Dong, Z.; Comajoan Cara, M.; Dahale, G.R.; Forestano, R.T.; Gleyzer, S.; Justice, D.; Kong, K.; Magorsch, T.; Matchev, K.T.; Matcheva, K.; et al. $Z_2 \times Z_2$ Equivariant Quantum Neural Networks: Benchmarking against Classical Neural Networks. *Axioms* **2024**, *13*, 188. [\[CrossRef\]](#)
11. Forestano, R.T.; Comajoan Cara, M.; Dahale, G.R.; Dong, Z.; Gleyzer, S.; Justice, D.; Kong, K.; Magorsch, T.; Matchev, K.T.; Matcheva, K.; et al. A Comparison between Invariant and Equivariant Classical and Quantum Graph Neural Networks. *Axioms* **2024**, *13*, 160. [\[CrossRef\]](#)
12. Dosovitskiy, A.; Beyler, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations, Online, 3–7 May 2021.

13. Di Sipio, R.; Huang, J.H.; Chen, S.Y.C.; Mangini, S.; Worring, M. The Dawn of Quantum Natural Language Processing. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022; pp. 8612–8616. [\[CrossRef\]](#)
14. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.u.; Polosukhin, I. Attention is All you Need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2017; Volume 30.
15. Li, G.; Zhao, X.; Wang, X. Quantum Self-Attention Neural Networks for Text Classification. *arXiv* **2022**, arXiv:2205.05625. [\[CrossRef\]](#)
16. Cherrat, E.A.; Kerenidis, I.; Mathur, N.; Landman, J.; Strahm, M.C.; Li, Y.Y. Quantum Vision Transformers. *Quantum* **2024**, *8*, 1265. [\[CrossRef\]](#)
17. Unlu, E.B.; Comajoan Cara, M.; Dahale, G.R.; Dong, Z.; Forestano, R.T.; Gleyzer, S.; Justice, D.; Kong, K.; Magorsch, T.; Matchev, K.T.; et al. Hybrid Quantum Vision Transformers for Event Classification in High Energy Physics. *Axioms* **2024**, *13*, 187. [\[CrossRef\]](#)
18. Kollias, G.; Kalantzis, V.; Salonidis, T.; Ubaru, S. Quantum Graph Transformers. In Proceedings of the ICASSP 2023—2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; pp. 1–5. [\[CrossRef\]](#)
19. CERN. CMS Open Data. 2023. Available online: <http://opendata.cern.ch/docs/about-cms> (accessed on 24 September 2023).
20. The ATLAS Collaboration. *Quark versus Gluon Jet Tagging Using Jet Images with the ATLAS Detector*; Technical Report; CERN: Geneva, Switzerland, 2017. Available online: <https://cds.cern.ch/record/2275641> (accessed on 12 May 2024).
21. The CMS Collaboration. New Developments for Jet Substructure Reconstruction in CMS. 2017. Available online: <https://cds.cern.ch/record/2275226> (accessed on 8 May 2024).
22. Cheng, T. Recursive Neural Networks in Quark/Gluon Tagging. *Comput. Softw. Big Sci.* **2018**, *2*, 3. [\[CrossRef\]](#)
23. Louppe, G.; Cho, K.; Becot, C.; Cranmer, K. QCD-aware recursive neural networks for jet physics. *J. High Energy Phys.* **2019**, *2019*, 57. [\[CrossRef\]](#)
24. Andrews, M.; Alison, J.; An, S.; Burkle, B.; Gleyzer, S.; Narain, M.; Paulini, M.; Poczos, B.; Usai, E. End-to-end jet classification of quarks and gluons with the CMS Open Data. *Nucl. Instrum. Methods Phys. Res. Sect. A Accel. Spectrometers Detect. Assoc. Equip.* **2020**, *977*, 164304. [\[CrossRef\]](#)
25. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [\[CrossRef\]](#)
26. Bishop, C.M.; Bishop, H. *Deep Learning*, 1st ed.; Springer: Cham, Switzerland, 2023.
27. Schmidhuber, J. Annotated History of Modern AI and Deep Learning. *arXiv* **2022**, arXiv:2212.11279.
28. Fukushima, K. Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements. *IEEE Trans. Syst. Sci. Cybern.* **1969**, *5*, 322–333. [\[CrossRef\]](#)
29. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. JMLR Workshop and Conference Proceedings, Fort Lauderdale, FL, USA, 11–13 April 2011; pp. 315–323.
30. Hendrycks, D.; Gimpel, K. Gaussian Error Linear Units (GELUs). *arXiv* **2016**, arXiv:1606.08415.
31. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer normalization. *arXiv* **2016**, arXiv:1607.06450.
32. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:1512.03385.
33. Beyer, L.; Zhai, X.; Kolesnikov, A. Better plain ViT baselines for ImageNet-1k. *arXiv* **2022**, arXiv:2205.01580. [\[CrossRef\]](#)
34. Rebertost, P.; Mohseni, M.; Lloyd, S. Quantum Support Vector Machine for Big Data Classification. *Phys. Rev. Lett.* **2014**, *113*, 130503. [\[CrossRef\]](#)
35. Wiebe, N.; Kapoor, A.; Svore, K.M. Quantum Algorithms for Nearest-Neighbor Methods for Supervised and Unsupervised Learning. *Quantum Inf. Comput.* **2015**, *15*, 316–356.
36. Johri, S.; Debnath, S.; Mocherla, A.; Singk, A.; Prakash, A.; Kim, J.; Kerenidis, I. Nearest centroid classification on a trapped ion quantum computer. *npj Quantum Inf.* **2021**, *7*, 122. [\[CrossRef\]](#)
37. Franceschini, R.; Kim, D.; Kong, K.; Matchev, K.T.; Park, M.; Shyamsundar, P. Kinematic variables and feature engineering for particle phenomenology. *Rev. Mod. Phys.* **2023**, *95*, 045004. [\[CrossRef\]](#)
38. Ellis, R.K.; Stirling, W.J.; Webber, B.R. *QCD and Collider Physics*; Cambridge University Press: Cambridge, UK, 2011; Volume 8. [\[CrossRef\]](#)
39. Salam, G.P. Towards Jetography. *Eur. Phys. J. C* **2010**, *67*, 637–686. [\[CrossRef\]](#)
40. Larkoski, A.J.; Moul, I.; Nachman, B. Jet Substructure at the Large Hadron Collider: A Review of Recent Advances in Theory and Machine Learning. *Phys. Rept.* **2020**, *841*, 1–63. [\[CrossRef\]](#)
41. Kogler, R.; Nachman, B.; Schmidt, A.; Asquith, L.; Winkels, E.; Campanelli, M.; Delitzsch, C.; Harris, P.; Hinzmann, A.; Kar, D.; et al. Jet Substructure at the Large Hadron Collider: Experimental Review. *Rev. Mod. Phys.* **2019**, *91*, 045003. [\[CrossRef\]](#)
42. Marzani, S.; Soyez, G.; Spannowsky, M. *Looking Inside Jets: An Introduction to Jet Substructure and Boosted-Object Phenomenology*; Springer: Berlin/Heidelberg, Germany, 2019; Volume 958. [\[CrossRef\]](#)
43. Feickert, M.; Nachman, B. A Living Review of Machine Learning for Particle Physics. *arXiv* **2021**, arXiv:2102.02770.
44. Guest, D.; Cranmer, K.; Whiteson, D. Deep Learning and its Application to LHC Physics. *Ann. Rev. Nucl. Part. Sci.* **2018**, *68*, 161–181. [\[CrossRef\]](#)

45. Albertsson, K.; Altoe, P.; Anderson, D.; Andrews, M.; Araque Espinosa, J.P.; Aurisano, A.; Basara, L.; Bevan, A.; Bhimji, W.; Bonacorsi, D.; et al. Machine Learning in High Energy Physics Community White Paper. *J. Phys. Conf. Ser.* **2018**, *1085*, 022008. [[CrossRef](#)]
46. Radovic, A.; Williams, M.; Rousseau, D.; Kagan, M.; Bonacorsi, D.; Himmel, A.; Aurisano, A.; Terao, K.; Wongjirad, T. Machine learning at the energy and intensity frontiers of particle physics. *Nature* **2018**, *560*, 41–48. [[CrossRef](#)] [[PubMed](#)]
47. Carleo, G.; Cirac, I.; Cranmer, K.; Daudet, L.; Schuld, M.; Tishby, N.; Vogt-Maranto, L.; Zdeborová, L. Machine learning and the physical sciences. *Rev. Mod. Phys.* **2019**, *91*, 045002. [[CrossRef](#)]
48. Bourilkov, D. Machine and Deep Learning Applications in Particle Physics. *Int. J. Mod. Phys. A* **2020**, *34*, 1930019. [[CrossRef](#)]
49. Schwartz, M.D. Modern Machine Learning and Particle Physics. *arXiv* **2021**, arXiv:2103.12226. [[CrossRef](#)]
50. Karagiorgi, G.; Kasieczka, G.; Kravitz, S.; Nachman, B.; Shih, D. Machine Learning in the Search for New Fundamental Physics. *arXiv* **2021**, arXiv:2112.03769.
51. Boehnlein, A.; Diefenthaler, M.; Sato, N.; Schram, M.; Ziegler, V.; Fanelli, C.; Hjorth-Jensen, M.; Horn, T.; Kuchera, M.P.; Lee, D.; et al. Colloquium: Machine learning in nuclear physics. *Rev. Mod. Phys.* **2022**, *94*, 031003. [[CrossRef](#)]
52. Shanahan, P.; Terao, K.; Whiteson, D. Snowmass 2021 Computational Frontier CompF03 Topical Group Report: Machine Learning. *arXiv* **2022**, arXiv:2209.07559.
53. Collaboration, C.M.; Chatrchyan, S.; Hmayakyan, G.; Khachatryan, V.; Sirunyan, A.M.; Adam, W.; Bauer, T.; Bergauer, T.; Bergauer, H.; Dragicevic, M.; et al. The CMS Experiment at the CERN LHC. *JINST* **2008**, *3*, S08004. [[CrossRef](#)]
54. CMS Collaboration. Description and performance of track and primary-vertex reconstruction with the CMS tracker. *JINST* **2014**, *9*, P10009. [[CrossRef](#)]
55. CMS Collaboration. Energy Calibration and Resolution of the CMS Electromagnetic Calorimeter in pp Collisions at $\sqrt{s} = 7$ TeV. *JINST* **2013**, *8*, P09009. [[CrossRef](#)]
56. Abdullin, S.; Abramov, V.; Acharya, B.; Adams, M.; Akchurin, N.; Akgun, U.; Anderson, E.W.; Antchev, G.; Ayan, S.; Aydin, S.; et al. Design, performance, and calibration of CMS hadron-barrel calorimeter wedges. *Eur. Phys. J. C* **2008**, *55*, 159–171. [[CrossRef](#)]
57. Abdullin, S.; Abramov, V.; Acharya, B.; Adams, M.; Akchurin, N.; Akgun, U.; Anderson, E.W.; Antchev, G.; Arcidy, M. Design, performance, and calibration of the CMS Hadron-outer calorimeter. *Eur. Phys. J. C* **2008**, *57*, 653–663. [[CrossRef](#)]
58. CMS Coordinate System. Available online: https://tikz.net/axis3d_cms/ (accessed on 6 March 2024).
59. Herrmann, N.; Arya, D.; Doherty, M.W.; Mingare, A.; Pillay, J.C.; Preis, F.; Prestel, S. Quantum utility—Definition and assessment of a practical quantum advantage. In Proceedings of the 2023 IEEE International Conference on Quantum Software, Chicago, IL, USA, 2–8 July 2023; pp. 162–174. [[CrossRef](#)]
60. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. *arXiv* **2019**, arXiv:1711.05101. [[CrossRef](#)]
61. Loshchilov, I.; Hutter, F. SGDR: Stochastic Gradient Descent with Warm Restarts. *arXiv* **2017**, arXiv:1608.03983.
62. Bradbury, J.; Frostig, R.; Hawkins, P.; Johnson, M.J.; Leary, C.; Maclaurin, D.; Necoala, G.; Paszke, A.; VanderPlas, J.; Wanderman-Milne, S.; et al. JAX: Composable Transformations of Python+NumPy Programs. 2023. Available online: <http://github.com/google/jax> (accessed on 24 September 2023).
63. Heek, J.; Levskaia, A.; Oliver, A.; Ritter, M.; Rondepierre, B.; Steiner, A.; van Zee, M. Flax: A Neural Network Library and Ecosystem for JAX. 2023. Available online: <http://github.com/google/flax> (accessed on 24 September 2023).
64. Zhang, S.X.; Allcock, J.; Wan, Z.Q.; Liu, S.; Sun, J.; Yu, H.; Yang, X.H.; Qiu, J.; Ye, Z.; Chen, Y.Q.; et al. TensorCircuit: A Quantum Software Framework for the NISQ Era. *Quantum* **2023**, *7*, 912. [[CrossRef](#)]
65. Steiner, A.; Kolesnikov, A.; Zhai, X.; Wightman, R.; Uszkoreit, J.; Beyer, L. How to train your ViT? Data, Augmentation, and Regularization in Vision Transformers. *arXiv* **2022**, arXiv:2106.10270. [[CrossRef](#)]
66. Cubuk, E.D.; Zoph, B.; Shlens, J.; Le, Q. RandAugment: Practical Automated Data Augmentation with a Reduced Search Space. In Proceedings of the Advances in Neural Information Processing Systems, Online, 6–12 December 2020; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 18613–18624.
67. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. mixup: Beyond Empirical Risk Minimization. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
68. Pérez-Salinas, A.; Cervera-Lierta, A.; Gil-Fuster, E.; Latorre, J.I. Data re-uploading for a universal quantum classifier. *Quantum* **2020**, *4*, 226. [[CrossRef](#)]
69. Dong, Z.; Kong, K.; Matchev, K.T.; Matcheva, K. Is the machine smarter than the theorist: Deriving formulas for particle kinematics with symbolic regression. *Phys. Rev. D* **2023**, *107*, 055018. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.