JetVision-Mamba: Selective State Space Models for Jet Classification in High Energy Physics

Dimitris Ntounis
Stanford University
450 Jane Stanford Way, Stanford, California 94305, USA &
SLAC National Accelerator Laboratory
2575 Sand Hill Road, Menlo Park, California 94025, USA
dntounis@stanford.edu

Abstract

Jet flavor tagging is a critical but challenging classification task for High Energy Physics (HEP) experiments, as it allows physicists to deduce the type –or flavor – of particle that originated a hadronic jet, enabling measurements of the Higgs boson and other fundamental particles, while suppressing the huge backgrounds for Quantum Chromodynamics (QCD) processes. While deep learning algorithms, most recently based on Graph Neural Network (GNN) and Transformer architectures, have been successfully used for jet tagging, in this work, we propose a novel algorithm based on the Mamba State Space Model (SSM) architecture but extended to two-dimensions, where jets are represented as jet images. We train our model on a large dataset of ten flavors of jets and evaluate its performance against state of the art models. Although further refinements are necessary in order for it to acquire state-of-the-art performance, this work represents the first application of Mamba SSMs in the domain of jet flavor tagging.

1. Introduction

In HEP colliders, such as the Large Hadron Collider (LHC) at CERN in Geneva, beams of protons accelerated to speeds very close to the speed of light are brought to intersect at the centers of massive detectors, such as the ATLAS [1] and CMS [5] experiments. The collisions of these particles produce rare, heavy particles, such as the W, Z and Higgs bosons, the study of which is essential for understanding the fabric of Nature and the origin of the Cosmos. These particles are unstable and immediately decay to lighter ones, such as gluons (g), quarks (q), photons (γ) and leptons (electrons e and muons μ). Quarks and gluons subsequently *hadronize*, meaning that that they produce a collimated spray of secondary particles, mostly including a type of particles called hadrons. This spray of particles, which we call a *jet*, is the experimental manifestation of quarks and gluons and is directly the object that the experiments detect.

Physicists are interested in measuring the properties of the particles included in a jet and reverse-engineering the type of initial particle that originated it. This helps them identify, for example, if the jet originated from a Higgs boson decaying to two bottom-type quarks, which we denote as $H \rightarrow b\bar{b}$, or from a top qark decaying to a bottom and two lighter quarks ($t \rightarrow bqq^{'}$) etc. Most importantly, this also allows us to differentiate between jets originating from interesting physical processes, such as the ones just mentioned, or from QCD, which is the main background in our detector, and is typically more abundant by around *6 orders of magnitude*. This is precisely why jet tagging is so crucial: it allows us to classify jets as coming, for instance, from Higgs boson decays versus QCD interactions, thus suppressing the contribution of the latter and increasing the experimental sensitivity of detecting rare processes such as the former. Indeed, using such jet tagging models, often called *jet taggers*, was essential for the discovery of the Higgs boson by the ATLAS and CMS experiments in 2012 [1, 6].

Jet flavor tagging is essentially a *multi-class classification task*, where the input features are some representation of the properties of the jet constituent particles and the output is the predicted label of the jet. Each jet is treated as a collection of up to N_{jc} jet constituents, with each one having up to N_{pf} particle features that we can exploit. Therefore, for one jet, our problem is formulated as the following classification task:

$$[N_{jc}, N_{pf}] \text{ features} \xrightarrow{\text{classify}} C \text{ classes}$$

In this project, we choose to represent jets as 2D images, where the horizontal (vertical) axis represents the az-

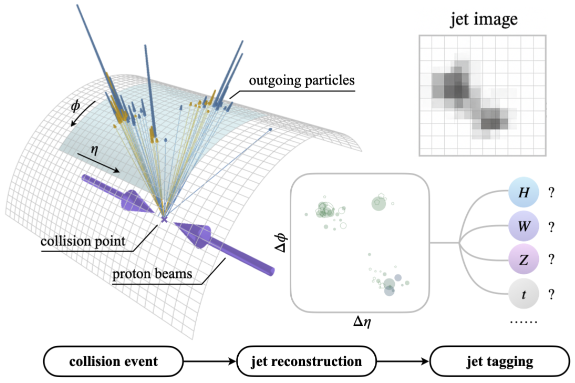


Figure 1: Illustration of jet tagging at the CERN LHC. The goal is deduce the parent particle, e.g. Higgs,W,Z boson or top quark that created the collimated spray of particles, or jet, by reconstructing the properties of the jet and its constituents. In our approach, we are specifically depicting the jet as a "jet image". Graphic inspired from [24].

imuthal (pseudorapidity[1]) distance $\Delta\phi(\Delta\eta)$ of the jet particles with respect to the center of the jet. The jet image has multiple channels, one for each distinct property of the jet constituents, related to the particles' kinematics, particle identification and trajectory displacement. We develop a novel algorithm that employs 2D Mamba layers to capture the spatial correlations in the jet images. A schematic representation of our approach is given in Fig. 1.

This paper is structured as follows: We begin by reviewing related work on jet flavor tagging in Section 2. We then present our proposed method in Section 3, analyze the dataset we train it on in Section 4 and present our results and comparison with state-of-the-art (SOTA) jet taggers in Section 5. Finally, our conclusions and ideas for future work are given in Section 6.

2. Related Work

Jet flavor tagging has evolved significantly over the past decades, transitioning from traditional physics-based approaches to sophisticated deep learning architectures. We categorize the existing literature into several distinct approaches, each with unique strengths and limitations.

2.1. Traditional Physics-Based Approaches

Early jet tagging algorithms relied heavily on physics-motivated observables and hand-crafted features. The Combined Secondary Vertex (CSV) algorithm [7] and its variants represented the SOTA for many years, utilizing secondary vertex reconstruction and track impact parameters to identify jets coming from heavy-flavor quarks, such as the bottom (b) quark. These methods, while interpretable and computationally efficient, were limited by their reliance on expert domain knowledge and struggled to capture complex correlations between jet constituents. The Boosted Decision Tree (BDT) approaches [3] also improved upon simple cut-based methods by combining multiple observables, but remained fundamentally constrained by the high-level nature of the input features.

2.2. Convolutional Neural Networks and Jet Images

The introduction of jet images marked a paradigm shift in jet tagging [8, 9]. By representing jets as 2D images in the η-ϕ plane, these approaches enabled the application of computer vision techniques to HEP. Subsequent work explored various CNN architectures [18, 19], demonstrating significant improvements over traditional methods. However, these image-based approaches faced challenges in handling the irregular and sparse nature of jet data, often requiring careful preprocessing and leading to information loss during the pixelization process. Despite these limitations, CNN-based methods established the viability of deep learning for jet classification and inspired further research into more sophisticated architectures.

2.3. Graph Neural Network Approaches

Recognizing that jets are naturally represented as irregular point clouds rather than regular grids, Graph Neural Networks (GNNs) emerged as a powerful paradigm for jet tagging. PARTICLENET [23], based on Dynamic Graph CNN (DGCNN) [27], treats jet constituents as nodes in a graph and learns representations through message passing. This approach handles the variable number of particles per jet and preserves the permutation invariance inherent in particle physics. Subsequent GNN-based methods [17, 26] explored different graph construction strategies and message passing mechanisms. Overall, GNNs were able to achieve excellent performance although they often required careful hyperparameter tuning and could be sensitive to the choice of edge connectivity, somewhat limiting their robustness across different physics scenarios.

2.4. Transformer-Based Architectures

The success of Transformers in natural language processing inspired their adaptation to jet tagging. PARTICLETRANSFORMER [24] applies self-attention mechanisms to learn relationships between jet constituents, achieving SOTA performance across multiple benchmarks. The attention mechanism naturally handles variable-length sequences and captures long-range dependencies between particles. However, the quadratic complexity of self-attention with respect to the number of particles poses computational challenges, particularly for jets with many

[1] The pseudorapidity is defined as $\eta = -\ln\tan(\theta/2)$, where θ the polar angle in cylindrical coordinates of the particle with respect to the axis of the colliding particles. For more information see [25].

constituents. Recent work has explored more efficient attention variants [20, 11], but the fundamental scalability limitation remains a significant concern for real-time applications in experimental environments.

2.5. Other Deep Learning Approaches

Various other deep learning architectures have been explored for jet tagging. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks [16] were among the early applications, treating jets as sequences of particles. More recent work has investigated variational autoencoders for anomaly detection [13], generative adversarial networks for jet simulation [10], and ensemble methods combining multiple architectures [4]. While these approaches have shown promise in specific contexts, they have generally been superseded by the more systematic GNN and Transformer approaches.

2.6. State Space Models and Sequential Processing

State Space Models (SSMs) have recently gained attention in machine learning for their ability to efficiently model long sequences. The Mamba architecture [14] introduced selective state spaces that can focus on relevant information while maintaining linear computational complexity. Vision Mamba [31] successfully adapted these concepts to computer vision tasks, demonstrating competitive performance with Transformers while offering superior efficiency. However, the application of SSMs to particle physics has remained largely unexplored, with only limited work on time-series analysis of detector signals [2].

2.7. Positioning Our Approach

Our work addresses several key limitations in existing approaches. Unlike traditional CNNs that struggle with irregular jet data, our 2D Mamba architecture processes multi-channel jet images while maintaining spatial relationships through selective scanning patterns. Compared to GNNs, which require explicit graph construction, our approach automatically learns spatial dependencies through the SSM mechanism. Most importantly, while Transformers suffer from quadratic complexity, our Mamba-based approach achieves linear scaling with the number of pixels, potentially enabling more efficient processing of high-resolution jet representations. This work represents the first application of selective state space models to jet flavor tagging, opening new avenues for efficient and scalable jet classification algorithms.

3. Methods

In this section, we detail our proposed JetVision-Mamba architecture and discuss the baseline methods against which we compare our results.

3.1. Baseline Methods

PARTICLENET [23] treats jets as irregular point clouds where each constituent particle is represented as a node in a dynamic graph. The architecture constructs k-nearest neighbor graphs in the $\eta - \phi$ plane and applies EdgeConv operations to capture local particle interactions. For each particle i, the EdgeConv operation aggregates information from its neighbors $\mathcal{N}(i)$:

$$\mathbf{x}'_i = \max_{j \in \mathcal{N}(i)} \text{ReLU}\left(\mathbf{\Theta} \cdot (\mathbf{x}_j - \mathbf{x}_i) + \mathbf{\Phi} \cdot \mathbf{x}_i\right)$$

where $\mathbf{x}_i$ is the feature vector of particle i, and $\mathbf{\Theta}, \mathbf{\Phi}$ are learnable weight matrices. This graph-based approach handles a variable number of particles per jet while preserving permutation invariance, but requires careful tuning of the k-nearest neighbor connectivity and can be sensitive to the choice of distance metrics in the $\eta - \phi$ space.

PARTICLETRANSFORMER [24] applies the transformer architecture to model long-range dependencies between all jet constituents simultaneously. Each particle is treated as a token, and self-attention mechanisms compute pairwise interactions across all particles:

$$\text{Attention}_{ij} = \text{softmax}\left(\frac{Q_i K_j^T}{\sqrt{d_k}} + \boldsymbol{U}_{ij}\right)$$

where $\boldsymbol{U}_{ij}$ is a learned pairwise interaction term that encodes physics-motivated inductive biases between particles i and j. While PARTICLETRANSFORMER achieves SOTA performance, its quadratic scaling with the number of particles ($\mathcal{O}(N^2)$) presents computational challenges for jets with many constituents, limiting its applicability in real-time environments.

3.2. Proposed Method: JetVision-Mamba

Our approach combines the spatial representation advantages of jet images with the computational efficiency of selective SSMs. The main idea is to convert inherently unordered set of jet constituents into structured 2D images that preserve spatial relationships, then processing these images with 2D Mamba blocks that achieve linear computational complexity.

3.2.1 Physics-Motivated Jet Image Preprocessing

We transform each jet from an unordered collection of particles into a multi-channel 2D image that preserves both spatial and feature information. Our preprocessing pipeline implements several physics-motivated transformations:

1. **Jet Centering:** We center each jet using ${p_T}^2$-weighted

[2] p_T represents the momentum in the plane transverse to the axis of the colliding beams and is often used as a proxy for the energy of a jet.

averages to ensure rotational and translational invariance:

$$\eta_{\text{center}} = \frac{\sum_i p_{T,i}\eta_i}{\sum_i p_{T,i}} \tag{1}$$

$$\phi_{\text{center}} = \arctan 2\left(\sum_i p_{T,i}\sin\phi_i, \sum_i p_{T,i}\cos\phi_i\right) \tag{2}$$

2. **Spatial Binning:** We create a 33×33 pixel grid spanning $\Delta R = 0.8$ in both η and ϕ directions, matching the standard jet reconstruction cone size.

3. **Multi-Channel Feature Engineering:** We use 15 particle features, which we will introduce in Section 4, with each particle feature becoming a separate image channel, with feature-specific preprocessing applied based on the expected dynamic range of each quantity[3] This multi-channel representation preserves both the spatial structure of jets and the rich feature information of constituent particles. An example of a jet image we created from a JetClass image is given in Fig. 2.

.

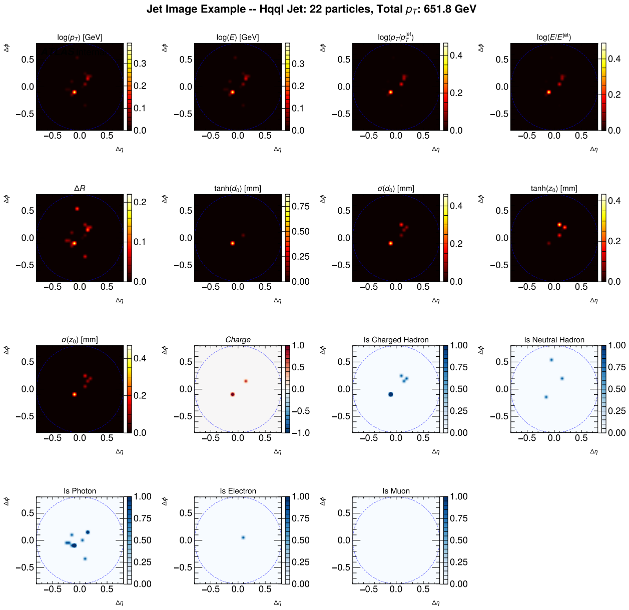


Figure 2: Example of a preprocessed jet image from the JetClass jet. The horizontal and vertical axes correspond to the $\Delta\eta$ and $\Delta\phi$ of each jet constituent with respect to the jet axis, and the 15 different plots correspond to the 15 image channels/input particle features.

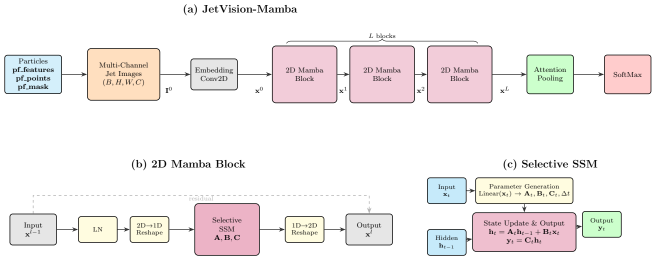


Figure 3: The architecture of (a) JetVision-Mamba, (b) the 2D Mamba block and (c) the selective SSM.

3.2.2 2D Selective State Space Models

Traditional state space models evolve hidden states through linear recurrence relations:

$$\mathbf{h}_t = \mathbf{A}\mathbf{h}_{t-1} + \mathbf{B}\mathbf{x}_t, \quad \mathbf{y}_t = \mathbf{C}\mathbf{h}_t\ .$$

The key innovation in Mamba [14] is the selective mechanism where parameters become input-dependent:

$$\mathbf{h}_t = \mathbf{A}_t\mathbf{h}_{t-1} + \mathbf{B}_t\mathbf{x}_t, \quad \mathbf{y}_t = \mathbf{C}_t\mathbf{h}_t$$

where $\mathbf{A}_t, \mathbf{B}_t, \mathbf{C}_t$ are computed from the input $\mathbf{x}_t$ through learned linear projections. This selectivity allows the model to focus on relevant information while compressing irrelevant details, which in our case is crucial for identifying discriminative patterns in jets.

For 2D applications, we extend this concept using the 2DMamba architecture [30] which converts 2D spatial information into 1D sequences through structured scanning patterns while preserving spatial continuity. Unlike previous approaches that simply flatten 2D structures, 2DMamba employs bidirectional scanning that processes the image in both forward and backward directions to maintain local neighborhood relationships. Our implementation uses raster scanning (left-to-right, top-to-bottom) as the primary traversal pattern, with the bidirectional mechanism ensuring that each pixel can access information from both preceding and succeeding spatial locations, crucial for capturing the complex spatial correlations present in jet substructure.

3.2.3 Architecture Implementation

Our JetVision-Mamba architecture (Figure 3) consists of five main components:

1. **Input Processing:** Multi-channel jet images of shape $(B, 33, 33, 15)$ where B is the batch size and 15 repre-

[3] For continuous variables like p_T and energy, we apply log-transformations to handle their wide dynamic range. For the impact parameters which have small values and are related to how far away the trajectories of the particles are from the point where the two beams collide, we use a tanh transformation for numerical stability. For more details see 4. All other features remain unaltered.

sents the number of particle feature channels.

2. **Initial Embedding:** We use a 2D convolutional layer with 3×3 kernels to project the 15 input channels to 128 feature dimensions, followed by LayerNorm and GELU activation. This is meant to preserve spatial locality while increasing representational power.

3. **2D Mamba Block Stack:** We employ 4-6 2D Mamba blocks, each containing:

   - LayerNorm for input stabilization
   - 2D-to-1D reshape operation using raster scanning
   - Selective SSM with state dimension $d_{\text{state}} = 16$
   - 1D-to-2D reshape to restore spatial structure
   - Residual connection for gradient flow

4. **Global Attention Pooling:** A multi-head attention mechanism with 4 heads aggregates spatial information into a fixed-size jet representation of dimension d_{model}, using learned query vectors to focus on the most discriminative spatial regions.

5. **Classification Head:** A two-layer MLP ($d_{\text{model}} \to d_{\text{model}} \to 10$) with LayerNorm, GELU activation and dropout (rate = 0.1) maps the jet embedding to class probabilities via softmax.

Implementation Details: For the implementation of our model we use the Weaver framework [22], which is a framework for streamlined machine learning applications in HEP based on pytorch [12]. On top of that, we wrote custom GPU-optimized preprocessing functions for the jet image creation. For the selective SSM implementations we used the optimized CUDA kernels from [15, 29].

3.2.4 Training Objective

We optimize the model using cross-entropy loss over the 10 jet flavor classes:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{10} y_j^{(i)} \log p_j^{(i)} \tag{3}$$

where N is the number of jets in a batch, $y_j^{(i)}$ is the one-hot encoded ground truth label for jet i and class j, and $p_j^{(i)}$ is the predicted probability. We use the Ranger optimizer [28] with initial learning rate 5×10^{-4} and cosine annealing schedule, training with mixed-precision (FP16).

4. Dataset and Features

We evaluate our JetVision-Mamba approach on the JetClass dataset [24], a large-scale benchmark specifically designed for jet flavor classification tasks. This section details the dataset characteristics, our preprocessing pipeline, and the feature engineering approach used to convert particle-level information into multi-channel jet images.

4.1. JetClass Dataset Overview

The JetClass dataset contains 125 million simulated jets distributed across 10 physics classes representing different particle decay processes. The dataset is partitioned into 100M training jets, 5M validation jets, and 20M test jets, with equal representation across all classes to ensure balanced learning.

The ten jet classes span the full spectrum of Standard Model processes relevant to LHC physics:

- **Higgs boson decays**: $H \to b\bar{b}$ (Hbb), $H \to c\bar{c}$ (Hcc), $H \to gg$ (Hgg), $H \to 4q$ (H4q), $H \to \ell\nu qq'$ (Hqql)
- **Electroweak boson decays**: $W \to qq'$ (Wqq), $Z \to q\bar{q}$ (Zqq)
- **Top quark decays**: $t \to bqq'$ (Tbqq), $t \to b\ell\nu$ (Tbl)
- **QCD background**: Light quark and gluon jets (QCD)

This classification scheme captures the essential physics signatures used in LHC analyses, where distinguishing signal processes (Higgs, W/Z, top) from QCD background represents the primary experimental challenge.

4.2. Particle-Level Features

Each jet in the dataset contains up to 128 constituent particles, with zero-padding applied for jets with fewer constituents. For each particle, JetClass provides 17 features organized into three physics-motivated categories as shown in Table 1. These features capture the complete kinematic, identification, and tracking information necessary for comprehensive jet characterization.

The **kinematic features** encode the four-momentum information of each particle, including both absolute quantities ($\log p_T$, $\log E$) and relative measures with respect to the parent jet. The angular separation ΔR provides crucial information about the jet's internal structure and collimation.

Particle identification features distinguish between different particle species through detector-based classification algorithms. The charge measurement and particle type flags (electron, muon, photon, charged/neutral hadrons) enable the reconstruction of the underlying physics processes.

Trajectory displacement features capture the impact parameters of charged particles, which are essential for

Table 1: Particle input features used for jet tagging in the JETCLASS dataset. Table adapted from [24].

Category	Variable	Definition
Kinematics	$\Delta\eta$	Difference in pseudorapidity η between the particle and the jet axis
	$\Delta\phi$	Difference in azimuthal angle ϕ between the particle and the jet axis
	$\log p_T$	Logarithm of the particle's transverse momentum p_T
	$\log E$	Logarithm of the particle's energy
	$\log\left(\frac{p_T}{p_T^{(\text{jet})}}\right)$	Logarithm of the particle's p_T relative to the jet p_T
	$\log\left(\frac{E}{E^{(\text{jet})}}\right)$	Logarithm of the particle's energy relative to the jet energy
	ΔR	Angular separation to the jet axis $\left(\sqrt{(\Delta\eta)^2+(\Delta\phi)^2}\right)$
Particle identification	charge	Electric charge of the particle
	Electron	If particle is an electron
	Muon	If particle is a muon
	Photon	If particle is a photon
	CH	If particle is a charged hadron
	NH	If If particle is a neutral hadron
Trajectory displacement	$\tanh d_0$	Hyperbolic tangent of the transverse impact parameter
	$\tanh d_z$	Hyperbolic tangent of the longitudinal impact parameter
	σ_{d_0}	Uncertainty of the transverse impact parameter
	σ_{d_z}	Uncertainty of the longitudinal impact parameter

identifying jets originating from heavy quarks (b and c). The hyperbolic tangent transformation applied to d_0 and d_z provides numerical stability while preserving the discriminative power of these variables.

4.3. Data Preprocessing and Jet Image Creation

Our preprocessing pipeline transforms the unordered collection of jet constituents into structured multi-channel 2D images suitable for computer vision techniques. This process involves several physics-motivated steps designed to preserve spatial relationships while ensuring rotational and translational invariance.

4.3.1 Jet Standardization

Following established practices in jet image analysis [18], we apply a standardization procedure to each jet:

1. **Centering**: We translate each jet to place its p_T-weighted centroid at the origin in the η-ϕ plane, ensuring translational invariance.

2. **Rotation**: We align the principal axis of the jet (determined by the moment of inertia tensor) with a canonical direction, providing rotational standardization.

3. **Reflection**: We apply a consistent reflection to ensure uniform orientation across all jets.

This preprocessing ensures that the network focuses on physically meaningful jet substructure rather than arbitrary geometric orientations.

4.3.2 Multi-Channel Image Generation

We convert each standardized jet into a 33×33 pixel image spanning $\Delta R = 0.8$ in both η and ϕ directions, matching the jet reconstruction cone size. The choice of 33×33 resolution provides sufficient granularity to capture jet substructure while maintaining computational efficiency. Each of the 15 particle features (excluding $\Delta\eta$ and $\Delta\phi$ which become spatial coordinates) forms a separate image channel through 2D histogram binning.

4.3.3 Normalization and Standardization

The JETCLASS dataset provides particles features that are already preprocessed for machine learning applications. Continuous variables like $\log p_T$ and $\log E$ are standardized to zero mean and unit variance across the training set. Impact parameters are transformed using $\tanh$ functions to handle their naturally wide dynamic range. Our jet image creation process preserves these preprocessing choices while adding spatial binning and channel-wise normalization within each image.

No data augmentation is applied, as the physics-motivated standardization procedure already ensures that the network learns translation and rotation invariant representations. Additional augmentation could potentially introduce non-physical correlations that would degrade performance on real experimental data.

5. Results

Below is our evaluation of our JetVision-Mamba approach on the JETCLASS dataset, comparing against state-of-the-art baselines PARTICLENET and PARTICLETRANSFORMER. This section details our experimental setup, evaluation metrics, and provides both quantitative performance analysis and qualitative insights into model behavior.

5.1. Experimental Setup

Training was performed on NVIDIA A100-SXM4-40GB GPUs at the SLAC Shared Scientific Data Facility (S3DF). We trained three JetVision-Mamba variants with different model capacities, as shown in Table 3: JVM-Small ($d_{\text{model}} = 64$, $n_{\text{layers}} = 6$), JVM-Medium ($d_{\text{model}} = 128$, $n_{\text{layers}} = 4$), and JVM-Large ($d_{\text{model}} = 128$, $n_{\text{layers}} = 6$), in order to study the trade-offs between model size and performance.

Hyperparameter Selection: We used the Ranger optimizer with an initial learning rate of 5×10^{-4} and cosine annealing schedule. Training employed mixed-precision (FP16) with batch sizes ranging from 128-512 depending on model size and GPU memory constraints. Given th large dataset size and limited time, models were trained for 2

Table 2: Jet tagging performance on the JETCLASS dataset for the three variants of our JetVision-Mamba (JVM) model, as well as PARTICLENET and PARTICLETRANSFORMER.

	All classes		$H \to b\bar{b}$	$H \to c\bar{c}$	$H \to gg$	$H \to 4q$	$H \to \ell\nu qq'$	$t \to bqq'$	$t \to b\ell\nu$	$W \to qq'$	$Z \to q\bar{q}$
	Accuracy	AUC	$\text{Rej}_{50\%}$	$\text{Rej}_{50\%}$	$\text{Rej}_{50\%}$	$\text{Rej}_{50\%}$	$\text{Rej}_{99\%}$	$\text{Rej}_{50\%}$	$\text{Rej}_{99.5\%}$	$\text{Rej}_{50\%}$	$\text{Rej}_{50\%}$
JVM-Small	0.783	0.9741	3356	585	85	466	291	2053	401	187	161
JVM-Medium	0.784	0.9744	3759	592	84	407	329	2160	478	193	165
JVM-Large	0.785	0.9746	3322	604	85	492	331	2079	434	193	169
ParticleNet	0.844	0.9849	7634	2475	104	953	3339	10526	11236	347	283
ParT	**0.861**	**0.9877**	**10638**	**4149**	**123**	**1869**	**5435**	**32258**	**16129**	**542**	**402**

Table 3: Model variants of JetVision-Mamba (JVM) and their parameters.

	d	n_{layers}
JVM-Small	64	6
JVM-Medium	128	4
JVM-Large	128	6

epochs only and the epoch with lowest validation loss was kept. For regularization, we used dropout rate set to 0.1.

Baseline Comparison: For PARTICLENET and PARTICLETRANSFORMER, we used the official pre-trained models from the official repository [24]. These models were trained on the same JETCLASS dataset using optimized hyperparameters reported in the original publication.

5.2. Evaluation Metrics

We evaluate model performance using standard metrics for multi-class jet tagging:

Multi-class Metrics:

- **Accuracy**: Overall classification accuracy across all 10 jet classe.s
- **Area Under Curve (AUC)**: Macro-averaged AUC across all one-vs-rest binary classifications/[4].

Signal vs. Background Rejection: For each signal class S, we compute the background rejection at fixed signal efficiency:

$$\text{Rej}_{X\%} \equiv \frac{1}{\text{FPR}} \text{ at TPR} = X\%$$

where TPR and FPR are the true positive and false positive rates, respectively. The classification score for signal vs. background discrimination is computed as:

$$\text{score}_{S \text{ vs } B} \equiv \frac{p_S}{p_S + p_{\text{QCD}}}$$

This metric is particularly relevant for HEP applications where distinguishing rare signal processes from overwhelming QCD background is the primary challenge.

5.3. Quantitative Results

Table 2 presents performance comparisons in terms of AUC, accuracy and background rejection across all models and jet classes. Our JetVision-Mamba models achieve competitive performance while maintaining significantly lower parameter counts than transformer-based approaches.

Overall Performance: JVM-Large achieves 78.5% accuracy and 0.9746 AUC, representing satisfactory performance on this challenging 10-class classification task. While trailing the SOTA models PARTICLETRANSFORMER (86.1% accuracy, 0.9877 AUC) and PARTICLENET (84.4% accuracy, 0.9849 AUC), our approach demonstrates the viability of selective state space models for jet classification. Indeed, Fig. 8 demonstrates the improvement in discriminating $H \to b\bar{b}$ from background processes using the JVM-Large classifier.

Signal vs. Background Discrimination: For the important $H \to b\bar{b}$ vs. QCD task, JVM-Large achieves a background rejection of 3,322 at 50% signal efficiency, compared to 10,638 for PARTICLETRANSFORMER and 7,634 for PARTICLENET. While again lower than baselines, this performance is sufficient for many physics applications and could come with significant computational advantages.

Class-Specific Analysis: Performance varies significantly across jet types, reflecting the inherent difficulty of different classification tasks. Heavy flavor jets ($H \to b\bar{b}$, $H \to c\bar{c}$) show reasonable discrimination, while more challenging classes like $H \to gg$ exhibit lower rejection rates across all models, indicating fundamental physics limitations rather than architecture-specific issues.

5.4. Computational Efficiency

Table 4 demonstrates the computational advantages of our approach. JVM-Large achieves comparable inference throughput (1,120 samples/sec) to PARTICLETRANSFORMER (1,290 samples/sec) while using only 802k parameters compared to 2.14M for ParticleTransformer. The linear scaling of Mamba blocks enables efficient processing that becomes increasingly advantageous for larger input sizes.

Parameter Efficiency: Our largest model uses 62% fewer parameters than ParticleTransformer while achieving

[4] For that we used `roc_auc_score` from `scikit-learn` [21] with options `average == 'macro'` and `multi_class =='ovo'`.

Table 4: Number of trainable parameters, FLOPS and inference throughput, as measured in our training setup.

	# params	FLOPs	Inference Throughput [samples/sec]
JVM-Small	227 k	185 M	1250
JVM-Medium	569 k	405 M	1230
JVM-Large	802 k	598 M	1120
ParticleNet	370 k	538 M	740
ParT	2.14 M	342 M	1290

91% of its accuracy, demonstrating favorable parameter efficiency. This reduction is particularly valuable for deployment in resource-constrained environments like real-time trigger systems.

FLOP Analysis: JVM-Large requires 598M FLOPs compared to 342M for ParticleTransformer, reflecting the overhead of 2D image processing and of the SSM blocks. However, the linear scaling properties of Mamba suggest that this gap might narrow for larger input sizes where transformer quadratic complexity becomes prohibitive.

5.5. Qualitative Analysis

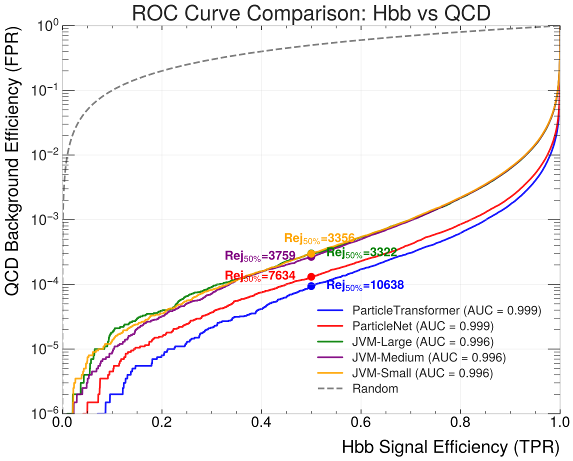


Figure 4: ROC curves for the Hbb vs QCD classification task for the three variations of the JetVision-Mamba models as well as ParticleNet and ParticleTransformer.

ROC Curves:

Figure 4 shows ROC curves for the critical $H \rightarrow b\bar{b}$ vs. QCD classification task across all models. The curves show that the different JetVision-Mamba variants considered have overall inferior performance compared to ParticleNet and ParticleTransformer, although overall all models are able to suppress QCD by more than three orders of magnitude at a signal efficiency of 50%.

Confusion Matrix Analysis:

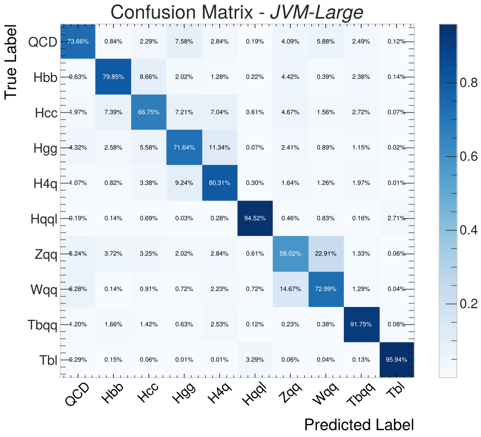


Figure 5: 10-class confusion matrix for the JVM-Large model

Figure 5 presents the 10-class confusion matrix for JVM-Large. The model shows strong diagonal performance with most confusion occurring between physically related classes (specifically Higgs boson and W, Z boson decays). Importantly, the QCD background is overall well-separated from the signal classes, which is important for background suppression.

Score Distributions:

Figure 6 illustrates the output score distributions for JVM-Large. Panel (a) shows the 10-class softmax outputs for true $H \rightarrow b\bar{b}$ jets, demonstrating clear peak structure for the correct class. Panel (b) shows the binary classification scores for $H \rightarrow b\bar{b}$ vs. QCD, revealing good separation between signal and background distributions with limited overlap.

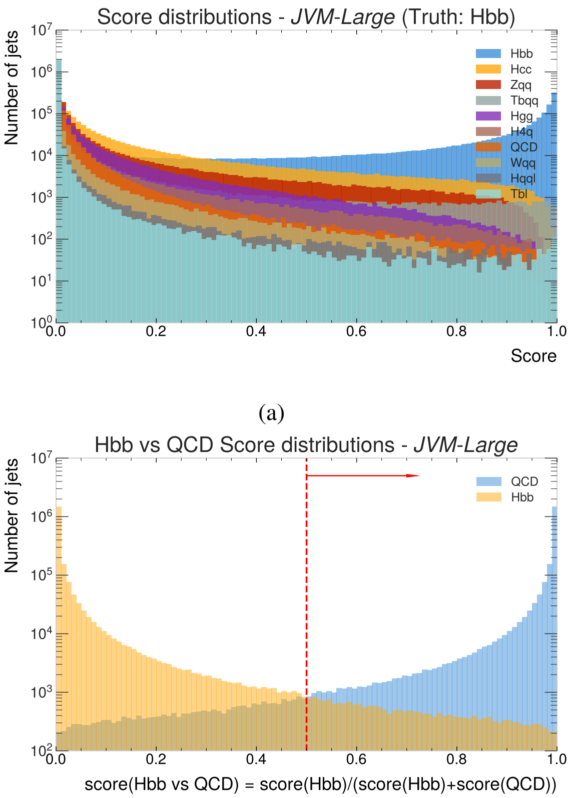
Learned Representations:

Figure 7 presents t-SNE and UMAP visualizations of the learned jet embeddings for JVM-Large, ParticleNet, and ParticleTransformer. The dimensionality reduction reveals that JetVision-Mamba learns well-separated cluster structures for different jet types, though somewhat more ”blurry” and with different geometric arrangements compared to the the other models. The preservation of class structure in the embedding space is another indication that the 2D Mamba architecture is successful at capturing discriminative jet features.

5.6. Architecture Ablation Study

The three JetVision-Mamba variants provide insights into architecture design choices:

Depth vs. Width Trade-offs: Comparing JVM-Medium and JVM-Small with relatively similar parameter counts (569k vs. 227k), we observe that increased model di-



(a)

(b)

Figure 6: (a) Output softmax scores for the ten classes for all true-Hbb jets and (b) binary classification score for Hbb vs QCD for the JVM-Large model

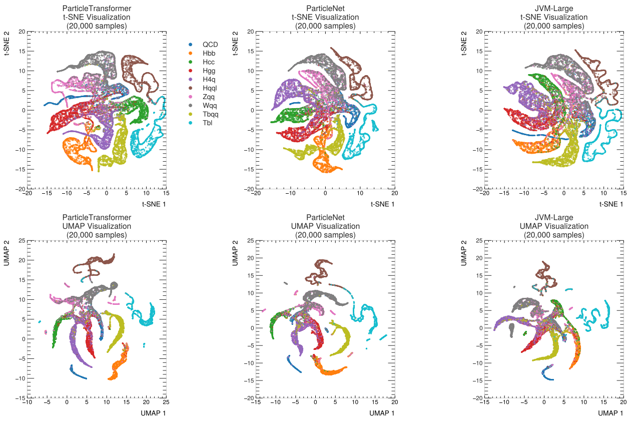


Figure 7: tSNE and UMAP plots for the JVM-Large,PARTICLENET and PARTICLETRANSFORMER models.

mension provides better performance than increased depth for this task. This seems to suggest that, for jet classification tasks, richer feature representations are more valuable than deeper hierarchical processing.

Scaling Behavior: JVM-Large shows modest improvements over JVM-Medium despite increased depth, indicating diminishing returns from additional Mamba layers. This plateau suggests that the jet classification task may not require extremely deep architectures, consistent with the relatively local nature of jet substructure patterns. Alternatively, it could indicate that the performance plateau in the JVM architectures is not related to the depth/width of the models, but to another limitation, namely the jet representation as an image.

5.7. Generalization and Overfitting Analysis

We monitored training and validation performance throughout training to assess generalization. The models show stable convergence without significant overfitting, with validation metrics close to the training ones. An example is given in Fig. 10. This indicates that the use of dropout, LayerNorm, and early stopping appears sufficient for regularization.

Cross-Physics Generalization: The consistent performance across different Higgs decay modes suggests that our approach learns generalizable jet substructure features rather than process-specific artifacts. This is crucial for deploying our model in actual experimental environments where the jet compositions may differ from the training distributions. The performance of the model was also found to be robust across different ranges of the jet transverse momentum p_T, as is shown in Fig. 9.

5.8. Limitations

While JetVision-Mamba demonstrates competitive efficiency and reasonable performance, several limitations merit discussion:

Performance Gap: The 7-8% accuracy gap compared to SOTA methods represents a non-negligible performance difference for physics applications. Future work should explore architectural improvements such as multi-scale processing and enhanced attention mechanisms, as well as alternative jet representations, for instance unordered sets instead of images.

Image Resolution: Our current 33×33 image resolution may limit the capture of fine-grained jet substructure. Higher resolutions could improve performance but require careful optimization to maintain computational advantages.

Scanning Patterns: We primarily used raster scanning for 2D-to-1D conversion. Exploring alternative scanning patterns (spiral, zigzag) or multi-directional approaches could better preserve spatial relationships in jet images, although these might also come with an increased computational overhead.

Despite these limitations, JetVision-Mamba establishes selective SSMs as a viable and efficient approach for jet classification, potentially opening new directions for research in physics-informed deep learning architectures.

6. Conclusions & Future Work

This work presents the first application of selective SSMs to jet flavor classification in HEP, introducing JetVision-Mamba as a novel architecture that combines physics-motivated image representations with efficient sequence modeling. Our approach demonstrates that 2D Mamba models can achieve competitive computational efficiency while maintaining reasonable classification performance.

6.1. Key Contributions and Findings

Our primary contribution lies in unifying between modern sequence modeling techniques and HEP applications by converting unordered jet constituents into structured multi-channel images which are used as inputs for 2D SSM layers. The JetVision-Mamba architecture achieves 78.5% accuracy and 0.9746 AUC on the 10-class JetClass dataset while using 62% fewer parameters than ParticleTransformer. Most significantly, our approach exhibits linear computational complexity compared to the quadratic scaling of transformer-based methods, representing a fundamental advantage for scaling to higher-sequence-length jet representations.

Our results for interpreting the nature of jet classification tasks. The fact that JetVision-Mamba is inferior to ParticleNet and ParticleTransformer could indicate that explicit modeling of particle-to-particle interactions provides superior discriminative power compared to our spatial convolution approach. This indicates that jet substructure is to some extent non-local in nature, meaning that distant particles can have strong physical correlations, benefitting more from flexible interaction modeling than from structured spatial processing.

6.2. Future Research Directions

We can think of several promising routes to address the current limitations of our model. Architectural enhancements represent the most immediate opportunity, where we could think of hybrid approaches that combine Mamba efficiency with explicit particle interaction modeling. Most importantly, it would be worth evaluating the same architectural backbone against a jet representation different than jet images, such as point cloud or particle graph, in order to ascertain the fundamental strengths of the Mamba architecture for jet classification tasks.

Overall, the broader significance of this work lies in demonstrating that modern sequence modeling techniques can be successfully adapted to HEP after considering the domain-specific requirements and constraints. As datasets and computing requirements continue to grow, approaches like JetVision-Mamba that pay closer attention to computational efficiency might become increasingly valuable for enabling offline or even real-time analysis of the vast datasets produced by future particle physics experiments.

References

[1] G. Aad et al. The ATLAS Experiment at the CERN Large Hadron Collider. *JINST*, 3:S08003, 2008.

[2] M. Andrews et al. New directions for surrogate models and differentiable programming for High Energy Physics detector simulation. *Comput. Softw. Big Sci.*, 5(1):12, 2021.

[3] ATLAS Collaboration. ATLAS b-jet identification performance and efficiency measurement with $t\bar{t}$ events in pp collisions at $\sqrt{s} = 13$ TeV. *Eur. Phys. J. C*, 79(11):970, 2019.

[4] E. Bols, J. Kieseler, M. Verzetti, M. Stoye, and A. Stakia. Jet Flavour Classification Using DeepJet. *JINST*, 15(12):P12012, 2020.

[5] S. Chatrchyan et al. The CMS Experiment at the CERN LHC. *JINST*, 3:S08004, 2008.

[6] S. Chatrchyan et al. Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC. *Phys. Lett. B*, 716:30–61, 2012.

[7] CMS Collaboration. Identification of heavy-flavour jets with the CMS detector in pp collisions at 13 TeV. *JINST*, 13(05):P05011, 2018.

[8] J. Cogan, M. Kagan, E. Strauss, and A. Schwarztman. Jet-images: computer vision inspired techniques for jet tagging. *JHEP*, 02:118, 2015.

[9] L. de Oliveira, M. Kagan, L. Mackey, B. Nachman, and A. Schwartzman. Jet-images — deep learning edition. *JHEP*, 07:069, 2016.

[10] L. de Oliveira, M. Paganini, and B. Nachman. Learning particle physics by example: location-aware generative adversarial networks for physics synthesis. *Comput. Softw. Big Sci.*, 1(1):4, 2017.

[11] J. Duarte et al. Graph Neural Networks for Particle Tracking and Reconstruction. *arXiv*, 2021.

[12] A. P. et al. Pytorch: An imperative style, high-performance deep learning library, 2019.

[13] M. Farina, Y. Nakai, and D. Shih. Searching for new physics with deep autoencoders. *Phys. Rev. D*, 101(7):075021, 2020.

[14] A. Gu and T. Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. 2023.

[15] A. Gu and T. Dao. Mamba: linear-time sequence modeling with selective state spaces – code. `https://github.com/state-spaces/mamba`, 2023. GitHub repository, commit accessed 2025-06-04.

[16] D. Guest, K. Cranmer, and D. Whiteson. Deep Learning and its Application to LHC Physics. *Ann. Rev. Nucl. Part. Sci.*, 68:161–181, 2018.

[17] J. Guo, J. Li, and T. Li. Jet tagging in the Lund plane with graph networks. *JHEP*, 03:052, 2021.

[18] P. T. Komiske, E. M. Metodiev, and M. D. Schwartz. Deep learning in color: towards automated quark/gluon jet discrimination. *JHEP*, 01:110, 2017.

[19] G. Louppe, K. Cho, C. Becot, and K. Cranmer. QCD-aware recursive neural networks for jet physics. *JHEP*, 01:057, 2019.

[20] V. Mikuni and F. Canelli. ABCNet: An attention-based method for particle tagging. *Eur. Phys. J. Plus*, 136(8):858, 2021.

[21] F. e. a. Pedregosa. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[22] H. Qu. weavercore package. `https://github.com/hqucms/weaver-core`.

[23] H. Qu and L. Gouskos. ParticleNet: Jet Tagging via Particle Clouds. *Phys. Rev. D*, 101(5):056019, 2020.

[24] H. Qu, C. Li, and S. Qian. Particle Transformer for Jet Tagging. 2 2022.

[25] M. D. Schwartz. TASI lectures on collider physics. In *Theoretical Advanced Study Institute in Elementary Particle Physics: Anticipating the Next Discoveries in Particle Physics*, pages 65–100, 2018.

[26] J. Shlomi, P. Battaglia, and J.-R. Vlimant. Graph neural networks in particle physics. *Mach. Learn. Sci. Tech.*, 2(2):021001, 2021.

[27] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic Graph CNN for Learning on Point Clouds. *ACM Trans. Graph.*, 38(5), 2019.

[28] L. Wright. Ranger: a synergistic optimizer combining radam, lookahead and gradient centralization. `https://github.com/lessw2020/Ranger-Deep-Learning-Optimizer`, 2019. GitHub repository, commit accessed 2025-06-04.

[29] J. Zhang, A. T. Nguyen, X. Han, V. Q. Trinh, H. Qin, D. Samaras, and M. S. Hosseini. 2DMamba: Efficient state space model for image representation – code. `https://github.com/AtlasAnalyticsLab/2DMamba`, 2024. GitHub repository, commit accessed 2025-06-04.

[30] J. Zhang, A. T. Nguyen, X. Han, V. Q.-H. Trinh, H. Qin, D. Samaras, and M. S. Hosseini. 2dmamba: Efficient state space model for image representation with applications on giga-pixel whole slide image classification, 2025.

[31] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang. Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model. 2024.

Appendix

A. Additional plots

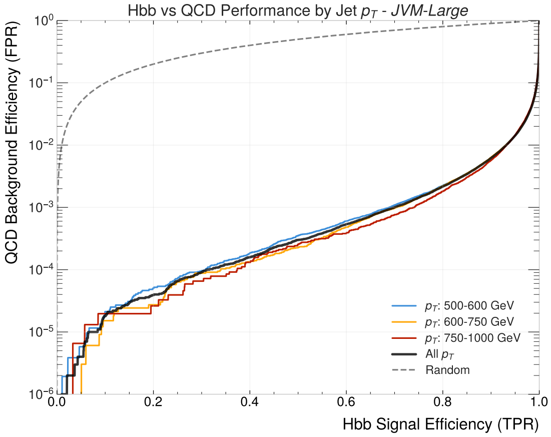


Figure 9: ROC curves for the Hbb vs QCD classification task for the JVM-Large models in bins of the jet transverse momentum p_T.

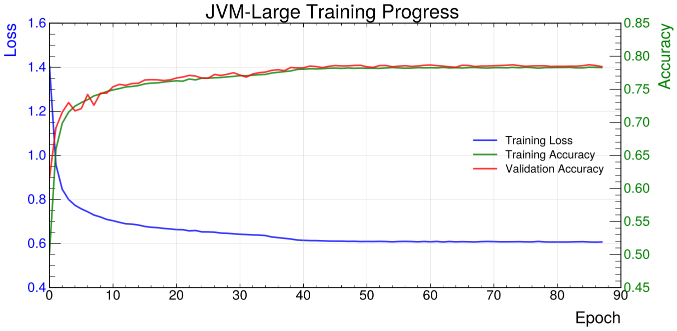


Figure 10: Training loss and training and validation accuracy for the JVM-Large model. For this training one run, one epoch was defined to contain 1.024M samples, such that 98 epochs correspond to one pass over the entire JETCLASS training set.

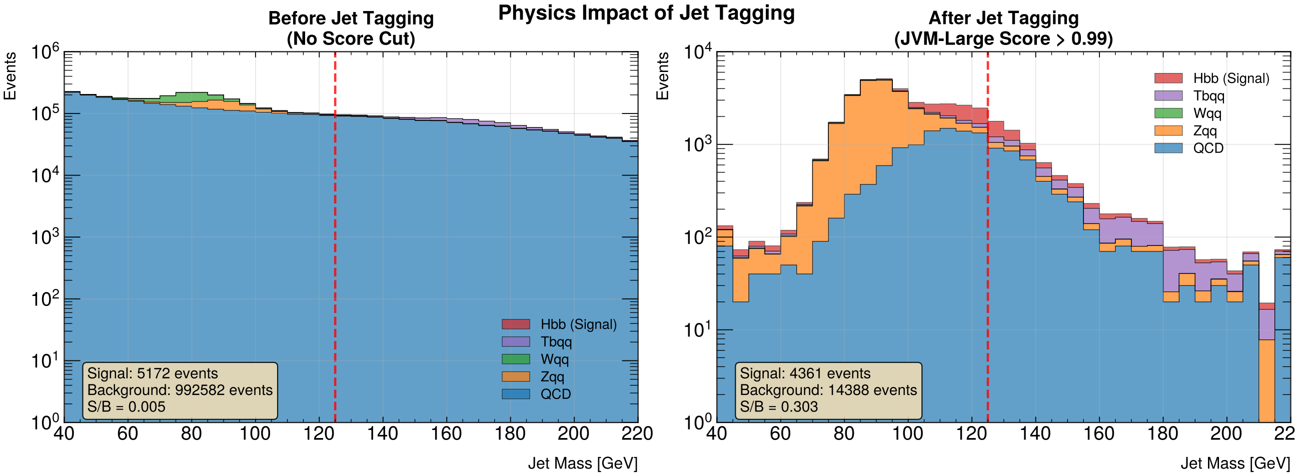


Figure 8: Distribution of the jet mass before (left) and after (right) applying a cut on the JVM-Large Hbb vs QCD binary classification score. For these distributions, all jets from the JETCLASS training set were used, with their yields normalized to the expected number of events expected to be produced after combining the Run 2 (2015-2018) and Run 3 (2022-2025) runs of the LHC.