Karlsruhe Institute of Technology

ETP-KA/2024-24

Event Selection for $B^+ \to K^{*+}\tau^+\tau^-$ Using Transformers at Belle II

Johannes Bertsch

Bachelor Thesis

at the Department of Physics
Institute of Experimental Particle Physics (ETP)
Advisor: Prof. Torben Ferber
Co-Advisor: Dr. Pablo Goldenzweig

14 October 2024 – 31 January 2025

Karlsruher Institut für Technologie
Fakultät für Physik
D-76128 Karlsruhe

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

Karlsruhe, 31 January 2025

. . . . . .

(Johannes Bertsch)

Disclaimer

Data analyses in High Energy Physics such as the measurement presented in this thesis are a collaborative effort. The SuperKEKB particle accelerator which provides the particle beams essential for all studies at Belle II was built and is operated and maintained by the SuperKEKB accelerator group. The Belle II detector was built and is maintained and operated by the Belle II collaboration. The Belle II collaboration also creates the simulated and recorded data sets and maintains the computing infrastructure necessary to process them. The software environment necessary for studies with Belle II data plays an important role and was created and is maintained by the collaboration.

Parts of this thesis are based on the work of Lennard Damer [1], specifically the following contributions:

1. The dataset (see subsection 4.1.1).
2. The performance scores of the FastBDT classifier, used as a benchmark for this work (see section 5.2).
3. The feature importance scores of the FastBDT classifier (see section 4.4).

Additionally, the signal extraction process is performed by Lennard Damer. His contributions are explicitly acknowledged and referenced throughout this thesis whenever his work is utilized.

Statement on the Employment of Techniques based on Artificial Intelligence

This thesis incorporates the use of Artificial Intelligence (AI) tools to help with grammatical or stylistic improvement of text, and program code creation:

1. Grammarly[1] is utilized throughout the thesis for spell and grammar checks, as well as for paraphrasing individual, selected sentences to improve clarity and precision in academic writing. I have approved all suggested changes.

2. GitHub Copilot[2] is used to aid the development of Python code, in particular for debugging, explaining and commenting on existing code, and visualization of data, which does not constitute the core scientific work of this thesis. I have approved and tested all suggestions to provide robust and reliable results. The use of GitHub Copilot has been explicitly acknowledged whenever used.

[1] https://www.grammarly.com/ (accessed on 2nd January 2025)
[2] https://github.com/features/copilot (accessed on 2nd January 2025)

Contents

1. Introduction

The Standard Model (SM) of particle physics is a well-established framework describing the fundamental particles and their interactions, but it fails to account for phenomena like dark matter, the baryon asymmetry of the universe, and neutrino oscillations. The search for new physics includes exploring interactions such as flavour changing neutral currents (FCNCs), which are highly suppressed in the Standard Model, making them a sensitive probe for the SM.

Current observations show a deviation from the standard model at 3.31σ in the ratio $R_{\mathrm{D}^{(*)}}$, described in section 2.2 [2]. This anomaly hints at an enhanced branching ratio of $b \to s\tau^+\tau^-$ processes like the $\mathrm{B}^+ \to \mathrm{K}^{*+}\tau^+\tau^-$ transition [3]. Measuring an enhancement of this branching ratio would provide further evidence for potential New Physics (NP) beyond the SM. Such a result could point to Lepton Flavour Universality (LFU) violation, potentially caused by exotic new particles mediating these processes.

The Belle II experiment is expected to perform well in terms of sensitivity in $\mathrm{B}^+ \to \mathrm{K}^{*+}\tau^+\tau^-$ decays [4], making it a promising tool in the search for the branching ratio of this process.

Experiments like Belle II collect large amounts of data, but only a small fraction of the events correspond to the process of interest. This is especially the case for rare decays like $\mathrm{B}^+ \to \mathrm{K}^{*+}\tau^+\tau^-$. The selection of events is performed by a Machine Learning (ML) model, which is viable due to the complexity and scale of the data. The model is optimized in a supervised training on simulated data. Increasing the performance of ML models for event selection enhances the quality of measurements at particle accelerator experiments without collecting additional data. A more effective event selection process can increase the signal yield, and thus improve the sensitivity of the experiment.

A widely used model for event selection is the Boosted Decision Tree (BDT), which is highly effective at classifying tasks, with a frequently used implementation at Belle II being FastBDT [5], which is a cache-friendly and fast implementation of the BDT.

A recent development in ML is the introduction of the Transformer model by Vaswani et al. [6]. An attention mechanism (see subsection 4.2.2), that regards dependencies within the input data is the foundation of the Transformer. The Transformer shows exceptional performance on natural language processing tasks [7] and is the foundation for tools like the Generative Pre-trained Transformer (GPT) [8].

Transformer-based models have also been applied in High Energy Physics. Examples are the Particle Transformer (ParT) for the task of jet-tagging [9] and the Event Classifier Transformer (ECT) for the task of event classification [10].

In this work, different approaches to utilizing Transformer-based models for classification are tested, with efforts made to improve the performance of current methods, such as FastBDT. Different models are considered, and one model is selected for a detailed performance study. The classification for the $B^+ \to K^{*+}\tau^+\tau^-$ analysis [1], which employs the FastBDT model, is redone using a Transformer-based model on the same dataset, and the results are compared. To allow for a fair comparison, all steps involved in training and hyperparameter optimization are replicated from the original analysis. The sensitivity of the analysis is evaluated by estimating the upper limit of the branching fraction of the process of interest.

In the following chapters, a brief overview of the physical foundations and the experimental status of the $B^+ \to K^{*+}\tau^+\tau^-$ process is provided (chapter 2). Additionally, the Belle II experiment is described (chapter 3). The analysis [1], replicated using a Transformer-based model, is presented, along with a discussion of the different tools used in this work and an explanation of the Transformer (chapter 4). Next, the results of different models tested and evaluated using various evaluation metrics are presented (chapter 5). Finally, the results are summarized, and an outlook is given, along with a discussion of possible next steps and applications (chapter 6).

2. Foundations

The principle of Lepton Flavour Universality (LFU) in the Standard Model of particle physics is the focus of this chapter, along with an exploration of its potential violations as signs of NP. The emphasis in section 2.1 is on the principle of LFU and flavour changing neutral current (FCNC) processes, which are a promising probe of LFU. Section 2.2 explores how recent results hint at an enhancement of $b \to s\tau^+\tau^-$ transitions. In section 2.3, the current experimental status is discussed, referencing results from the BaBar and Belle collaboration. This chapter aims to give a motivation for the search for the $B^+ \to K^{*+}\tau^+\tau^-$ process.

The main ideas and the structure of this chapter follow the Master Thesis of Lennard Damer [1].

2.1. Lepton Flavor Universality (LFU)

LFU is a principle in the SM of particle physics, which states that the three generations of leptons, the electron (e), the muon (μ), and the tauon (τ) possess the same properties apart from their mass differences. From observed decay rates of muons and tau leptons, it is found that all lepton flavors have identical electroweak interaction strengths [11]. LFU is only violated in the SM through the Yukawa interaction after symmetry breaking, because of the different masses of the leptons. A further violation of LFU would point at NP, such as new interactions between quarks and leptons. A method for testing the SM is the study of FCNCs. FCNCs are highly suppressed in the SM, as they are prohibited at tree level, with the lowest order FCNC processes involving a loop (see Figure 2.1).

FCNC processes are further suppressed in the SM given the unitarity of the Cabibbo-Kobayashi-Maskawa (CKM) matrix. Contributions to the interaction strength of different quarks interfere destructively. This is often referred to as the Glashow–Iliopoulos–Maiani (GIM) mechanism [13], historically explaining the suppression of the $s \to d\mu^+\mu^-$ process and successfully predicting the charm quark.

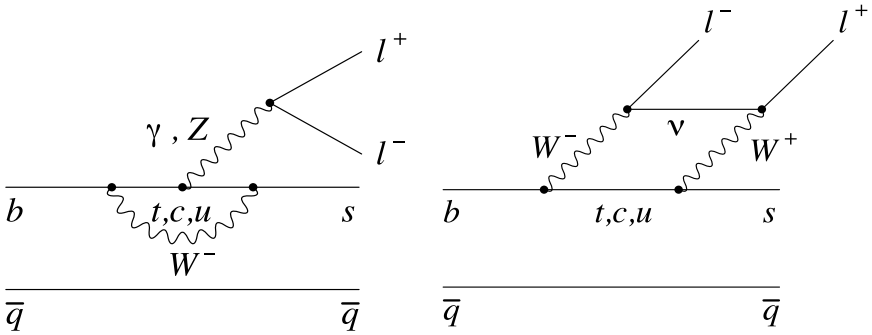


Figure 2.1.: Lowest order Feynman diagrams for the FCNC process $b \to s\tau\tau$. The left diagram is called electroweak penguin, and the right diagram is called box diagram. Both diagrams are strongly suppressed in the SM. Taken from [12].

2.2. Enhancement of $b \to s\tau^+\tau^-$ Transitions

FCNC processes like $b \to s\tau^+\tau^-$ are good test for the SM as they are strongly suppressed. The SM prediction for the branching fractions of $b \to s\tau^+\tau^-$ decays is $O(10^{-7})$ [14]. An enhancement of branching ratios of such processes compared to the SM prediction points at New Physics beyond the Standard Model.

A deviation from the SM prediction of the branching ratio has been measured in the process $b \to c\ell^-\bar{\nu}_\ell$ [15], which is a tree level process mediated by a charged current. To measure the deviation from the SM of this process, the ratio $R_{\mathrm{D}^{(*)}}$ is calculated (see Equation 2.1), and compared to the SM prediction. $R_{\mathrm{D}^{(*)}}$ describes the relative branching fractions $\mathcal{B}$ of the $\mathrm{B} \to \mathrm{X}\tau^-\bar{\nu}_\tau$ and the $\mathcal{B}(\mathrm{B} \to \mathrm{X}\ell^-\bar{\nu}_\ell)$ processes

$$R_{\mathrm{X}} = \frac{\mathcal{B}(\mathrm{B} \to \mathrm{X}\tau^-\bar{\nu}_\tau)}{\mathcal{B}(\mathrm{B} \to \mathrm{X}\ell^-\bar{\nu}_\ell)} \quad , \ell = e, \mu. \tag{2.1}$$

R_{X} has been measured for decays containing the $b \to c\ell^-\bar{\nu}_\ell$ process like $\mathrm{B} \to \mathrm{X}\ell^-\bar{\nu}_\ell$, $\mathrm{X} = (\mathrm{D}, \mathrm{D}^*)$. The deviation of $R_{\mathrm{D}^{(*)}}$ from the SM has been measured at 3.31σ [2].

Capdevila et al. [3] have shown, that, assuming a common NP explanation for the deviation of R_{D} R_{D^*} and, an enhancement of the $b \to s\tau^+\tau^-$ of up to three orders of magnitude is expected under fairly general assumptions. The enhancement could be explained in several extensions of the SM that allow LFU violation in the third generation [4]. A visualization of the correlation between $R_{\mathrm{X}}/R_{\mathrm{X}}^{\mathrm{SM}}$ and the branching ratio of the $B \to K^*\tau^+\tau^-$ process, together with the current measurement of $R_{\mathrm{X}}/R_{\mathrm{X}}^{\mathrm{SM}}$ is shown in Figure 2.2.

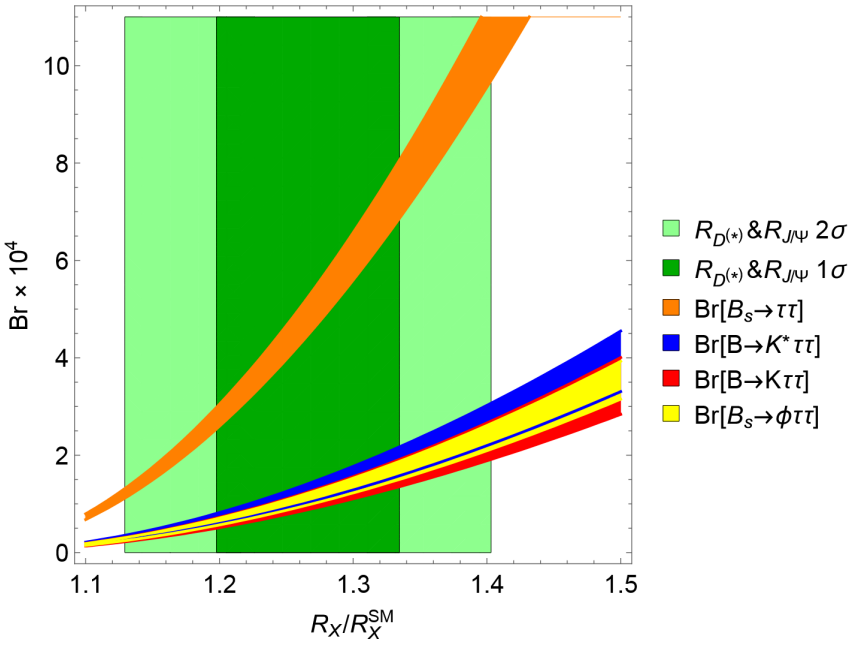


Figure 2.2.: Branching ratios of different $b \to s\tau\tau$ processes (including uncertainties) as a function of R_X/R_X^{SM}. The blue ribbon represents the decay studied in this work, where the branching ratio of the $B \to K^* \tau\tau$ process could be increased by three orders of magnitude. The green ribbon indicates the current experimental range for R_X/R_X^{SM}, obtained by performing the weighted average of R_D, R_{D^*} and $R_{J/\Psi}$. Taken from [3].

This process with two τ leptons in the final state is well suited for the Belle II experiment, because of the low multiplicity of final states compared to hadron-colliders [4], and the ability to reconstruct multiple decay modes of the tau lepton (see Figure 2.3). The enhanced branching ratio of $B^+ \to K^{*+}\tau^+\tau^-$ would be in the observable range of Belle II.

2.3. Current Experimental Status

The search for the $b \to s\tau^+\tau^-$ process has been performed at "B factories" like SuperKEKB and BaBar. B factories are electron-positron colliders operating at the $\Upsilon(4S)$ resonance, where almost exclusively $B\bar{B}$ pairs are produced. In both experiments, one of the B Mesons is reconstructed to set constraints on the other B Meson (signal B Meson). The decay of the signal B Meson is then searched for activity compatible with the process of interest. To identify the τ lepton, which decays very quickly with a mean lifetime of $(290.3 \pm 0.5) \cdot 10^{-15}$ s [16], one has to take into account the different decay modes of the τ lepton (see Figure 2.2) and search for the decay products.

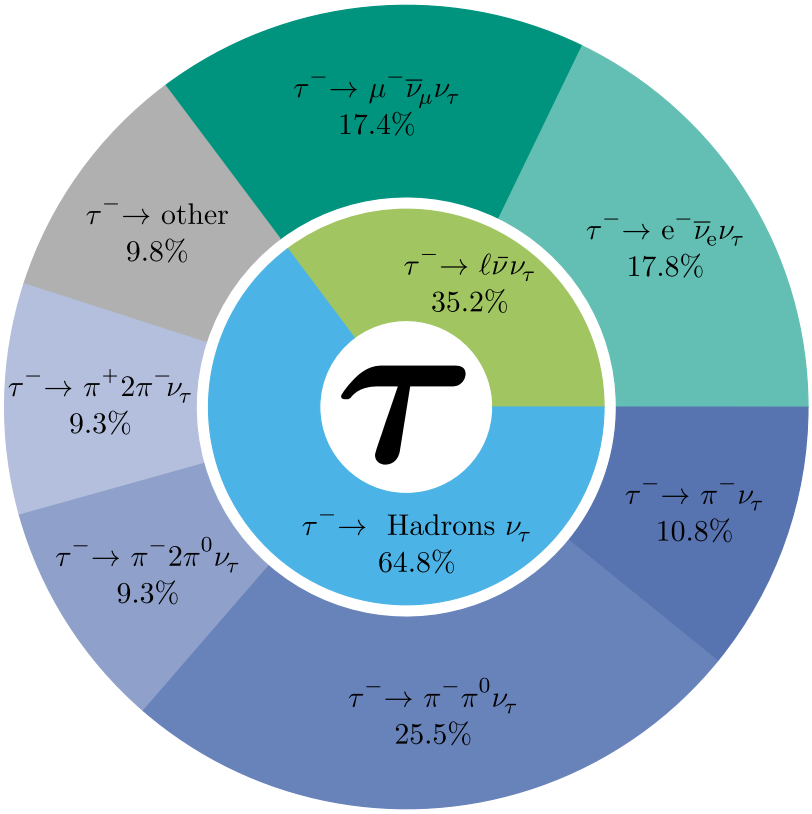


Figure 2.3.: Illustration of the branching fractions of the most common τ decay processes. The green inner circle represents the leptonic τ decays, which are made up of decays into electrons or muons almost equally. The blue inner circle represents hadronic and other decay modes. Adapted from [17].

Search at the BaBar experiment The BaBar collaboration has studied the process $B^+ \to K^+\tau^+\tau^-$ with a data sample, collected at the center-of-mass energy of the $\Upsilon(4S)$ resonance, corresponding to a total integrated luminosity of $424\,\text{fb}^{-1}$ [12]. In its search, the BaBar collaboration only reconstructed τ leptons, that decayed in a leptonic decay $\tau^- \to \ell^- \bar{\nu}_\ell \nu_\tau$ with $\ell = e, \mu$, which makes up 35.2% of all possible τ decays (see Figure 2.2). The BaBar collaboration was not able to find any evidence for a signal, but they were able to find an upper limit for the branching fraction $\mathcal{B}(B^+ \to K^+\tau^+\tau^-) < 2.25 \cdot 10^{-3}$ at 90% CL [12].

Search at the Belle II experiment At Belle II, the process $B^0 \to K^{*0}\tau^+\tau^-$ was studied with a data sample, collected at the center-of-mass energy of the $\Upsilon(4S)$ resonance, corresponding to a total integrated luminosity of $711\,\text{fb}^{-1}$[18]. In contrast to the BaBar search, the $\tau^- \to \pi^- \nu_\tau$ process was included in the reconstruction in addition to the leptonic decay modes, covering 46% of the decay modes. The Belle collaboration also

was not able to find evidence for a signal, but they were able to find an upper limit for the branching fraction $\mathcal{B}(B^0 \to K^{*0}\tau^+\tau^-) < 3.1 \cdot 10^{-3}$ at 90% CL [18].

The SM prediction for the branching fractions of both processes is $\mathcal{O}(10^{-7})$ [14]. The branching fractions could be enhanced by NP effects (see section 2.2), so both results are compatible with such NP effects.

3. The Belle II Experiment

Belle II is a particle physics experiment at the SuperKEKB collider at KEK, which aims to measure the decay of B mesons produced in electron-positron collisions. The goal is to measure parameters to test the SM and probe the existence of new particles [4].

In section 3.1, the SuperKEKB collider, and the production of B Mesons is explained. Section 3.2 discusses the composition of the Belle II detector and its several sub-detectors.

3.1. The SuperKEKB Collider

The SuperKEKB is an asymmetric-energy electron-positron double-ring collider, where 7 GeV electrons are collided with 4 GeV positrons (see Figure 3.1). The beams intersect at the interaction point (IP), which is at the center of the Belle II detector. The energies of the colliding particles result in a center-of-mass (COM) energy of $\sqrt{s} = 10.58\,\text{GeV}$, which corresponds to the $\Upsilon(4S)$ resonance. The $\Upsilon(4S)$ decays into two B mesons with a branching fraction of 96% [16]. Through this process loads of B mesons are produced and SuperKEKB is thus referred to as a B factory. The COM energy just slightly exceeds the $B\bar{B}$ production threshold, so the B mesons are produced almost at rest in the COM frame of reference. This is necessary to observe rare decays like $B^+ \to K^{*+}\tau^+\tau^-$ with sufficient statistics.

The advantage of this collider is, that the B mesons are produced with no additional particles. The background at SuperKEKB is typically smaller than at hadron-colliders due to the low multiplicity of final states and the absence of event pile-up [4]. Because of the asymmetric beam energies, the final state particles are boosted in the lab frame of reference, allowing precise measurements of mixing parameters and lifetimes. This is especially relevant for measuring time-dependent CP violation.

SuperKEKB is designed to reach a luminosity 30 times higher than its predecessor

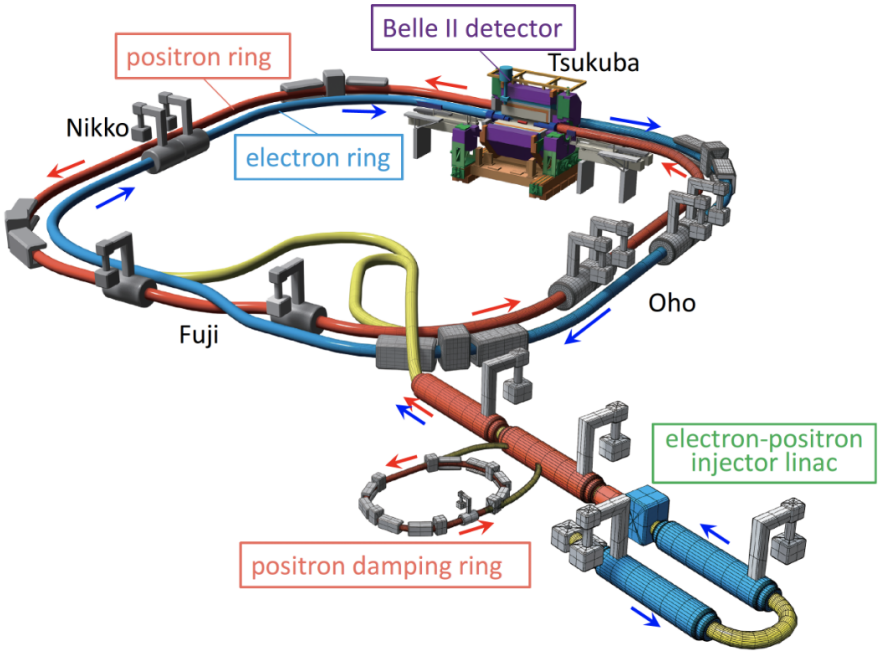


Figure 3.1.: Schematic view of the SuperKEKB Collider. The electrons and positrons are injected into the ring after being accelerated by a linear accelerator. The electron and positron beams collide at the Tsukuba section. Taken from [19].

KEKB by increasing the beam current and shrinking the beam size. The current luminosity record was reached in December 2024 with $\mathcal{L} = 5.1 \cdot 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ [20].

3.2. The Belle II Detector

The Belle II detector has a cylindrical structure centered around the interaction point and is made up of several sub-detectors (see Figure 3.2). Due to the asymmetric beam energies, the detector is built asymmetrically, where the direction of the electrons is called the forward direction, and the direction of the positrons is called the backward direction. The cylindrical region around the IP is called the barrel, and the ends of the barrel are called endcaps. Between the Electromagnetic Calorimeter (ECL) and the K_L^0 / Muon Detector (KLM), there is a superconducting solenoid coil that provides a magnetic field of 1.5 T in the direction along the beam pipe, that causes charged particles to follow a curved trajectory. The particle momentum and the charge of the particle can be determined by measuring the curvature of the trajectories.

The sub-detectors are placed from the innermost Pixel Detector (PXD) to the outer-

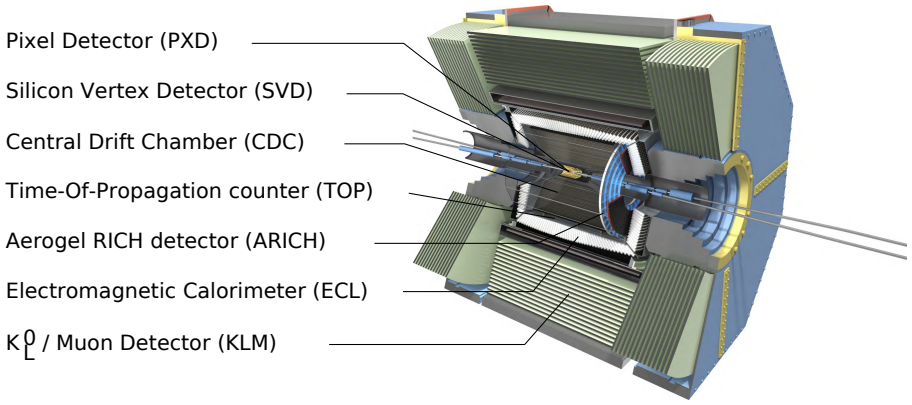


Figure 3.2.: Schematic view of the several sub-detectors of Belle II detector. Taken from [21].

most KLM in a way, that the particles, that are detected by one of the outer detectors, pass through the inner detectors.

Pixel Detector (PXD) and Silicone Vertex Detector (SVD) The innermost detector is used to find the vertex of the decay. The PXD consist of two layers of sensors with with radii at 14 mm and 22 mm [22]. The PXD is followed by the Silicone Vertex Detector (SVD), consisting of four silicone strip detector layers. The purpose of these detectors is to measure the vertices of decays. In the SVD, the energy loss along the trajectory $\frac{dE}{dx}$ is measured. The information is used for particle identification.

Central Drift Chamber (CDC) Inside the CDC there are wires spanned parallel to the beam magnetic field of the solenoid. The CDC is filled with a gas mixture of ethane and helium. Charged particles cause ionization of the gas molecules, and the ionized particles drift towards the wires, where they are measured. This allows the trajectories of the particles and the momenta to be reconstructed.

Time-Of-Propagation Counter (TOP) In the barrel region, there is the Time-of-Propagation Counter (TOP), used for particle identification. The TOP consists of a quartz radiator, where Cherenkov photons are produced when particles with sufficient energy hit. The Cherenkov photons are totally reflected internally and detected using photomultiplier tubes. The emission angle of the Cherenkov photons is reconstructed by measuring the time of propagation of the photons. The velocity of the initial particle is directly linked to the angle of the Cherenkov photons. [22].

Aerogel RICH Detector (ARICH) The Aerogel RICH Detector (ARICH) in the forward endcap is designed to separate kaons from pions over the full kinematic range, i.e. from 0.5 to 4 GeV. The detector consists of an aerogel radiator, where Cherenkov photons are produced by charged particles with sufficient energy, and an array of position-sensitive photon detectors to measure the Cherenkov Photons [22]. The information gathered in the TOP and the ARICH, is used for particle identification.

Electromagnetic Calorimeter (ECL) The ECL consists of CsI(Tl) scintillating crystals. Its main purpose is to detect photons and neutral hadrons, which produce showers within the crystal. The energy of incoming particles is measured with high resolution. The ECL is also used to distinguish between electrons, which deposit all of their energy, and charged hadrons, which deposit only a small fraction of their energy. The polar angle region of $12.4^\circ < \theta < 155.1^\circ$, with a gap of ~$1^\circ$ between the barrel and the endcap region, is covered by the ECL [22].

K_L^0 / Muon Detector (KLM) The KLM consists of a stack of alternating iron plates and active detector elements and is located outside the superconducting solenoid. The iron plates serve as the solenoid's magnetic flux return and provide material in which the K_L^0 mesons shower hadronically [22].

4. Tools and Related Work

A variety of tools are used in this work, most notably the Transformer, which is used as a classifier and tested against the widely used BDT method. In this chapter, the search for $B^+ \to K^{*+}\tau^+\tau^-$ using FastBDT as a classifier is described (section 4.1), which serves as a benchmark for the Transformer-based classifiers. The datasets that are used in this work along with the FastBDT algorithm are discussed. A detailed explanation of the Transformer is given in section 4.2. The application of Transformers on tabular data and in High Energy Physics is examined. In section 4.3, the hyperparameter optimization procedure in this work is described. To make the classifier more explainable, algorithms for measuring the importance of input features are addressed in section 4.4.

4.1. Search for $B^+ \to K^{*+}\tau^+\tau^-$ using FastBDT

This section gives a brief overview of the analysis by Lennard Damer [1].

The classifier's performance in this analysis is used as a benchmark and sanity check for the performance of the Transformer-based model. The dataset and the choice of input features are taken from this analysis. The classifier, the dataset, and the signal extraction strategy are described briefly.

4.1.1. Dataset

Monte Carlo (MC) simulations are used to model the processes within the detector after a collision, with truth information provided by the event generator. The MC data is thus used for training a classifier, that is later applied to separate signal from background processes. The events are generated at the $\Upsilon(4S)$ resonance energy. For validating the MC data, it is compared to the recorded data.

For signal events, a sample of 50 million $B\bar{B}$ events is simulated, where one B meson is restricted to decay into a $K^{*+}\tau^+\tau^-$ product (that further decays generically), while

the other decays generically. For the main physics background events, consisting of $q\bar{q}$ and $B\bar{B}$ events, datasets equivalent to $1\,ab^{-1}$ of integrated luminosity for each background type are used. A summary of the datasets used at $\Upsilon(4S)$ energy is provided in Table 4.1 From the data, the physical process is reconstructed. The data is split into four channels based on the reconstruction of the signal side final states. Several input features for the classifier are constructed, based on the properties of the events. For a detailed description, refer to [1].

Table 4.1.: Simulated dataset at the $\Upsilon(4S)$ energy used in this work. Taken from [1].

Process	Integrated Luminosity [fb^{-1}]	Generated events $\times 10^6$
$B^+ \to K^{*+}\tau^+\tau^-$		50
$e^+e^- \to u\bar{u}$	1000	1586
$e^+e^- \to d\bar{d}$	1000	396
$e^+e^- \to s\bar{s}$	1000	362
$e^+e^- \to c\bar{c}$	1000	1300
$e^+e^- \to \Upsilon(4S) \to B^0\bar{B}^0$	1000	508
$e^+e^- \to \Upsilon(4S) \to B^+B^-$	1000	539

4.1.2. FastBDT

After the reconstruction, the data is passed to a classifier, that separates between signal and background. As a classifier, FastBDT is used, which is a speed-optimized and cache-efficient implementation of the stochastic Gradient Boosted Decision Tree (SGBDT) for event classification [5]. The input features are selected based on their ability to separate between signal and background (see Table A.5).

The current sensitivity of the $B^+ \to K^{*+}\tau^+\tau^-$ analysis using FastBDT by Lennard Damer is $\mathcal{B}(B^+ \to K^{*+}\tau^+\tau^-) < 5.76 \times 10^{-3}$ at 90% Confidence Level.

4.2. Transformers

Transformers, introduced by Vaswani et al. [6], have proven successful for tasks like natural language processing [7] and have also been applied to data analyses in High Energy Physics [9, 10]. The Transformer is based on an attention mechanism (see subsection 4.2.2), which regards dependencies within the input data. The Transformer takes a sequence of a given length as an input. The elements of the sequence are embedded into a multidimensional

embedding space (see subsection 4.2.1). The attention mechanism computes attention scores between the elements of the input sequence to determine the relevance of one part of the input to another. These scores are then used to weigh and update the representation of each element, focusing on the most relevant information for the task (see subsection 4.2.2). The output of the attention mechanism is then passed to a simple Multi Level Perceptron (MLP). For the binary classification task, the output dimension of the MLP is set to one, and a sigmoid function is applied. The sigmoid function

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{4.1}$$

maps any real-valued input to the interval $(0, 1)$. The function is well suited to the task, as it is differentiable, and manipulates the model outputs to be similar to the truth information of the events, which is either zero for a background event or one for a signal event. The fundamental structure of such a Transformer-based classifier is displayed in Figure 4.1.

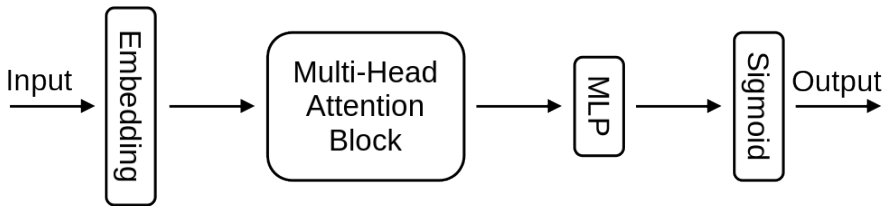


Figure 4.1.: The fundamental architecture of a Transformer used for a classification task.

A special challenge for the application of Transformers in this work is the adaptation of the Transformer on tabular data. Several approaches towards applying Transformer-based networks to tabular data have been made [10, 23–25] (see subsection 4.2.3 and subsection 4.2.4).

4.2.1. Embedding

The embedding feeds the input data to the attention block in such a way, that the attention function is able to process it. The input data is transformed into a multidimensional embedding space, which makes it easier for the model to distinguish between different inputs. The embedding is performed by MLPs for numerical inputs. Categorical input features could also be passed to the attention function by corresponding them to representations in the embedding space via a lookup table. However, there are no categorical features in the dataset studied in this work.

4.2.2. Attention

The attention mechanism is the transformer's core component, which relates different positions of the input sequence to compute a representation of the sequence. Three distinct, independent representations (query Q, key K and value V) of the input data are computed

$$\begin{aligned} Q &= W_{\mathrm{q}}\mathbf{z} + b_{\mathrm{q}}, \\ K &= W_{\mathrm{k}}\mathbf{z} + b_{\mathrm{k}}, \\ V &= W_{\mathrm{v}}\mathbf{z} + b_{\mathrm{v}}, \end{aligned} \tag{4.2}$$

where $\mathbf{z}$ is the embedded input data, and the matrix W and the vector b are composed of trainable weights. Q, K, and V are passed to the attention block. In particular, the "Scaled Dot-Product Attention" (see Figure 4.2) is used, where the input consists of queries Q and keys K of dimension d_{k} and values V of dimension d_{v}. The compatibility between query Q and key K is determined by computing the dot-product QK^T. The dot product of Q and K is scaled by $1/\sqrt{d_{\mathrm{k}}}$ for numerical stability and a softmax function is applied to obtain the weights on the values V

$$\mathrm{softmax}(\mathbf{x})_i = \frac{e^{x_i}}{\sum_j e^{x_j}}. \tag{4.3}$$

The output of the attention function is computed as a weighted sum of the values V, where the weights assigned to the values V are determined by dot-product QK^T [6].

The output of the attention function is computed as

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_{\mathrm{k}}}}\right)V. \tag{4.4}$$

There are adaptations to the basic attention function, that aim to increase performance, like the Multi-Head Attention (MHA) and the class attention:

Multi-Head Attention (MHA) Instead of using a single attention function, the input is split into h different parts (heads), and the attention is computed for each head independently in parallel. The MHA allows the model to focus on different relationships within the input at once [6]. A visualization of the MHA is shown in Figure 4.2.

Class Attention The class-attention block was introduced by Touvron et al. [26] to improve the performance of transformers in image classification. It has a structure similar to the ordinary attention block, but a global class token x_{class} is introduced. The class token x_{class} serves as a summary representation of the entire input data, that combines

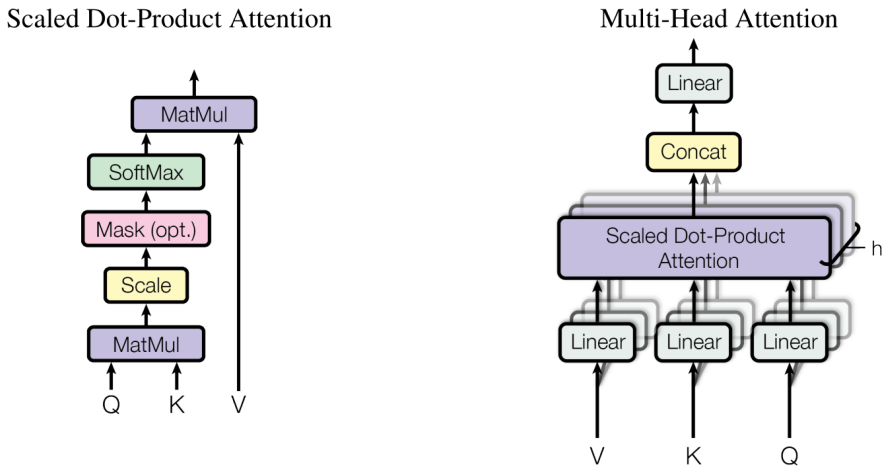


Figure 4.2.: (left) Scaled Dot-Product Attention computes attention scores by taking the dot product of query Q and key K, scaling by the square root of the key dimension d_k, and applying a softmax. The output is found by a weighted sum of the values, with the weights being the attention scores. (right) Multi-Head Attention extends this by applying multiple attention mechanisms in parallel, allowing the model to focus on different relationships within the input at once, before concatenating and projecting the outputs. Taken from [6].

information from all other tokens in the sequence through the attention mechanism. The attention between x_{class} and all tokens $\mathbf{z} = [x_{\text{class}}, x_{\text{input}}]$ is computed. The inputs to the MHA are

$$\begin{aligned} Q &= W_{\text{q}} x_{\text{class}} + b_{\text{q}}, \\ K &= W_{\text{k}} \mathbf{z} + b_{\text{k}}, \\ V &= W_{\text{v}} \mathbf{z} + b_{\text{v}}. \end{aligned} \tag{4.5}$$

This method aims to improve the performance in tasks like classification and is applied in different models like Particle Transformer [9], FT-Transformer [24] and Event Classifier Transformer [10].

4.2.3. Transformers on Tabular Data

The state-of-the-art for processing tabular data is tree-based ensemble methods like BDTs, whereas tasks like image- or natural language processing are performed best by deep learning techniques [23]. BDTs have several advantages, such as high prediction accuracy, fast training, and interpretability (feature importance). Nevertheless, attempts

have been made to apply Transformer-based models to tabular data [23–25]. The main challenge is that Transformers are designed for processing sequences (e.g. sentences in natural language processing), instead of tabular data. To make the input accessible to the Transformer, the data is embedded into a multidimensional embedding space (see subsection 4.2.1).

Several studies have demonstrated, that Transformer-based models, like the TabTransformer [23], the FT-Transformer [24], or the TabNet [25], show similar and in some cases slightly better performance than BDT models on a variety of different datasets. The TabTransformer and the FT-Transformer follow the general architecture depicted in Figure 4.1 with slight adaptations, while the TabNet is structured differently.

The models are tested in section 5.2.

TabTransformer The self-attention block is used, where categorical features are embedded using a lookup table, and numerical features are passed to the attention block directly [23].

FT-Transformer The network incorporates both self-attention and class-attention blocks. The embedding is performed by a linear layer for numerical input features and by a lookup table for categorical features [24].

TabNet TabNet uses an attention mechanism to dynamically select the most important input features for each sample at each decision step. By processing features sequentially through a series of decision steps, the model ensures that only the most relevant features are prioritized for classification. This approach improves model efficiency and interpretability by identifying feature importance for each prediction, enabling insights into the decision-making process [25].

4.2.4. Application of Transformers in High Energy Physics

The prospect of improving the performance on classification tasks by employing Transformers in High Energy Physics has led to the proposal of models designed specifically towards their application in High Energy Physics like the ParT [9] or the ECT [10].

Particle Transformer (ParT) The ParT [9] is a transformer-based model made for jet-flavor tagging. What makes this model unique is that the attention score is augmented by an interaction matrix, that is derived from the energy-momentum 4-vectors of the particles. The ParT has shown promising results in terms of performance, performing

better than state-of-the-art methods [9]. However, the ParT is not tested in this work, as its use case differs from the task studied in this work.

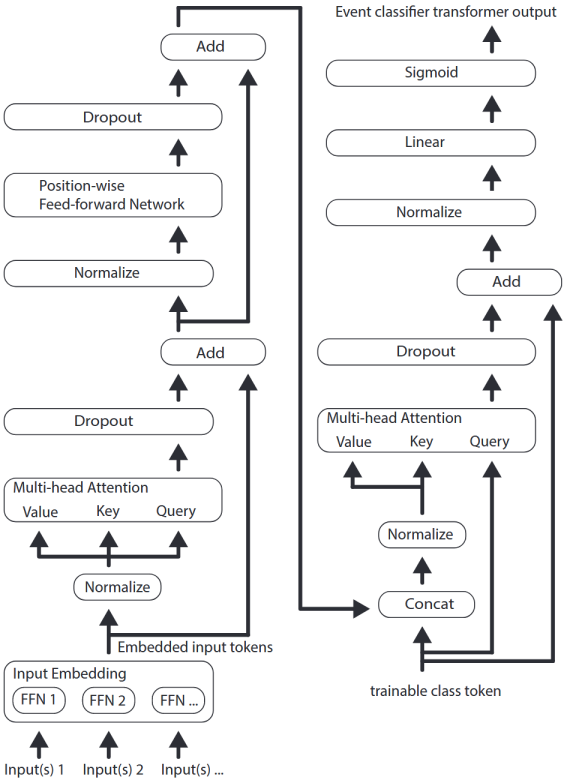


Figure 4.3.: Model architecture of the Event Classifier Transformer. The model integrates an embedding layer, self-attention and class attention layers with residual connections, FFNs, normalization layers, and linear layers. A sigmoid function ensures, that the model output is a value between zero and one, that suits the classification task. Taken from [10].

Event Classifier Transformer (ECT) The ECT [10] is a Transformer-based model designed for the separation of signal and background in High Energy Physics. The structure of the model is depicted in Figure 4.3. The input features are embedded by feedforward neural networks (FFNs) and passed to an MHA block after normalization. The output is summed to make a residual connection [27]. A position-wise FFN with normalization, dropout, and a residual connection is applied to each token. A class-MHA block (see subsection 4.2.2) is applied, again utilizing a residual connection. The output is passed to a linear layer with the output dimension one, and a sigmoid function is used to make the model suitable for a classification task. The model is tested in section 5.2.

4.3. Hyperparameter Optimization

A ML model has many tuneable parameters, called hyperparameters, that determine the structure of the model (e.g. the number and dimension of hidden layers) and the optimization process (e.g. the learning rate or the size of the data batches used for optimization). Optimizing those hyperparameters is a crucial task necessary for obtaining optimal results. In this work, the Optuna framework is used for this task [28]. The hyperparameter search spaces for the different tests conducted are shown in the appendix A. During the hyperparameter optimization, different hyperparameter settings are chosen, and the generalization error is estimated in the validation, by computing the loss of the validation set. The loss function used in this work is the Binary Cross Entropy [29]. There are two different possible validation strategies:

Validation using a Validation Set The training data is split into the nonoverlapping training and validation sets. The model is trained on the training set, and evaluated on the validation set, to estimate the generalization error [29]. Using a validation set further has the advantage, that the validation score is accessible during the process of training, allowing the use of a pruning algorithm and an early stopping algorithm, as described below. However, the validation set cannot be used for the training, making it only feasible on large datasets.

k-fold Cross Validation The data is split into k equally sized, nonoverlapping subsets ("folds"). The model is trained with $k - 1$ folds serving as the training set and the remaining set serving as the validation set. This process is repeated k times, each time using a different validation set. The k different results are averaged to get a performance metric. This algorithm takes considerably more computational cost, but has the advantage, that the the model is trained on a larger portion of the data [29]. However, the evaluation score is only accessible, once the model has finished training on all folds, making the use of pruning and early stopping algorithms impossible.

In Optuna, a sampling algorithm is used to narrow down the search space based on the record of suggested hyperparameters and the corresponding validation results. In this work, the `TPESampler` is applied [28].

In case the validation is performed using a validation set, pruning and early stopping algorithms are used. Optuna offers pruning algorithm stops unpromising trials at an early stage of training to decrease the runtime [28]. The early-stopping algorithm is used, that stops the training preemptively when the validation score does not improve for a given

number of epochs (patience). The weights of the iteration yielding the best validation score are saved. Both algorithms need to access the validation score during the training (e.g. after every epoch), making them incompatible with the k-fold cross validation.

4.4. Feature Importance

The feature importance measures the impact each input feature has on the model's output. Analyzing the feature importance provides valuable insights into the model and the input data. The choice of which input features should be included in the training of a model is based on its importance. Removing noisy features that do not separate well between signal and background can decrease overfitting and increase performance.

FastBDT offers two different methods for determining the feature importance. The intern feature importance algorithm works by summing up the separation gain, a quantity inherent to decision trees, of each feature by looping over all trees and nodes. This algorithm does not apply to Transformer-based networks because of the fundamental differences between the models.

The extern feature importance algorithm works by measuring the drop in performance (the AUC score, described in section 5.1) of the model if one feature is left out. This requires N fit operations, where N is the number of features. The accuracy of the feature importance can be further improved by recursively removing the most important feature, requiring $N(N+1)/2$ fit operations. This algorithm is model-agnostic and would work on any model. However, it is not a viable option for Transformer-based models, because of the long training time.

To compute the feature importance of the Transformer-based model, an adaptation to the extern feature importance algorithm is made. The model is trained once with all input features included and evaluated using the AUC score. The feature importance score of an input feature is found by measuring the drop in performance (the AUC score) after the values of this input feature are set to their mean value. This requires evaluating the model N times, where N is the number of input features. No re-training of the model is needed, which makes it a fast and computationally inexpensive method.

5. Evaluation of Classifier Performance

The task of the classifier is to distinguish between signal and background events. To compare the performance of different models, evaluation metrics described in section 5.1 are utilized. Different Transformer-based models are considered for the classification of the dataset (section 5.2), and one model is chosen for a detailed performance study. This model is used to redo the classification of the $B^+ \to K^{*+}\tau^+\tau^-$ analysis [1], where the classification was done using the FastBDT. To allow for a fair comparison, all steps in the process of training, optimizing hyperparameters, and setting constraints on overfitting are replicated from the analysis. To estimate the sensitivity of the analysis, a maximum likelihood scan is performed and compared to the results of the FastBDT. The feature importance algorithm is applied and compared to the results of the feature importance algorithm of FastBDT [5].

5.1. Evaluation metrics

Evaluation metrics are used to quantify the performance of the classifiers. The classifier, as characterized in Figure 4.1, returns an output value between zero and one for each input. Ultimately a threshold is chosen, so that outputs greater than the threshold are classified as signal and outputs less than one are classified as background. The ROC curve describes the behavior of the model at different thresholds. Finally, the classified data is used to find the sensitivity of the analysis.

5.1.1. The ROC Curve

The Receiver Operating Characteristic (ROC) curve is used to monitor the performance of a classifier at different thresholds. The characteristics of the model, i.e. the true positive rate (TPR) and the false positive rate (FPR), depend on the threshold. The TPR (signal efficiency),

the FPR (background efficiency) and the background rejection (Rej) are computed as

$$\text{TPR} = \frac{\text{TP}}{\text{P}} \tag{5.1}$$

$$\text{FPR} = \frac{\text{FP}}{\text{N}} \tag{5.2}$$

$$\text{Rej} = 1 - \text{FPR}, \tag{5.3}$$

where TP (true positives) is the number of truly classified signal events, FP (false positives) is the number of falsely classified background events, P is the total number of total signal events, and N is the total number of background events.

Scanning over different thresholds and computing the signal efficiency and the background rejection returns the ROC curve. The ROC curve is computed for the training set and the test set separately, where the ROC curve of the test set shows how well the model generalizes to unfamiliar data. A difference between the ROC curves of the training set and the test set suggests that the model may have overfitted. Overfitting occurs when the model learns the training data too well, including its noise and details, leading to poor generalization to unfamiliar data. An example of a ROC curve is given in Figure 5.1.

In this work, the ROC curves are computed using the `roc_curve` function from sklearn [30]. Two different evaluation metrics derived from the ROC curve are described below.

The Area under the ROC Curve (AUC) A metric to describe the performance is the AUC, which is computed by integrating over the ROC curve.

$$\text{AUC} = \int_0^1 \text{TPR}\, \text{dFPR} \tag{5.4}$$

A random classifier would get an AUC score of 0.5 and a perfect classifier would get an AUC score of 1.

In this work, the AUC is computed using `auc` from sklearn [30], which uses the trapezoidal rule for integration.

Background Rejection at a certain Signal Efficiency The AUC is a good metric to evaluate the overall performance of a classifier but fails to capture the performance of the classifier in specific regimes of signal efficiency. To provide a metric, that captures the performance in at specific signal efficiencies, the background rejection at a given

signal efficiency is computed

$$\mathrm{Rej}_X = 1 - \mathrm{FPR \text{ at } TPR} = X. \tag{5.5}$$

A random classifier would get a Rej_X score of $1 - X$ and a perfect classifier would get a Rej_X score of 1.

5.1.2. Sensitivity

The ultimate goal of this work is to reduce the upper limit on the measurement of the branching fraction of the $B^+ \to K^{*+}\tau^+\tau^-$ process, by improving the signal yield. The signal extraction strategy using a template maximum likelihood fit is described in detail in [1]. The resulting estimated sensitivity of the Transformer-based model is compared to the sensitivity of the FastBDT.

5.2. Evaluation of Transformers as Classifiers

As a first step, different Transformer-based models adapted to tabular data are applied to the dataset described in subsection 4.1.1 and compared to the FastBDT performance using the AUC score. The goal is to understand different Transformer-based approaches to classification tasks, assess their performance, and identify a potentially suitable model. Ultimately, a model is selected to redo the classification performed by the FastBDT. Four distinct models are evaluated as potential alternatives for redoing the classification task, replacing the original FastBDT model. One model is chosen to be integrated into the workflow of the $B^+ \to K^{*+}\tau^+\tau^-$ analysis [1], replacing the FastBDT. The sensitivity of the resulting analysis is estimated and compared to that of the original analysis using FastBDT. Additionally, the feature importance algorithm is applied to the Transformer-based model used in the classification task, and the resulting feature importance rankings are analyzed and compared to the FastBDT feature importance rankings.

5.2.1. Selection of a Model

To choose one model for re-doing the classification of the $B^+ \to K^{*+}\tau^+\tau^-$ analysis using FastBDT [1], different Transformer-based classifier models are considered. The models under consideration are described in subsection 4.2.3 and subsection 4.2.4. A simplified process of training and hyperparameter optimization, which is described in detail in the appendix A.1, is used to test the models on the dataset. The difference in the

hyperparameter optimization process is that a validation set is used instead of k-fold cross-validation, reducing computational cost but limiting the use of the full dataset for training and testing. Additionally, no constraints are set to reduce overfitting. Because of these differences, the results of this comparison have to be considered with caution. The purpose of this comparison is to show that Transformer-based classifiers apply to the event classification task and to give a suggestion for the choice of a model to investigate further in subsection 5.2.2.

The models are compared using the AUC score of the test set, which is not used in the training of the models, so the ability of the model to classify unfamiliar data is tested. The results of this comparison are displayed in Table 5.1.

Table 5.1.: Comparison of AUC values between different models. The different Transformer-based models are trained as described in the appendix A, using a simplified process of training and hyperparameter optimization, and evaluated using the AUC value of the test set of their respective ROC curves, which are shown in Figures A.1, A.2, A.3 and A.4. The results are compared to the AUC values of the FastBDT (from [1]). The highest AUC values are highlighted. The comparison has to be considered with caution, as the process of training and hyperparameter optimization of the Transformer-based models differs from the process of applying FastBDT. The code creation for this comparison is aided by the AI programming assistant GitHub Copilot.

	Test AUC value			
Model	$\ell\ell$-channel	$\ell\pi$-channel	$\pi\pi$-channel	ρ-channel
FastBDT	**0.887**	0.877	0.928	0.915
TabTransformer	0.873	**0.893**	0.926	**0.922**
TabNet	0.859	0.881	0.926	0.918
FT-Transformer	0.837	0.877	0.921	0.905
ECT	0.871	0.885	**0.931**	0.919

The result demonstrates that Transformer-based models show only minor differences to the FastBDT concerning the test AUC value, performing even slightly better on three out of four channels. The difference in performance between different Transformer-based models is small and might be caused by statistical fluctuation, or the choice of the hyperparameter search space.

The model chosen for further investigation is the ECT, as the model is designed for the same application [10] and has shown good performance in the comparison.

5.2.2. Training of the Event Classifier Transformer (ECT)

The ECT model[1] (see subsection 4.2.4) is chosen for a detailed performance study, where the settings and methods for optimization are replicated from the $B^+ \to K^{*+}\tau^+\tau^-$ analysis using FastBDT [1].

The data is split into a training set (70% of the data) and a test set (30% of the data). As a loss function, the Binary Cross Entropy Loss[2] is used, where weights are applied to account for the imbalance between the size of the signal and the background set:

$$W_i = \frac{N_{\text{tot}}}{N_{\text{classes}} \cdot N_i}, i \in \{\text{signal}, \text{background}\}, \tag{5.6}$$

where N_{classes} is 2, N_{tot} is the total number of events and N_{signal} and $N_{\text{background}}$ are the number of signal and background events. For the training process, the Adam optimizer [31] is utilized, and the entire dataset is processed across several epochs. The dataset is split into batches within each epoch to improve memory efficiency and speed. The Hyperparameter optimization is performed using Optuna (see section 4.3), using the k-fold cross validation with k=3, with the objective value, that is minimized, being the Binary Cross Entropy Loss of the test set. The search space for the hyperparameter optimization is specified in Table A.4. No pruning algorithm is applied, as it is incompatible with the k-fold cross validation. To visualize, how the value of individual hyperparameters impacts the objective value, the optimization history is shown in Figure A.5. Constraints are set, to disregard trials with unintentional behavior. One constraint is set to avoid overfitting, where the model does not generalize well to unfamiliar data in the test set:

$$\frac{|\text{AUC}_{\text{test}} - \text{AUC}_{\text{train}}|}{\text{AUC}_{\text{test}}} < 0.01. \tag{5.7}$$

Another constraint is set to make sure, the data is well represented in different folds during the k-fold cross validation, and the spread of AUC_{test} between different folds is small:

$$\frac{max(\text{AUC}_{\text{test}}) - min(\text{AUC}_{\text{test}})}{min(\text{AUC}_{\text{test}})} < 0.05. \tag{5.8}$$

[1] implementation adapted from: https://github.com/jaebak/EventClassifierTransformer (accessed on 06th November 2024)

[2] documentation: https://pytorch.org/docs/stable/generated/torch.nn.BCELoss.html (accessed on 15th January 2025)

Using the optimized hyperparameters, the models are trained for the entire training set for the final time. The Binary Cross Entropy Loss of the training set during the training is displayed in Figure A.6.

5.2.3. Evaluation of the ECT and Comparison to FastBDT

After the optimization of the model is completed, the result is evaluated using the metrics described in section 5.1. The model is used to classify the dataset into signal and background, and the results are plugged into the analysis pipeline, where the sensitivity is computed and compared to the FastBDT result.

The ROC curves for the ECT model are presented in Figure 5.1, alongside the FastBDT ROC curves for comparison. Additionally, the AUC values and the background rejection at TPR = 0.5 are shown in Table 5.2. In terms of performance, the ECT model exhibits results comparable to those of the FastBDT. An interesting aspect of this comparison is the difference in the test AUC value between ECT and FastBDT relative to the number of input features (see Figure 5.2). A better performance of the ECT is observed on datasets with a higher number of input features. This trend should be interpreted carefully and studied further, as the absolute differences are small and could be influenced by other factors, such as the number of events in each dataset.

Table 5.2.: Comparison of the performance of FastBDT and ECT. As evaluation metrics, the AUC value and the background rejection at TPR=0.5 are used. The evaluation metrics are computed on the test set, making them a probe for how successfully the model generalizes to unfamiliar data. The code creation for this comparison is aided by the AI programming assistant GitHub Copilot.

	$\ell\ell$-channel		$\ell\pi$-channel		$\pi\pi$-channel		ρ-channel	
Model	AUC	$\text{Rej}_{0.5}$	AUC	$\text{Rej}_{0.5}$	AUC	$\text{Rej}_{0.5}$	AUC	$\text{Rej}_{0.5}$
FastBDT	**0.887**	0.930	**0.877**	**0.937**	0.928	**0.956**	**0.915**	0.950
ECT	0.878	**0.931**	0.872	0.931	**0.929**	0.953	0.914	**0.952**

The classifier returns a value between zero and one for every input. In Figure 5.3, the classifier outputs of the signal process and the different background processes are shown. As expected, the model outputs values close to zero for most background events and values close to one for most signal events. For the model to work as a classifier, a threshold between zero and one is chosen, to define a signal region. The choice of the signal region impacts both the FPR and the TPR and thus affects the result of the analysis. To find the

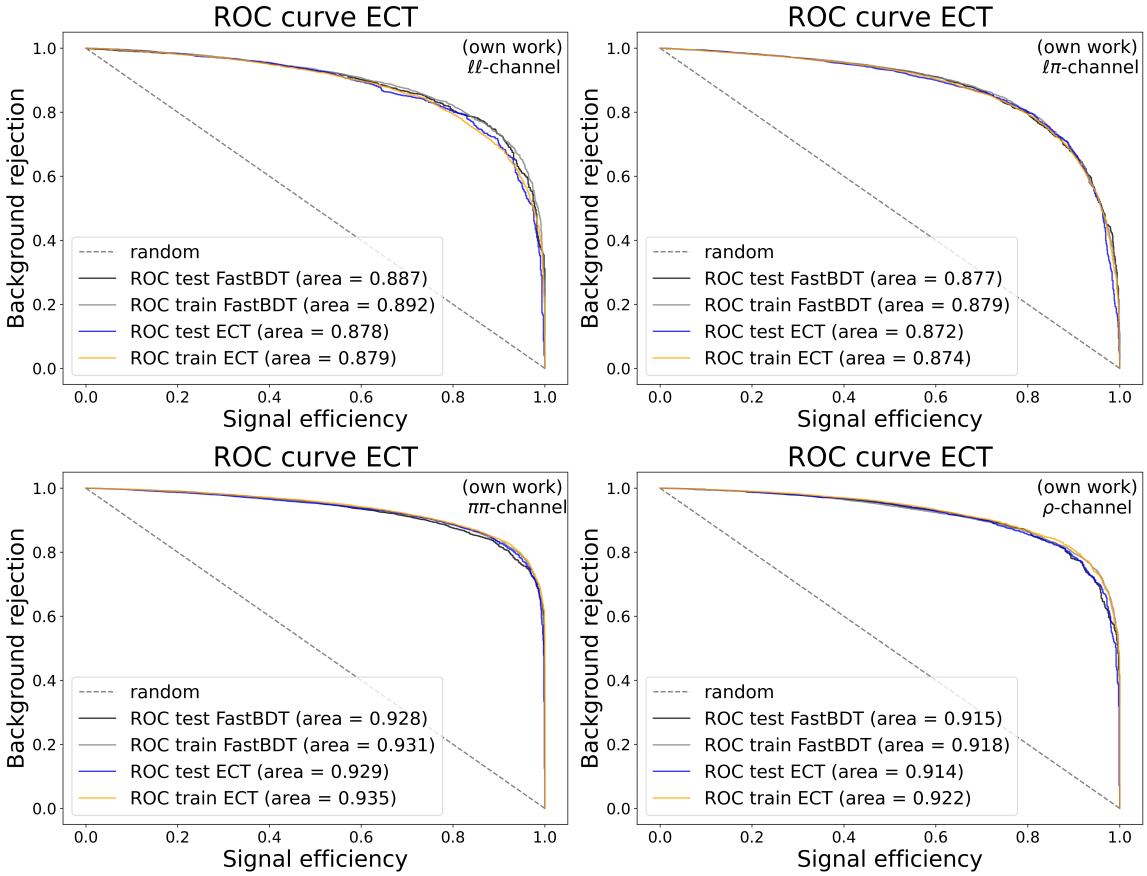


Figure 5.1.: Comparison of ROC curves between FastBDT (from [1]) and ECT on all four channels. There is no major deviation between the FastBDT and the ECT ROC curves.

threshold yielding optimal sensitivity, the Punzi Figure of Merit (FoM) [32]

$$\text{FoM} = \frac{\text{TPR}}{3/2 + \sqrt{N_B}} \tag{5.9}$$

is computed, with a desired significance of 3σ (as in [1]). N_B denotes the total background yield at a given threshold. The Punzi FoM is calculated for different thresholds, and the threshold yielding the highest FoM value is chosen. In Figure 5.4, a scan of FoM values for possible thresholds between zero and one is shown, where the highest FoM is used to select the threshold. Figure 5.5 shows the signal and background efficiency at different thresholds along with the threshold selected using the FoM. The selected threshold does not lie in a region of rapid signal efficiency change, ensuring stable classification.

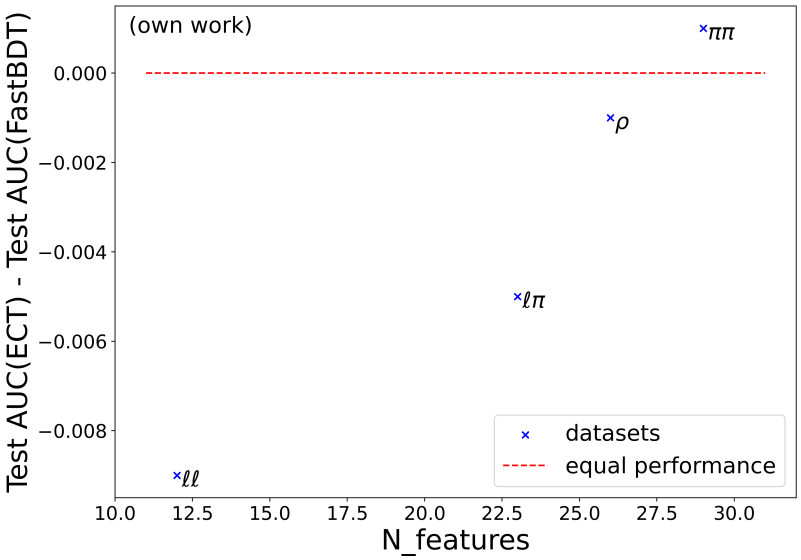


Figure 5.2.: Difference in the test AUC value between the FastBDT and the ECT plotted against the number of input features shown in Table A.5. The plot shows that there is a slight increase in performance of the ECT relative to the FastBDT in terms of test AUC, with a higher number of input features.

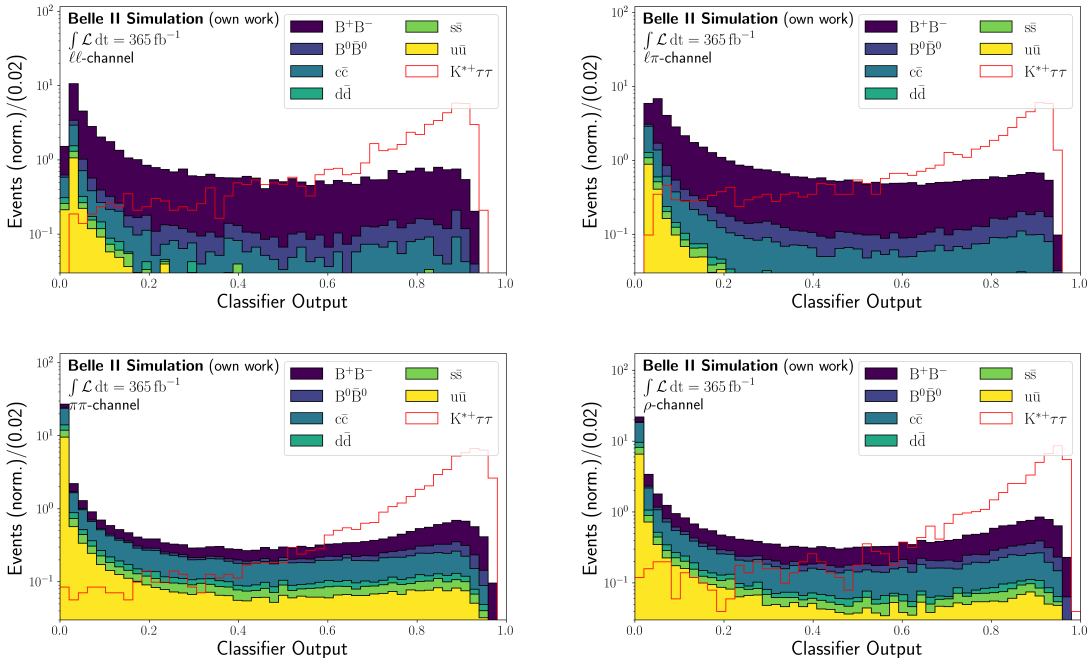


Figure 5.3.: Classifier outputs of the signal process and the different background processes of all four channels. A perfect classifier would assign outputs close to one for signal processes and outputs close to zero for background processes.

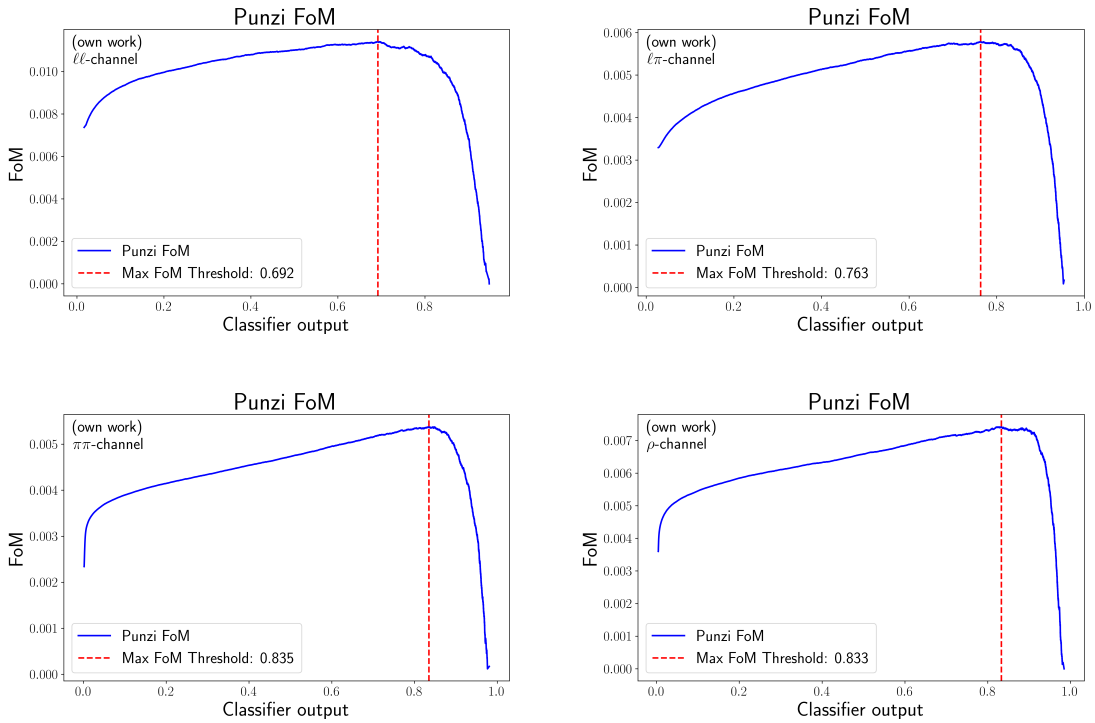


Figure 5.4.: The Punzi Figure of Merit at different classifier outputs. The classifier output corresponding to the highest FoM is chosen as the threshold for classification.

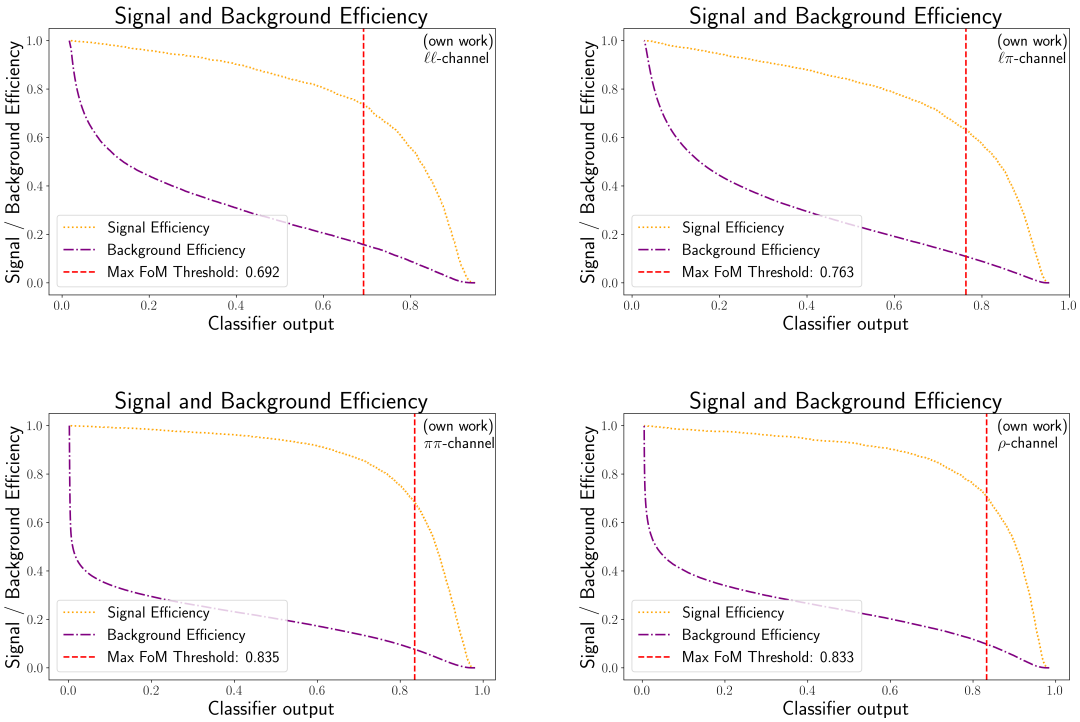


Figure 5.5.: Signal and background efficiency at different thresholds. The threshold, selected using the maximum FoM, does not lie in a region of rapid signal efficiency change, ensuring stable classification.

To obtain an estimation for the signal sensitivity, an "Asimov" dataset [33] is constructed, that matches the simulated dataset after the classifier is applied. The composition of the Asimov dataset is shown in Figure 5.6. To estimate the sensitivity of the analysis, a template maximum likelihood fit is performed, to find the signal strength μ_{UL} corresponding to the upper limit of the branching fraction $\mathcal{B}(\mathrm{B}^+ \to \mathrm{K}^{*+}\tau^+\tau^-)$ at 90% Confidence Level (see Figure 5.7). The upper limit of the signal strength μ_{UL} is a value that is multiplied by the number of simulated signal events N_{Signal}, to determine the number of signal events necessary to set the upper limit at 90% CL. A more detailed explanation of the signal extraction strategy is given in [1]. The upper limit of the branching fraction is computed using the signal strength μ_{UL}

$$\mathcal{B}(\mathrm{B}^+ \to \mathrm{K}^{*+}\tau^+\tau^-) < \frac{\mu_{\mathrm{UL}} \times N_{\mathrm{Signal}}}{\epsilon_{\mathrm{Signal}} \times 2 \times \mathcal{B}(\Upsilon(4S) \to \mathrm{B}^+\mathrm{B}^-) \times N_{\mathrm{B\bar{B}}}}, \tag{5.10}$$

where $\epsilon_{\mathrm{Signal}}$ denotes the combined signal efficiency (TPR) for all four channels and $N_{\mathrm{B\bar{B}}}$ corresponds to the number of $\mathrm{B\bar{B}}$ events in the entire dataset. The comparison of the estimated sensitivities using the different classifiers is shown in Table 5.3. A 4.86% increase in the sensitivity of the analysis is observed when substituting the FastBDT classifier with the ECT classifier.

Table 5.3.: Estimation of the upper limits on the branching ratio $\mathcal{B}(\mathrm{B}^+ \to \mathrm{K}^{*+}\tau^+\tau^-)$ using a template maximum likelihood fit of analyses applying different classifiers (FastBDT [1] and ECT (own work)). The upper limit on the branching ratio of the analysis using the ECT is 4.86% lower, indicating an improvement in sensitivity. The upper limits are computed by Lennard Damer.

Model	Upper limit on $\mathcal{B}(\mathrm{B}^+ \to \mathrm{K}^{*+}\tau^+\tau^-) \times 10^{-3}$ at 90 % CL
FastBDT	5.76
ECT	**5.48**

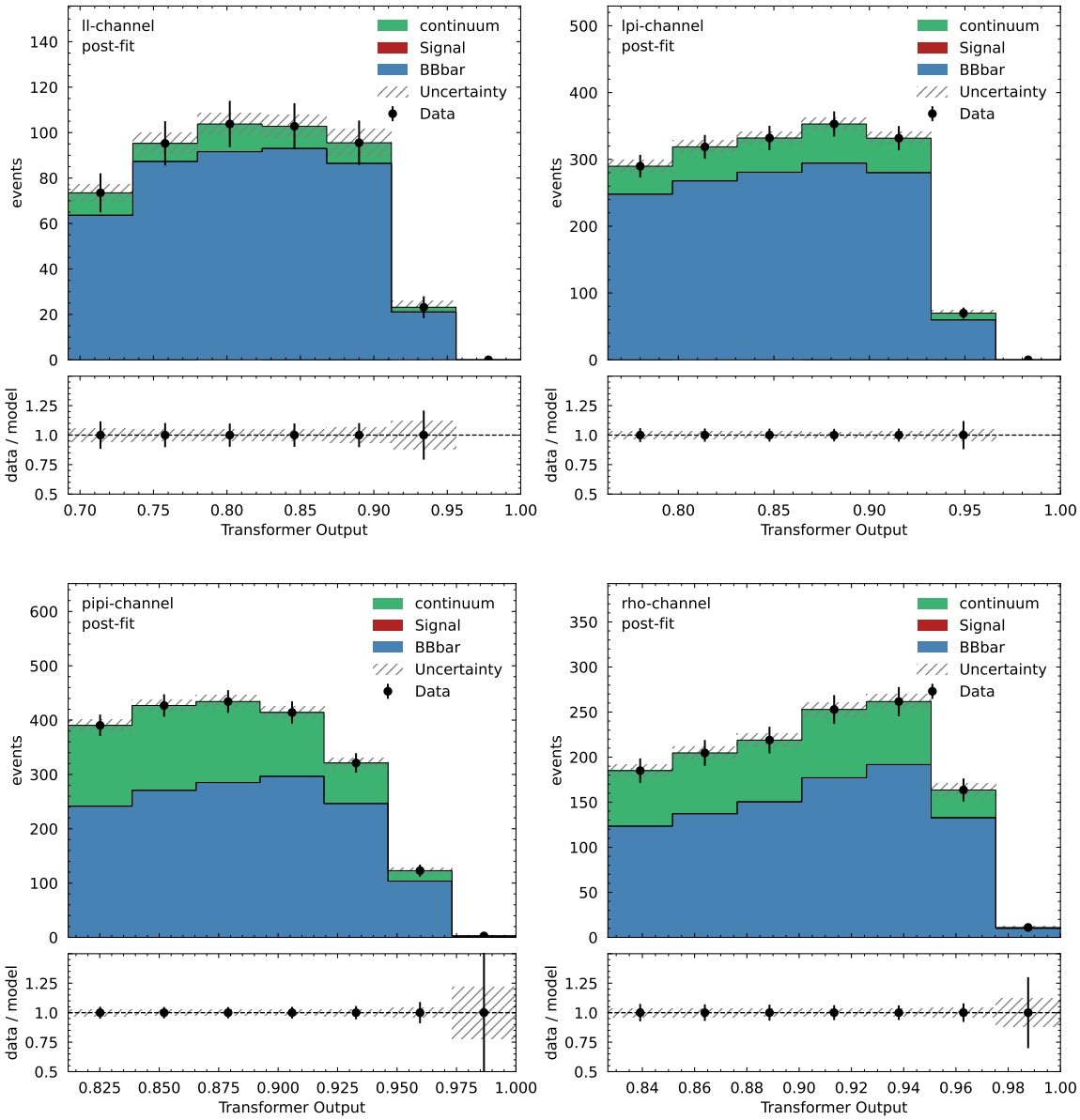


Figure 5.6.: Construction of the "Asimov" dataset from the MC data after applying the classifier cut found through the Punzi FoM. The dataset is used to estimate the sensitivity. The plots are created by Lennard Damer.

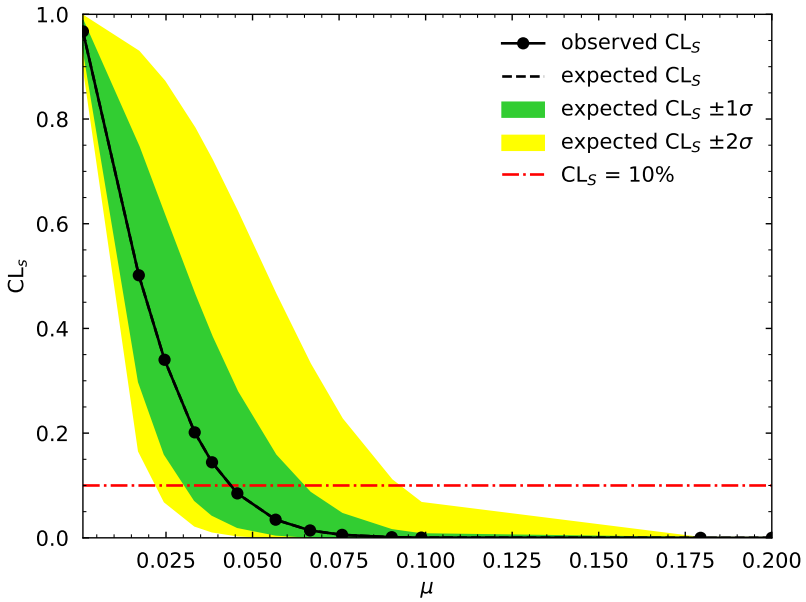


Figure 5.7.: Extraction of the signal strength μ that is used to find the sensitivity of the upper limit of the branching fraction at 90% CL. The upper limit of the signal strength μ_{UL} is a value that is multiplied by the number of simulated signal events N_{Signal}, to determine the number of signal events necessary to set the upper limit at 90% CL. The plot is created by Lennard Damer.

5.2.4. Feature Importance

The feature importance measures how much each feature contributes to the classification of the events. The FastBDT model offers two different feature importance algorithms, and one feature importance algorithm is adapted to Transformer-based models (see section 4.4). The algorithm for finding the feature importance of a Transformer-based model is applied to the ECT, and the results are compared to the inner feature importance algorithm of FastBDT in Figures A.7, A.8, A.9, and A.10. This comparison serves as a proof of concept for the feature importance algorithm for Transformer-based models.

The feature importance of the Transformer-based model closely resembles the feature importance of the FastBDT model, validating the feature importance algorithm. Differences may be explained either by the fundamental difference of the model or by the fundamental difference between the feature importance algorithms.

6. Conclusion and Outlook

At the Belle II experiment, the search for the $B^+ \to K^{*+}\tau^+\tau^-$ process, a decay heavily suppressed in the Standard Model, is of interest, as an enhanced branching ratio in this decay would hint at New Physics beyond the Standard Model. To support this analysis, Transformer-based models are tested for the classification task, and their performance is evaluated against the widely used FastBDT [5] classifier.

Several Transformer-based approaches are investigated, showing similar performance while following a simplified training and hyperparameter optimization process. One model, the Event Classifier Transformer [10], is selected to re-do the classification of the $B^+ \to K^{*+}\tau^+\tau^-$ analysis, that was initially performed using the FastBDT. The comparison of both classifiers reveals similar performance in terms of the AUC value, which confirms the feasibility of Transformer-based approaches as tools for analyses in High Energy Physics.

The Event Classifier Transformer is employed in the data analysis workflow of the $B^+ \to K^{*+}\tau^+\tau^-$ process, demonstrating that it is possible to integrate Transformer-based models into existing workflows. A 4.86% increase in sensitivity is observed when the FastBDT classifier is substituted with the ECT classifier (see Table 5.3). Additionally, a feature importance algorithm for the Transformer model is examined, producing predictions that are consistent with those of FastBDT, making the model's predictions explainable.

Open questions remain for future work and Transformers-based models should be further explored and optimized for application in High Energy Physics:

Evaluating other Models In this work, several different Transformer-based models are considered, and one is chosen for a detailed performance study. However, there are alternatives for the choice of the model architecture (see subsection 4.2.3). Using other Transformer-based models may further improve the performance of the classification task.

Evaluating other Loss Functions For training the models, the Binary Cross Entropy Loss is used in this work. There are other choices of a loss function, as suggested in [10], that may be used, to improve the performance.

Feature Importance The feature importance algorithm tested in this work yields results that are consistent with those of FastBDT. However, there are several different ways to extract the feature importance [34]. The application of the attention mechanism may enable further feature importance algorithms, that are based on the attention weights of the model [35, 36]. Improving the feature importance algorithm allows for a better selection of input features.

Evaluating Cases of Transformer Model Effectivity In the datasets evaluated in this work, the Transformer-based model demonstrates better performance in terms of AUC value compared to FastBDT when applied to datasets with many input features, while it falls behind on datasets with fewer input features (see Figure 5.2). However, the differences in performance are small and may be subject to other influences. Further investigation is required to identify the specific conditions under which the Transformer model outperforms FastBDT.

Evaluating difference in Event Selection between Transformer and FastBDT It should be investigated whether the FastBDT and the Transformer classifier select different events. If so, combining both models could enhance classification performance.

Making Transformer-based models accessible To make Transformer-based classifiers accessible for further analyses, the methods used in this work should be documented so that they are understandable and easy to use.

The results indicate that Transformer-based approaches are promising tools, complementing traditional Machine Learning methods such as FastBDT. With further investigation, these methods could potentially be applied to other analyses, expanding the toolkit available for performing classification tasks.

Bibliography

[1] Lennard Damer. "Search for $B^+ \to K^{*+}\tau^+\tau^-$ with hadronic tagging at the Belle II Experiment". MA thesis. Karlsruhe Institute of Technology (KIT), 2024.

[2] Y. Amhis et al. "Averages of b -hadron, c -hadron, and τ -lepton properties as of 2021". en. In: *Physical Review D* 107.5 (Mar. 2023), p. 052008. ISSN: 2470-0010, 2470-0029. DOI: 10.1103/PhysRevD.107.052008. URL: https://link.aps.org/doi/10.1103/PhysRevD.107.052008 (visited on 01/09/2025).

[3] Bernat Capdevila et al. "Searching for New Physics with $b \to s\tau^+\tau^-$ processes". In: *Physical Review Letters* 120.18 (May 2018), p. 181802. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.120.181802. URL: http://arxiv.org/abs/1712.01919 (visited on 08/08/2024).

[4] Latika Aggarwal et al. "Snowmass White Paper: Belle II physics reach and plans for the next decade and beyond". Sept. 2022. URL: http://arxiv.org/abs/2207.06307 (visited on 08/08/2024).

[5] Thomas Keck. "FastBDT: A speed-optimized and cache-friendly implementation of stochastic gradient-boosted decision trees for multivariate classification". Sept. 2016. DOI: 10.48550/arXiv.1609.06119. URL: http://arxiv.org/abs/1609.06119 (visited on 12/10/2024).

[6] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[7] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". May 2019. DOI: 10.48550/arXiv.1810.04805. URL: http://arxiv.org/abs/1810.04805 (visited on 01/29/2025).

[8] OpenAI. "GPT-4 Technical Report". Mar. 2024. DOI: 10.48550/arXiv.2303.08774. URL: http://arxiv.org/abs/2303.08774 (visited on 01/29/2025).

[9] Huilin Qu, Congqiao Li, and Sitian Qian. "Particle Transformer for Jet Tagging". Jan. 2024. URL: http://arxiv.org/abs/2202.03772 (visited on 08/08/2024).

[10] Jaebak Kim. "Training toward significance with the decorrelated event classifier transformer neural network". en. In: *Physical Review D* 109.9 (May 2024), p. 096035. ISSN: 2470-0010, 2470-0029. DOI: 10.1103/PhysRevD.109.096035. URL: https://link.aps.org/doi/10.1103/PhysRevD.109.096035 (visited on 01/14/2025).

[11] Mark Thomson. *Modern particle physics*. eng. Cambridge: Cambridge university press, 2013. ISBN: 978-1-107-03426-6.

[12] The BaBar Collaboration. "Search for $B^B \to K^+\tau^+\tau^-$ at the BaBar experiment". In: *Physical Review Letters* 118.3 (Jan. 2017), p. 031802. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.118.031802. URL: http://arxiv.org/abs/1605.09637 (visited on 12/29/2024).

[13] S. L. Glashow, J. Iliopoulos, and L. Maiani. "Weak Interactions with Lepton-Hadron Symmetry". en. In: *Physical Review D* 2.7 (Oct. 1970), pp. 1285–1292. ISSN: 0556-2821. DOI: 10.1103/PhysRevD.2.1285. URL: https://link.aps.org/doi/10.1103/PhysRevD.2.1285 (visited on 12/31/2024).

[14] JoAnne L. Hewett. "τ polarization asymmetry in $B \to X_S\tau^+\tau^-$". en. In: *Physical Review D* 53.9 (May 1996), pp. 4964–4969. ISSN: 0556-2821, 1089-4918. DOI: 10.1103/PhysRevD.53.4964. URL: https://link.aps.org/doi/10.1103/PhysRevD.53.4964 (visited on 12/31/2024).

[15] Marat Freytsis, Zoltan Ligeti, and Joshua T. Ruderman. "Flavor models for $\bar{B} \to D^{(*)}\tau\bar{\nu}$". In: *Physical Review D* 92.5 (Sept. 2015), p. 054018. ISSN: 1550-7998, 1550-2368. DOI: 10.1103/PhysRevD.92.054018. URL: http://arxiv.org/abs/1506.08896 (visited on 12/29/2024).

[16] S. Navas et al. "Review of Particle Physics". en. In: *Physical Review D* 110.3 (Aug. 2024), p. 030001. ISSN: 2470-0010, 2470-0029. DOI: 10.1103/PhysRevD.110.030001. URL: https://link.aps.org/doi/10.1103/PhysRevD.110.030001 (visited on 12/13/2024).

[17] Felix Gregor Kuno Metzner. "Preparation of a Measurement of $\mathcal{R}(D^{(*)})$ with Leptonic τ and Hadronic FEI Tag at the Belle Experiment". PhD Thesis. Karlsruher Institut für Technologie (KIT), 2022. DOI: 10.5445/IR/1000148812.

[18] T. V. Dong et al. “Search for the decay $B^0 \to K^{*0}\tau^+\tau^-$ at the Belle experiment”. en. In: *Physical Review D* 108.1 (July 2023), p. L011102. ISSN: 2470-0010, 2470-0029. DOI: 10.1103/PhysRevD.108.L011102. URL: https://link.aps.org/doi/10.1103/PhysRevD.108.L011102 (visited on 12/30/2024).

[19] Kazunori Akai, Kazuro Furukawa, and Haruyo Koiso. “SuperKEKB collider”. en. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 907 (Nov. 2018), pp. 188–199. ISSN: 01689002. DOI: 10.1016/j.nima.2018.08.017. URL: https://linkinghub.elsevier.com/retrieve/pii/S0168900218309616 (visited on 12/12/2024).

[20] Kenta Uno, Martin Bessner, and Yinghui Guan. “Quick review of 2024c run”. Jan. 2025. URL: https://indico.belle2.org/event/14053/contributions/86708/attachments/32108/47462/2025.01.06_Kuno.pdf.

[21] Moritz Bauer. “Measuring the Branching Fraction of $B \to \rho\ell\nu_\ell$ Decays with the Belle II Experiment”. PhD Thesis. Karlsruhe Institute of Technology (KIT), 2023.

[22] T. Abe et al. “Belle II Technical Design Report”. Nov. 2010. DOI: 10.48550/arXiv.1011.0352. URL: http://arxiv.org/abs/1011.0352 (visited on 12/16/2024).

[23] Xin Huang et al. “TabTransformer: Tabular Data Modeling Using Contextual Embeddings”. Dec. 2020. URL: http://arxiv.org/abs/2012.06678 (visited on 10/17/2024).

[24] Yury Gorishniy et al. “Revisiting Deep Learning Models for Tabular Data”. Oct. 2023. URL: http://arxiv.org/abs/2106.11959 (visited on 10/25/2024).

[25] Sercan O. Arik and Tomas Pfister. “TabNet: Attentive Interpretable Tabular Learning”. Dec. 2020. DOI: 10.48550/arXiv.1908.07442. URL: http://arxiv.org/abs/1908.07442 (visited on 01/06/2025).

[26] Hugo Touvron et al. “Going deeper with Image Transformers”. Apr. 2021. URL: http://arxiv.org/abs/2103.17239 (visited on 11/15/2024).

[27] Kaiming He et al. “Deep Residual Learning for Image Recognition”. Dec. 2015. DOI: 10.48550/arXiv.1512.03385. URL: http://arxiv.org/abs/1512.03385 (visited on 01/08/2025).

[28] Takuya Akiba et al. “Optuna: A Next-generation Hyperparameter Optimization Framework”. en. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Anchorage AK USA: ACM, July 2019, pp. 2623–2631. ISBN: 978-1-4503-6201-6. DOI: 10.1145/3292500.3330701. URL: https://dl.acm.org/doi/10.1145/3292500.3330701 (visited on 10/18/2024).

[29] Ian Goodfellow and Yoshua Bengio and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL: http://www.deeplearningbook.org.

[30] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[31] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". Jan. 2017. DOI: 10.48550/arXiv.1412.6980. URL: http://arxiv.org/abs/1412.6980 (visited on 01/15/2025).

[32] Giovanni Punzi. "Sensitivity of searches for new signals and its optimization". Dec. 2003. DOI: 10.48550/arXiv.physics/0308063. URL: http://arxiv.org/abs/physics/0308063 (visited on 11/26/2024).

[33] Glen Cowan et al. "Asymptotic formulae for likelihood-based tests of new physics". en. In: *The European Physical Journal C* 71.2 (Feb. 2011), p. 1554. ISSN: 1434-6044, 1434-6052. DOI: 10.1140/epjc/s10052-011-1554-0. URL: http://link.springer.com/10.1140/epjc/s10052-011-1554-0 (visited on 01/22/2025).

[34] Christoph Molnar. *Interpretable Machine Learning*. 2nd ed. 2022. URL: https://christophm.github.io/interpretable-ml-book.

[35] Tobias Leemann et al. "Attention Mechanisms Don't Learn Additive Models: Rethinking Feature Importance for Transformers". Jan. 2025. DOI: 10.48550/arXiv.2405.13536. URL: http://arxiv.org/abs/2405.13536 (visited on 01/27/2025).

[36] Samira Abnar and Willem Zuidema. "Quantifying Attention Flow in Transformers". en. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, 2020, pp. 4190–4197. DOI: 10.18653/v1/2020.acl-main.385. URL: https://www.aclweb.org/anthology/2020.acl-main.385 (visited on 01/27/2025).

[37] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". Dec. 2019. DOI: 10.48550/arXiv.1912.01703. URL: http://arxiv.org/abs/1912.01703 (visited on 01/14/2025).

[38] TensorFlow Developers. "TensorFlow". Oct. 2024. DOI: 10.5281/ZENODO.4724125. URL: https://zenodo.org/doi/10.5281/zenodo.4724125 (visited on 01/14/2025).

A. Appendix

A.1. Selection of a Model

In this section, the process of testing different models is described, to choose one model for a detailed comparison to the FastBDT. The training and hyperparameter optimization processes differ from the process in the analysis using FastBDT in [1] in that they are simpler and less computationally expensive. The hyperparameter optimization is performed using Optuna, as described in section 4.3. Instead of k-fold cross validation, a validation set is used, reducing the amount of data available for training and testing. As the validation score, the Binary Cross Entropy Loss of the validation set is utilized. A pruning algorithm (the `Median Pruner` from Optuna [28]) is applied, stopping unpromising trials at an early stage. Additionally, the constraint to avoid overfitting is not applied. The number of epochs is not specified as a hyperparameter, and instead, the early stopping algorithm is applied, stopping the training, if the validation score does not improve for a given number of epochs (patience), and saving the model weights of the best epoch.

The dataset outlined in subsection 4.1.1 is utilized to train all models, with each model being trained separately on all four channels using the input features specified in Table A.5. The dataset is split into the training set (70% of the data), the validation set (15% of the data), and the test set (15% of the data). The Adam optimizer [31] is used to train all four models. As a loss function, the Binary Cross Entropy Loss[1] [29] is used, where weights are applied to account for the imbalance between the size of the signal and the background set. The weights are computed as described in Equation 5.6

Some of the models tested are slightly adapted to make them suitable for the task, and the implementations are discussed below. Hyperparameter optimization is not performed on the TabNet model, because fundamental changes to the code would have been required.

[1] documentation: https://pytorch.org/docs/stable/generated/torch.nn.BCELoss.html (accessed on 15th January 2025)

TabTransformer [23] The model is implemented in the PyTorch [37] framework[2]. An adaptation to the model presented in [23] is made by integrating a linear embedding layer, instead of passing the data directly to the attention block. The hyperparameter search space is given Table A.1, and the resulting ROC curve is shown in Figure A.1.

FT-Transformer [24] The model is implemented in the TensorFlow [38] framework[3]. No adaptations to the model are made. The hyperparameter search space is given in Table A.2, and the resulting ROC curve is shown in Figure A.3.

TabNet [25] The model is implemented in the PyTorch [37] framework[4]. The default hyperparameter settings are used, and the resulting ROC curve is shown in Figure A.2.

Event Classifier Transformer [10] The model is implemented in the PyTorch [37] framework[5]. No adaptations to the model are made. The hyperparameter search space is given in Table A.3, and the resulting ROC curve is shown in Figure A.4.

The resulting test AUC are compared in Table 5.1. The resulting scores have to be considered with caution, because of the different hyperparameter optimization procedures and because no constraints to avoid overfitting are applied.

Table A.1.: Hyperparameter ranges used for the TabTransformer in section A.1. The optimization is performed over 100 iterations.

Hyperparameter	Range
activation function	{GeLu, ReLU}
embedding dimension	[3,12]
dropout	[0,0.6]
N layers	[2,5]
learning rate	$[10^{-6}, 10^{-3}]$
patience	[10,25]
batch size	64

[2] documentation: https://pytorch.org/docs/stable/generated/torch.nn.TransformerEncoder.html (accessed on 16th October 2024)

[3] implementation adapted from: https://github.com/aruberts/TabTransformerTF/tree/main (accessed on 29th October 2024)

[4] documentation: https://pypi.org/project/pytorch-tabnet/ (accessed on 29th October 2024)

[5] implementation adapted from: https://github.com/jaebak/EventClassifierTransformer (accessed on 06th November 2024)

Table A.2.: Hyperparameter ranges used for the FT-Transformer in section A.1. The optimization is performed over 50 iterations.

Hyperparameter	Range
attention dropout	[0,0.5]
FFN dropout	[0,0.5]
learning rate	$[10^{-5}, 10^{-2}]$
batch size	{32, 64, 128, 256, 512, 1024}
patience	20

Table A.3.: Hyperparameter ranges used for the ECT in section A.1. The optimization is performed over 100 iterations.

Hyperparameter	Range
dropout	[0,0.6]
linear factor	[1,12]
learning rate	$[10^{-6}, 10^{-3}]$
N heads	[2,8]
N nodes	[3,10] $\times$ N heads
patience	20
batch size	64

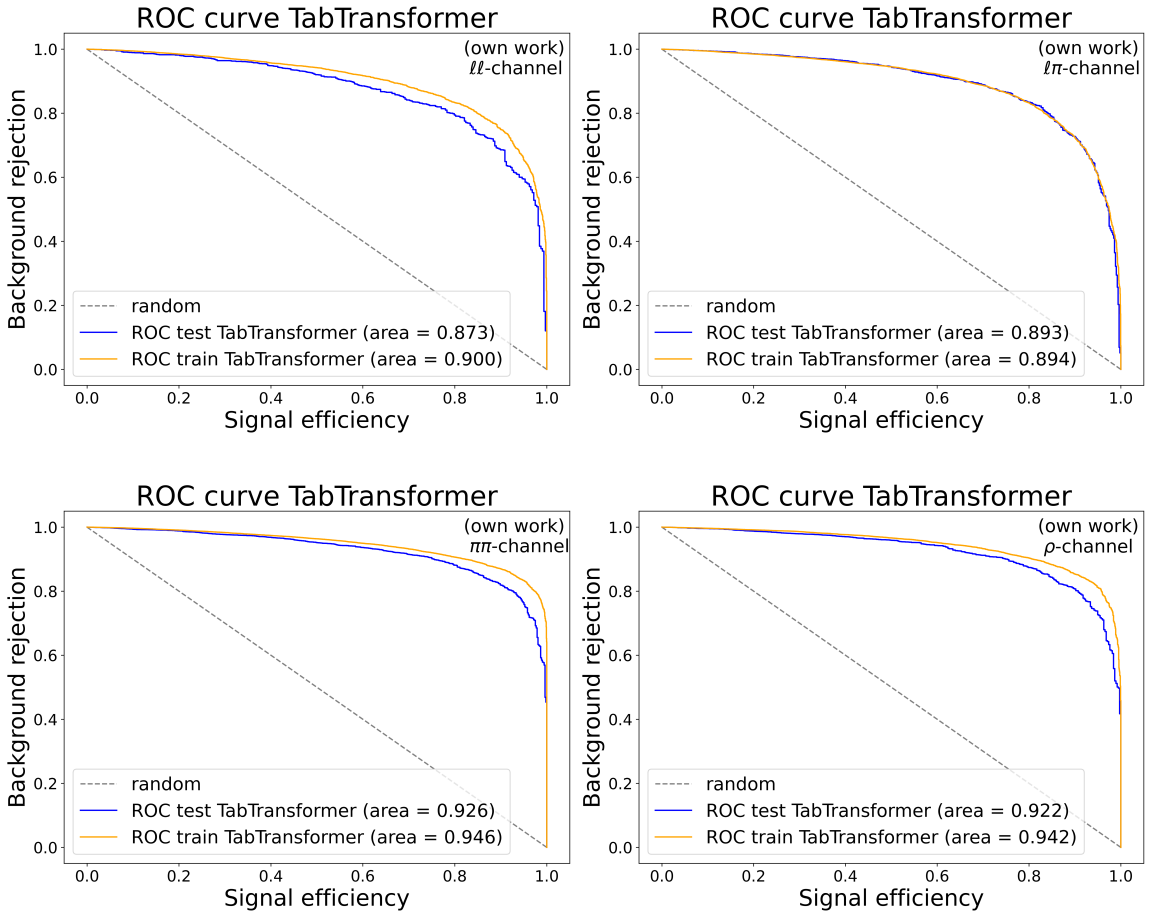


Figure A.1.: ROC curves of the TabTransformer on all four channels. The simplified process for training and selecting hyperparameters, as described in section A.1, is used.

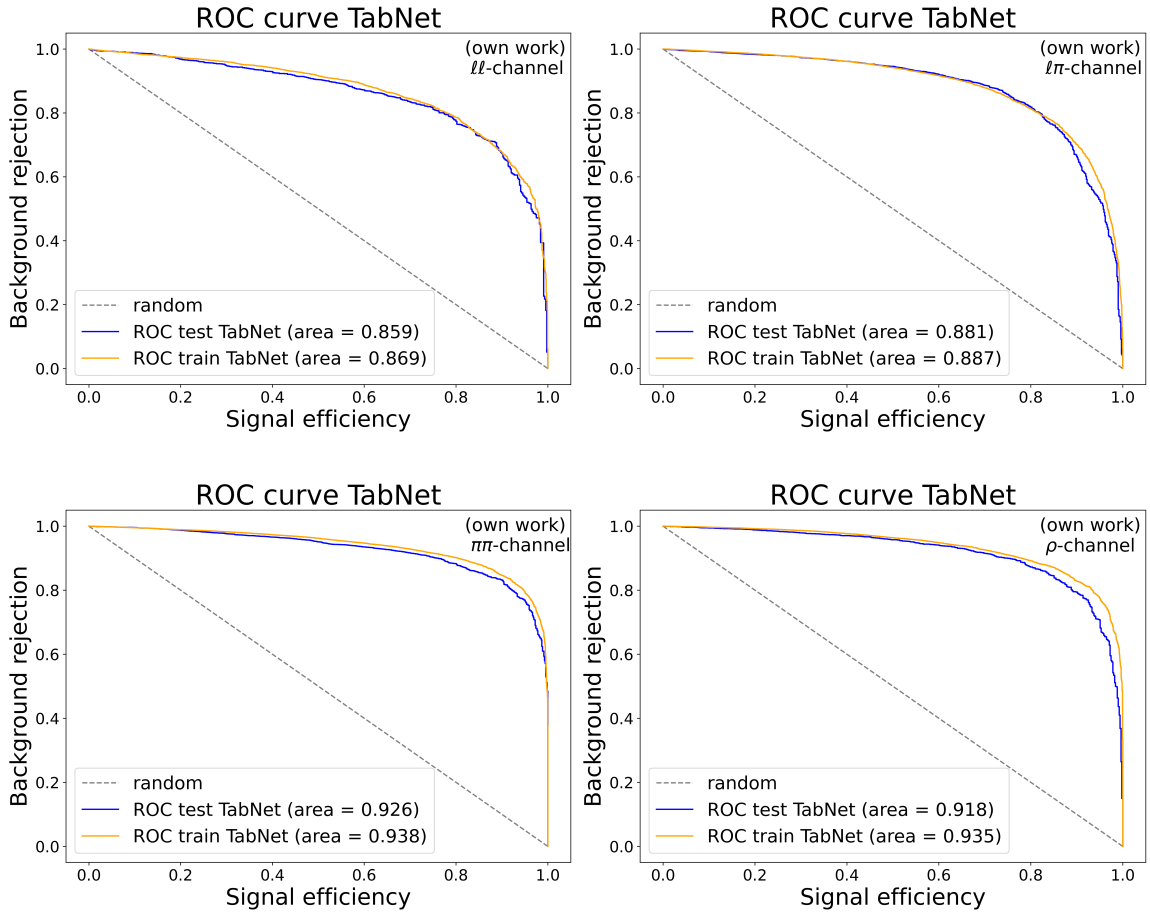


Figure A.2.: ROC curves of the TabNet on all four channels. The simplified process for training and selecting hyperparameters, as described in section A.1, is used.

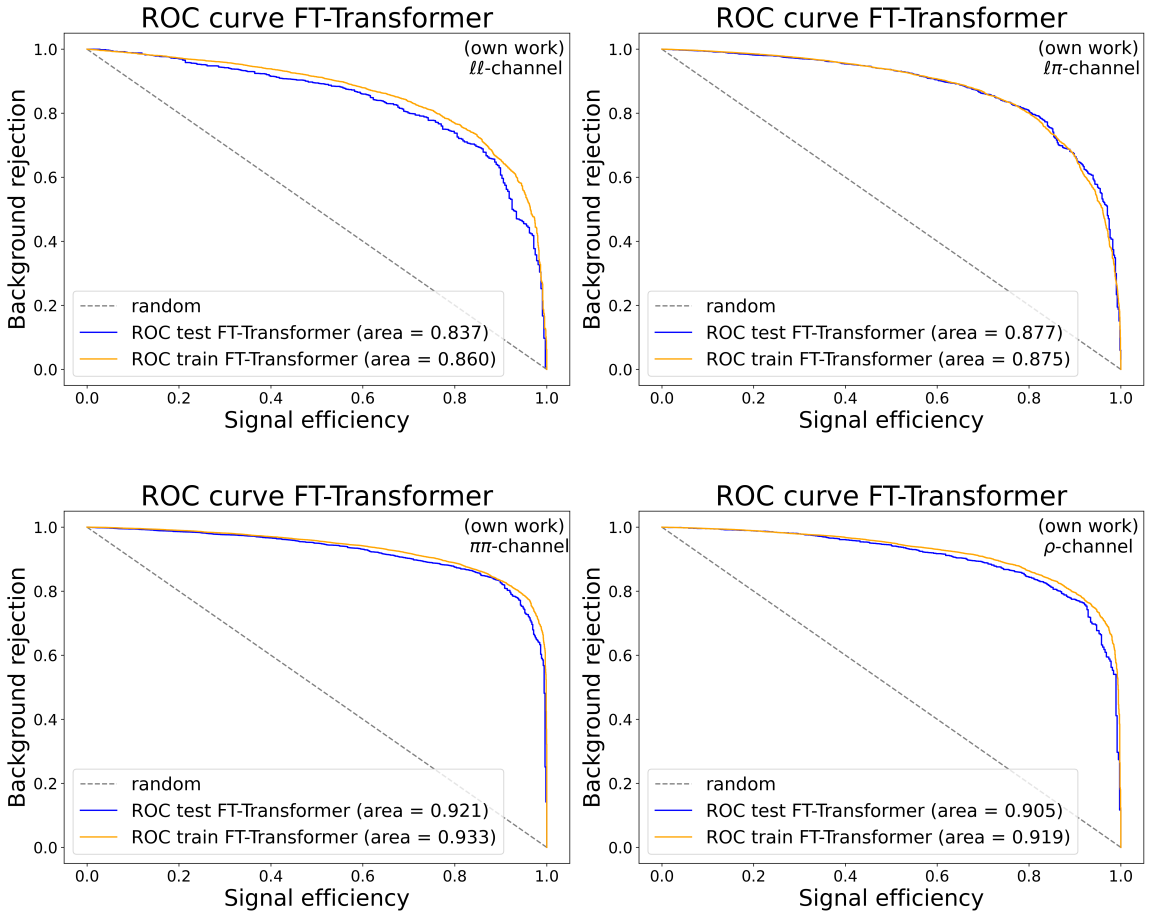


Figure A.3.: ROC curves of the FT-Transformer on all four channels. The simplified process for training and selecting hyperparameters, as described in section A.1, is used.

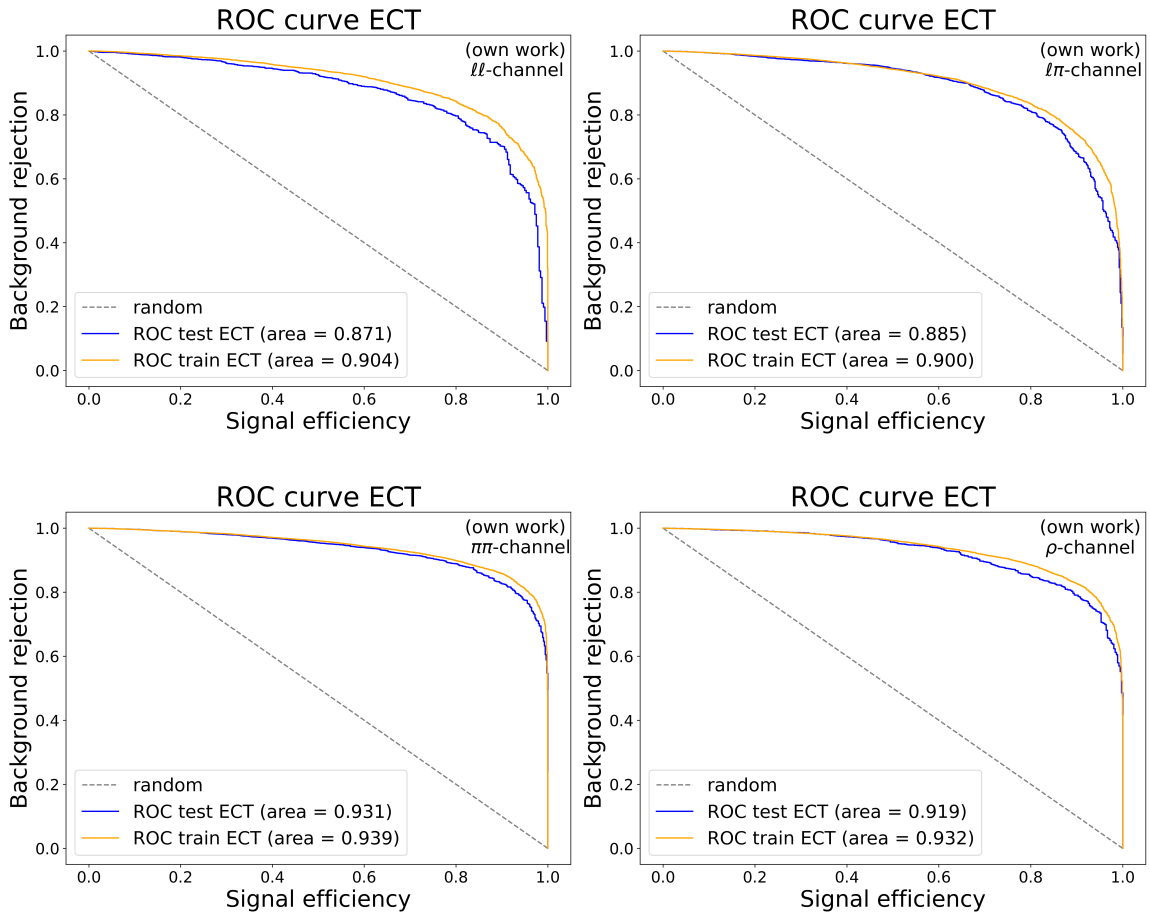


Figure A.4.: ROC curves of the ECT on all four channels. The simplified process for training and selecting hyperparameters, as described in section A.1, is used.

A.2. Evaluation of the Event Classifier Transformer

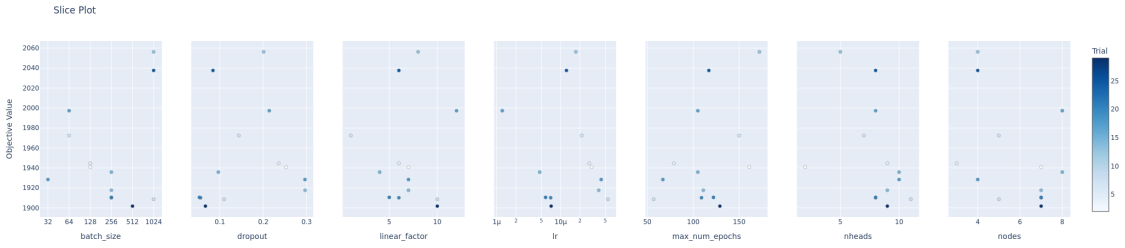
In this section, additional plots and tables related to subsection 5.2.2 are shown.

Table A.4.: Search space for the hyperparameter optimization. The parameters that describe the model are the dropout rate within the dropout layers, the number of heads of the MHA, and the linear factor and number of nodes that specify the dimensions of the linear layers. The parameters describing the optimization procedure are the learning rate of the classifier, the batch size, and the number of epochs. The optimization is computationally expensive due to k-fold cross validation, and the number of trials is set to N=30.

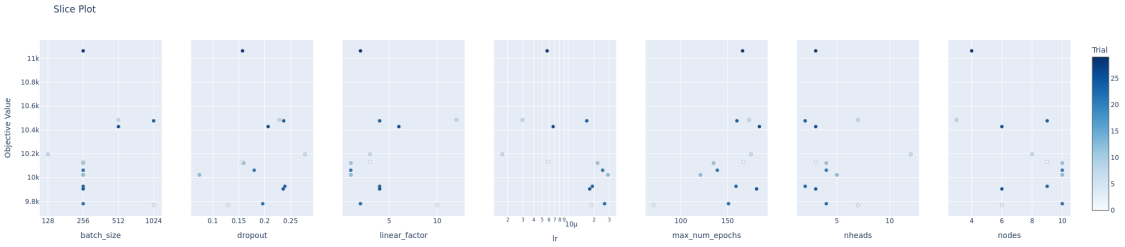
Hyperparameter	Range
dropout	[0,0.3]
N heads	[1,12]
linear factor	[1,12]
N nodes	[3,10] $\times$ N heads
learning rate	[1E-6, 1E-4]
batch size	{32, 64, 128, 256, 512, 1024}
N epochs	[50,200]

Table A.5.: List of input features used for the training of the FastBDT and the Transformer-based model. To allow for a fair comparison, the choice of input features is not changed from the analysis using the FastBDT. Adapted from [1].

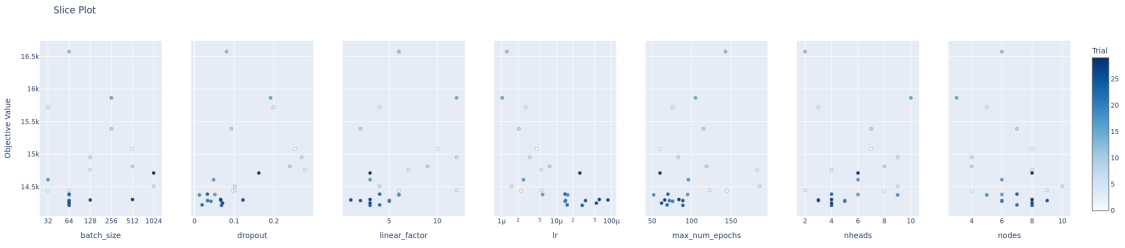
Training Variable	$\ell\ell$	$\ell\pi$	$\pi\pi$	ρ
p_{miss}^{CMS}	✓	✓	✓	✓
E_{miss}^{CMS}	✓	✓	✓	✓
E_t		✓	✓	✓
$M_{K^{*+}}$	✓		✓	✓
$M(K^{*+}; t_1)$	✓	✓	✓	✓
$M(K^{*+}; t_2)$	✓	✓	✓	✓
$\cos(Thrust_B; Thrust_{ROE})$	✓	✓	✓	✓
$\cos(Thrust_B; Thrust_z)$	✓	✓	✓	✓
$\mathrm{Thrust_B}$	✓	✓	✓	✓
$\mathrm{Thrust_{ROE}}$				✓
CLEO Cone 0		✓	✓	✓
CLEO Cone 1	✓	✓	✓	
CLEO Cone 2		✓	✓	✓
CLEO Cone 3		✓	✓	✓
CLEO Cone 4		✓	✓	
CLEO Cone 5				✓
CLEO Cone 6			✓	✓
CLEO Cone 7		✓	✓	
CLEO Cone 8		✓	✓	
H_{00}^{so}		✓		✓
H_{02}^{so}		✓	✓	
H_{03}^{so}	✓		✓	✓
H_{04}^{so}		✓	✓	✓
H_{12}^{so}		✓	✓	
H_{14}^{so}		✓	✓	✓
H_{20}^{so}		✓	✓	✓
H_{22}^{so}	✓	✓	✓	✓
H_{24}^{so}	✓		✓	✓
H_{01}^{oo}			✓	✓
H_{02}^{oo}		✓	✓	✓
H_{03}^{oo}			✓	✓
H_{04}^{oo}			✓	✓
Σ	**12**	**23**	**29**	**26**



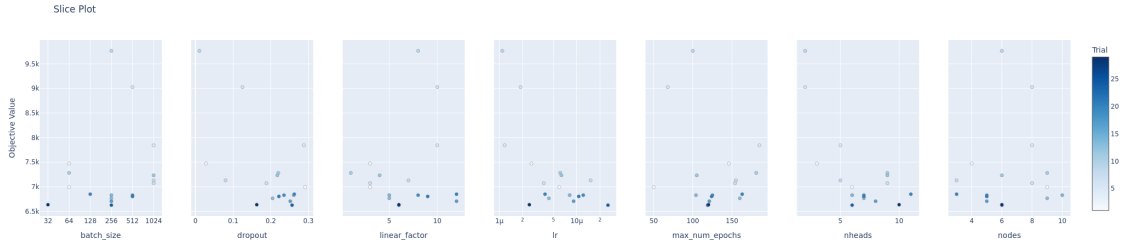
$\ell\ell$-channel



$\ell\pi$-channel



$\pi\pi$-channel



ρ-channel

Figure A.5.: History of the trials during hyperparameter optimization of the ECT model, where the process of training and hyperparameter optimization is replicated from the analysis using FastBDT. The objective value, that is minimized, is the mean Cross Entropy Loss of the validation sets during k-fold cross validation. The choice of the hyperparameter search space is the same for all channels (see Table A.4). Trials, that are excluded because of the constraints on overfitting and on the difference between folds, are not shown in this plot.

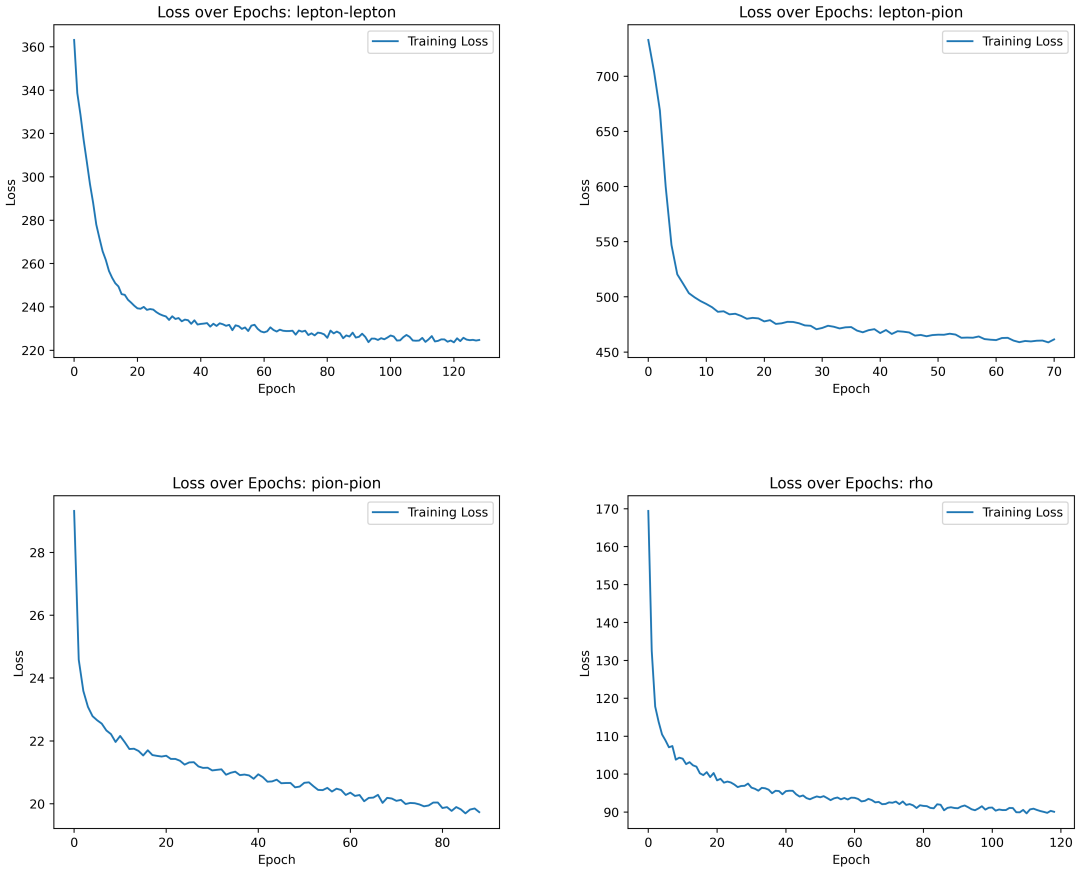


Figure A.6.: Binary Cross Entropy Loss of the training set during the training process of the ECT, where the process of training and hyperparameter optimization is replicated from the analysis using FastBDT.

A.3. Feature Importance

In this section, the comparison of the feature importance of FastBDT from [1] and the ECT. The relative scale of both feature importance algorithms differs due to the fundamental difference between approaches. The feature importance of the ECT closely resembles the feature importance of the FastBDT, validating the approach. The code creation for this comparison is aided by the AI programming assistant GitHub Copilot.

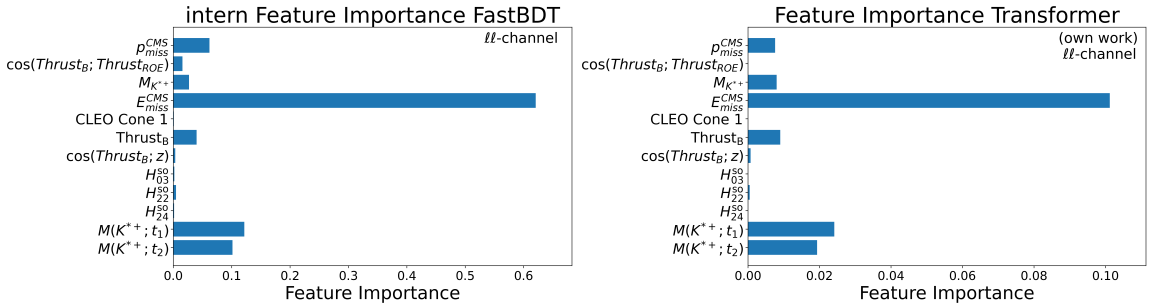


Figure A.7.: Feature importance of the FastBDT (left) and the ECT (right) of the $\ell\ell$-channel.

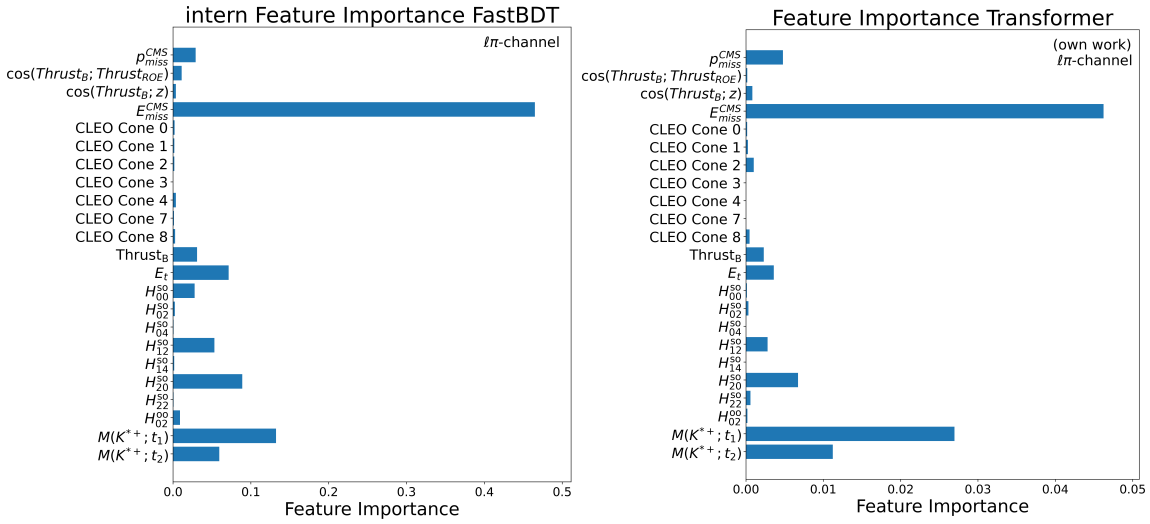


Figure A.8.: Feature importance of the FastBDT (left) and the ECT (right) of the $\ell\pi$-channel.

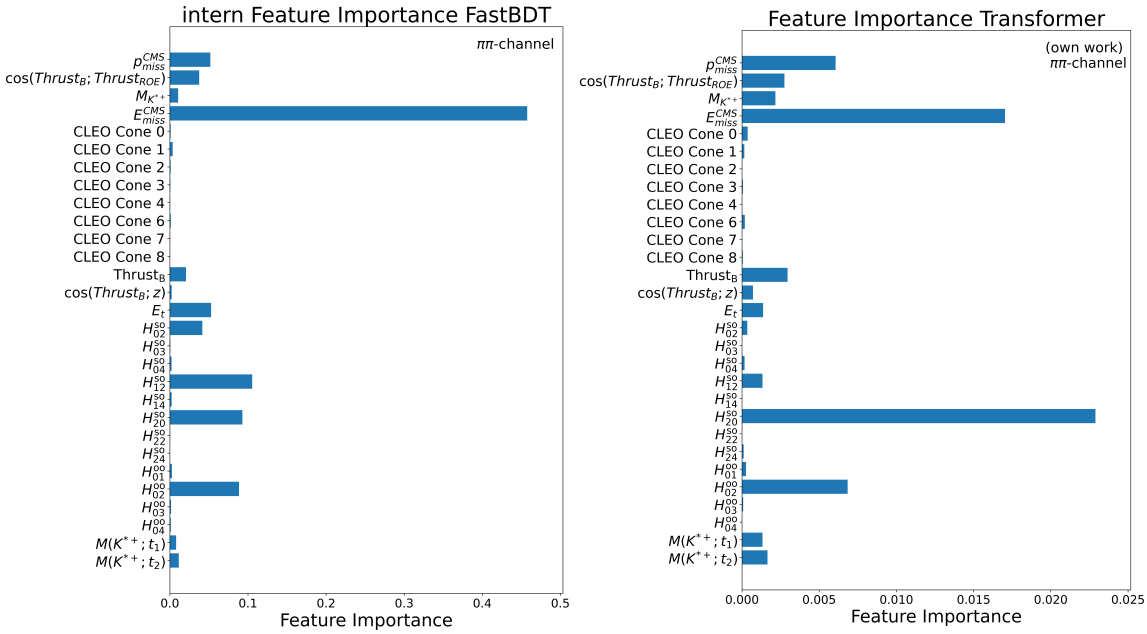


Figure A.9.: Feature importance of the FastBDT (left) and the ECT (right) of the $\pi\pi$-channel.

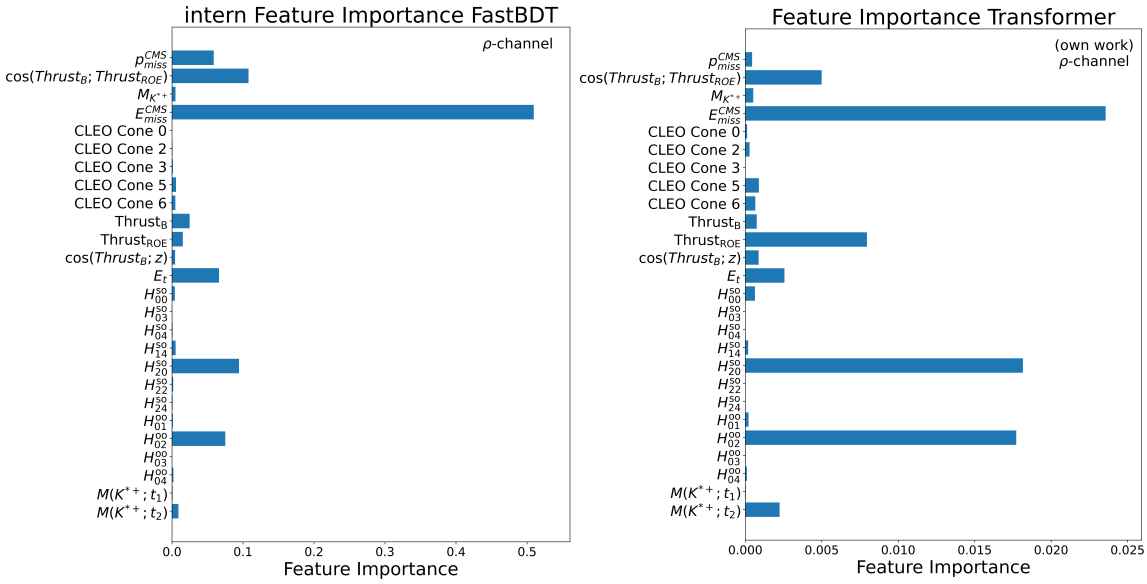


Figure A.10.: Feature importance of the FastBDT (left) and the ECT (right) of the ρ-channel.

Acknowledgements

I am grateful to Prof. Dr. Torben Ferber, Dr. Pablo Goldenzweig, and Lennard Damer for the good cooperation, and helpful comments on my work. I also want to thank Alexandra Mohrs, Max Burkardt, and Christian Bertsch for proofreading this thesis.

List of Figures

List of Tables

Acronyms

ARICH Aerogel RICH Detector. 11

AUC Area under the ROC Curve. 20, 22– 24, 26, 28, 35, 36, 42, 55, 56

BDT Boosted Decision Tree. 1, 12, 16, 17

CDC Central Drift Chamber. 10

CKM Cabibbo-Kobayashi-Maskawa. 3

CL Confidence Level. 6, 7, 13, 31, 33

COM center-of-mass. 8

ECL Electromagnetic Calorimeter. 9, 11

ECT Event Classifier Transformer. v, vi, 2, 17, 18, 24– 28, 31, 34, 35, 43, 47, 52, 53, 55, 56

FCNC flavour changing neutral current. 1, 3, 4

FFN feedforward neural network. 18

FoM Figure of Merit. 27, 29, 30, 32

FPR false positive rate. 21, 22, 26

GIM Glashow–Iliopoulos–Maiani. 3

GPT Generative Pre-trained Transformer. 2

IP interaction point. 8, 9

KLM K_L^0 / Muon Detector. 9– 11

LFU Lepton Flavour Universality. v, 1, 3, 4

MC Monte Carlo. 12, 32

MHA Multi-Head Attention. 15, 16, 18, 48

ML Machine Learning. 1, 2, 19, 36

MLP Multi Level Perceptron. 14

NP New Physics. 1, 3, 4, 7, 35

ParT Particle Transformer. 2, 17, 18

PXD Pixel Detector. 9, 10

ROC Receiver Operating Characteristic. v, 21, 22, 24, 26, 27, 42, 44– 47, 55

SGBDT stochastic Gradient Boosted Decision Tree. 13

SM Standard Model. 1, 3, 4, 7, 8, 35

SVD Silicone Vertex Detector. 10

TOP Time-of-Propagation Counter. 10, 11

TPR true positive rate. 21, 26, 31