Graph-based Full Event Interpretation: a graph neural network for event reconstruction in Belle II

M. AbuMusabh, J. Cerasoli, G. Dujany, C. Santos

March 13, 2025

Abstract

In this work we present the Graph-based Full Event Interpretation (GraFEI), a machine learning model based on graph neural networks to inclusively reconstruct events in the Belle II experiment. Belle II is well suited to perform measurements of B meson decays involving invisible particles (e.g. neutrinos) in the final state. The kinematical properties of such particles can be deduced from the energy-momentum imbalance obtained after reconstructing the companion B meson produced in the event. This task is performed by reconstructing it either from all the particles in an event but the signal tracks, or using the Full Event Interpretation, an algorithm based on Boosted Decision Trees and limited to specific, hard-coded decay processes. A recent example involving the use of the aforementioned techniques is the search for the $B^+ \rightarrow K^+ \nu \bar{\nu}$ decay, that provided an evidence for this process at about 3 standard deviations. The GraFEI model is trained to predict the structure of the decay chain by exploiting the information from the detected final state particles only, without making use of any prior assumptions about the underlying event. By retaining only signal-like decay topologies, the model considerably reduces the amount of background while keeping a relatively high signal efficiency. The performances of the model when applied to the search for $B^+ \rightarrow K^+ \nu \bar{\nu}$ are presented.

1 Introduction

One of the main tasks in any particle physics analysis is the reconstruction of decays. Since decays are often represented as decay trees and with the recent advances in machine learning, Graph Neural Networks (GNNs) have become an interesting tool for studying these decays. Indeed, GNNs are a particular class of neural networks acting on *graphs* $\mathcal{G}$, entities composed of $N \in \mathbb{N}^*$ nodes $V = \{v_i\}_{i \in [\![1,N]\!]}$, connected by edges $E = \{e_{v_i v_j} \equiv e_{ij}\}_{(i,j) \in [\![1,N]\!]^2, i \neq j}$. In this perspective, we developed the Graph-based Full Event Interpretation (GRAFEI), a machine learning model based on GNNs to inclusively reconstruct events at the Belle II experiment [1].

1

The Belle II experiment analyses e^+e^- collisions at the $\Upsilon(4S)$ resonance, producing $B\bar{B}$ pairs. Since the B mesons are produced in pairs, the kinematical properties of decays involving invisible particles (*e.g.* neutrinos) can be deduced from the energy-momentum imbalance obtained after reconstructing the companion B meson produced in the event. This task is performed by reconstructing it either from all the particles in an event but the signal tracks, called inclusive reconstruction, leading to a high efficiency but a low signal purity, or using the FULL EVENT INTERPRETATION (FEI) [2], an algorithm based on boosted decision trees and limited to specific, hard-coded decay processes, giving a result with a high signal purity but a low efficiency. A recent example involving the use of the aforementioned techniques is the search for the $B^+ \to K^+\nu\bar{\nu}$ decay, that provided an evidence for this process at about 3 standard deviations [3].

The goal of the GRAFEI model is to reconstruct the decay chain of the companion B meson produced in the event by exploiting the information from the detected final state particles only, without making use of any prior assumptions about the underlying event, as done in the inclusive reconstruction strategy. By retaining only signal-like decay topologies, similarly to the FEI, the model considerably reduces the amount of background while keeping a relatively high signal efficiency. The performances of the model when applied to the search for $B^+ \to K^+\nu\bar{\nu}$ are presented in this work as a proof of concept.

2 The graFEI model

2.1 Problem statement

Decay trees can be described by rooted directed acyclic tree graphs, represented algebraically by matrices like the adjacency matrix, a binary matrix which rows and columns are the nodes of the graph and its elements are either 0 or 1, indicating if the two nodes are connected or not. This approach is not suitable for reconstructing decay trees, because we would have to make assumptions on the intermediate particles. The GRAFEI reconstructs events inclusively using the information on the final state particles alone, without any prior assumption about the structure of the decay, a technique shown in [4–6] thanks to the *Lowest Common Ancestor (LCA)* matrix. Each entry of this matrix corresponds to a pair of final state particles, and its elements are the lowest ancestors common to pairs of particles. To avoid the use of a unique identifier for each ancestor, a system of classes is used: 6 for $\Upsilon(4S)$ mesons, 5 for B, 4 for D^*, 3 for D, 2 for K_s^0, 1 for π^0 or J/ψ and 0 for particles not belonging to the decay tree. This new representation of the LCA is called *LCAS* matrix, where the S stands for "stage". In addition to the LCAS matrix, the mass hypotheses of the final state particles can also be predicted. To do so, a second system of classes is defined: 1 for e, 2 for μ, 3 for π, 4 for K, 5 for p, 6 for γ and 0 for everything else.

Inside graphs information can be stored in nodes, edges, or the graph as a whole. Thus $\mathcal{G}$ is characterized by three sets of features:

- **Node features** The node features encode information about the fi-

nal state particles. The model uses the following node features: particle charge, p_T, p_z, dr, dz, electronID, protonID, muonID, kaonID, pionID, as well as some clusters' information. In the case where a value is not available (e.g. cluster variables computed for charged particles) the variable is set to 0.

- **Edge features** The edge features are quantities that depend on pairs of particles. The model uses the following edge features: the cosine of the azimutal angle between particles' momenta $(\cos\theta)$ and the distance of closest approach between two tracks (`DOCA`). In order to compute the `DOCA` for photons, the particle is assumed to pass through the origin of the coordinate system.

- **Global features** The global features can be information about the overall graph (e.g. the number of final state particles, the total momentum, ...). The model currently does not use any input global feature.

In order to reconstruct the LCAS matrix, the input of the network is a *fully connected* graph, where each node (i.e. each final state particle) is connected with all the others. The output of the model is a graph with the same structure as the input but updated values of its features.

2.2 Model architecture

The GRAFEI model consists of a series of *graph network blocks* [7]: the first block takes the input graph and updates its features. The output graph is then passed through a series of intermediate blocks (whose number is a hyperparameter of the model and is equal to 1 in the default configuration), which further update the features. Finally, the last block outputs a graph used to predict the quantities of interest. A *skip-connection* [8] is present around each intermediate graph network block. A schematic view of the model is shown in Figure 1.
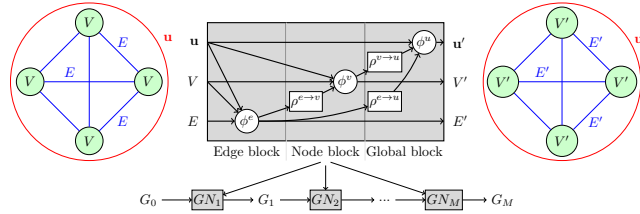

Figure 1: Schematic view of the GRAFEI model: the input graph G_0 if passed through a series of graph network blocks (GN_i) each in turn composed of three sub-blocks. The output graph G_M has the same structure of the input one but updated features, and is used to make predictions on the quantities of interest. Adapted from [7].

Each graph-network block is in turn composed of three sub-blocks: the edge, node, and global sub-blocks:

- **Edge sub-block** The edge sub-block is used to update the edge features. The core of the block is a MultiLayer Perceptron (MLP) ϕ^e, called the *update function* of the sub-block. For each edge e, the input vector of ϕ^e contains the edge features, the node features of the two nodes linked by the edge, and the global features. The output of the edge sub-block is also used as input to the node and global sub-blocks, via the use of *aggregation functions* $\rho^{e \to v}$ and $\rho^{e \to u}$. For each output feature, these functions perform the arithmetic average for edges connected to the same node and for all the edges in the same graph, respectively. The output of the edge sub-block in the last graph network block is used to predict the LCAS matrix of the graph. This is done by evaluating probabilities for each edge to belong to the 6 classes defined in the LCAS matrix.

- **Node sub-block** The node sub-block is used to update the node features. The core of the block is a MLP ϕ^v. For each node v, the input vector of ϕ^v contains the node features, the output of $\rho^{e \to v}$, and the global features. The output of the node sub-block is also used as input to the global sub-block, via the use of the aggregation function $\rho^{v \to u}$, which, for each output feature, performs the arithmetic mean for all the nodes in the same graph. The output of the node sub-block in the last graph network block is used to predict the mass hypotheses of the final state particles. This is done by evaluating probabilities for each node to belong to the 7 classes defined for the mass hypothesis.

- **Global sub-block** The global sub-block is not used to predict physical quantities for the time being, nonetheless it is used to propagate the information throughout the network. The core of the block is a MLP ϕ^u. The input vector of ϕ^u contains the global features and the output of $\rho^{e \to u}$ and $\rho^{v \to u}$.

The number of hidden layers of each MLP ϕ is a hyperparameter of the model. In the default configuration it is equal to 1 and is the same for all the update functions.

2.3   Training and evaluation

The activation function used in the MLP is the Exponential Linear Unit (ELU) [9], defined as, $\forall x \in \mathbb{R}$:

$$\text{ELU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ e^x - 1 & \text{if } x < 0 \,. \end{cases}$$

During the training phase, *dropout* [10] is used in order to reduce overtraining. It consists in randomly setting to 0 the output of some neurons. The probability is an hyperparameter of the model which is chosen to be 0.3 in the default setting. Dropout is applied to the first and hidden layers of each MLP, but not to the output layer.

In order to accelerate the training of the network, *batch normalization* [11] is applied to the output layer of each MLP. It consists in scaling

4

the values of neurons $x \in \mathbb{R}^n, n \in \mathbb{N}^*$ (before the activation function) according to the mean $\mu_\mathcal{B}$ and standard deviation $\sigma_\mathcal{B}$ for a small batch $\mathcal{B}$:

$$\hat{x}_\mathcal{B}^{(k)} = \frac{x_\mathcal{B}^{(k)} - \mu_\mathcal{B}}{\sqrt{\sigma_\mathcal{B}^2 + \epsilon}}$$

where ϵ is a small constant for numerical stability and $k \in [\![1, \mathrm{Card}\,(\mathcal{B})]\!]$ is the index of the neuron. The scaled values are then used as input to a linear layer, whose parameters (γ, β) are learned during training:

$$y_\mathcal{B}^{(k)} = \gamma^{(k)}\hat{x}_\mathcal{B}^{(k)} + \beta^{(k)}$$

Batch normalization is not applied in the last graph network block.

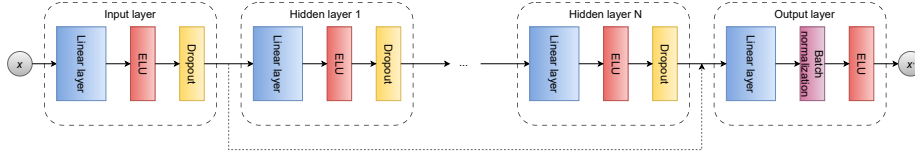The structure of a MLP ϕ is shown in Figure 2.

Figure 2: Schematic view of the structure of a MLP ϕ in the model. Batch normalization is not applied in the last graph network block. The dashed arrow indicates a skip-connection, which is present only in geometries with more than one hidden layer.

The model is trained using the Adam optimizer [12] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

The loss function used during the training phase has the following form:

$$\mathcal{L} = \mathcal{L}^{\mathrm{LCAS}} + \alpha \cdot \mathcal{L}^{\mathrm{Mass}}, \tag{1}$$

where $\mathcal{L}^{\mathrm{LCAS\,(Mass)}}$ is a cross-entropy loss [13] quantifying the difference between the model's output and the training data for the LCAS (mass hypotheses) predictions, while α is a parameter controlling the relative importance of the two terms, and is set to 1 in the default configuration. Various multi-objective optimization methods were tried, but this arbitrary choice lead to the best results. The cross-entropy loss over a batch is defined as:

$$\mathcal{L} = -\frac{1}{N}\sum_{n=1}^{N} \log \frac{\exp\left(x_{n,y_n}\right)}{\sum_{c=1}^{C}\exp\left(x_{n,c}\right)}, \tag{2}$$

where $N \in \mathbb{N}^*$ is the number of data, the sum over n spans the labeled data in the batch, the sum over c spans the classes used in the classification problem (the 6 ancestor labels for the LCAS and the 7 mass hypotheses for the final state particles), $x_{n,c}$ is the model's output for the n-th data and the c-th class while x_{n,y_n} is the model's output corresponding to the true class of the n-th data example. Such a function is close to 0 when the output corresponding to the true class x_{n,y_n} has a higher value than the output for the other classes, and is large otherwise.

5

In addition to the cross-entropy loss, two other metrics are used to evaluate the performances of the model. The **perfectLCA** metric is defined as the fraction of events in the dataset with a perfectly reconstructed LCAS matrix. The **perfectMasses** metric is defined as the fraction of events in the dataset where all the final state particles have the correct mass hypothesis assigned. The **perfectEvent** metric combines the two: it evaluates the fraction of events with a perfectly reconstructed LCAS matrix and where all the final state particles have the correct mass hypothesis assigned.

Moreover, an additional requirements called **validTree** is defined. Despite not being used during the training phase, it is useful to reject badly reconstructed events when applying the model on data. In general, an arbitrary matrix does not describe a valid LCAS tree representation. Consider exchanging a π^- and K^+ in Figure 3. This operation corresponds to altering *two* entries in the LCAS matrix. If only one is changed, the obtained matrix does not describe a coherent tree structure.
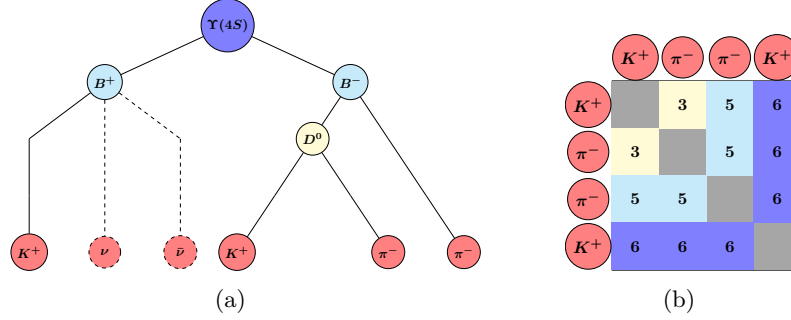

Figure 3: Schematic representation of a decay tree and its LCAS matrix representation.

In order to perform a selection on data, a numerical quantity derived from the GRAFEI is needed. This quantity should be interpreted as a likelihood for a given LCA matrix to describe a B decay, or a "B probability". Three quantities were investigated:

- $\texttt{BMean} \equiv -\frac{1}{I} \sum_{i=1}^{I} \log \frac{\exp(x_{i,y_i})}{\sum_{c=1}^{C} \exp(x_{i,c})}$;

- $\texttt{BProd} \equiv -\prod_{i=1}^{I} \log \frac{\exp(x_{i,y_i})}{\sum_{c=1}^{C} \exp(x_{i,c})}$;

- $\texttt{BGeom} \equiv \sqrt[I]{\texttt{BProd}}$;

where the definitions are analogous to the ones in Eq. 2, except for i which spans over the I elements of the LCA matrix below the diagonal. After studying the performances of the three quantities, the BGeom was chosen as the most effective, as one can see from figure 4.
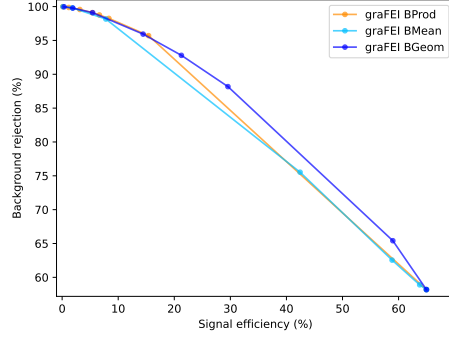
Figure 4: Signal efficiency VS background rejection for the three definitions of graFEI B probability.

3   Proof of concept: the $B^+ \to K^+ \nu\bar{\nu}$ decay

3.1   State of the art

This tool was applied to the search for the $B^+ \to K^+ \nu\bar{\nu}$ decay at Belle II. As previously mentionned, two techniques, inclusive reconstruction and FEI, were used for this search, leading to two distinct analyses. In the following, *ITA* will correspond to the inclusive reconstruction technique and *HTA* to the FEI technique. Those analyses lead, combined, to an evidence for a 2.7σ deviation from the standard model prediction, as shown by figures 5a and 5b.
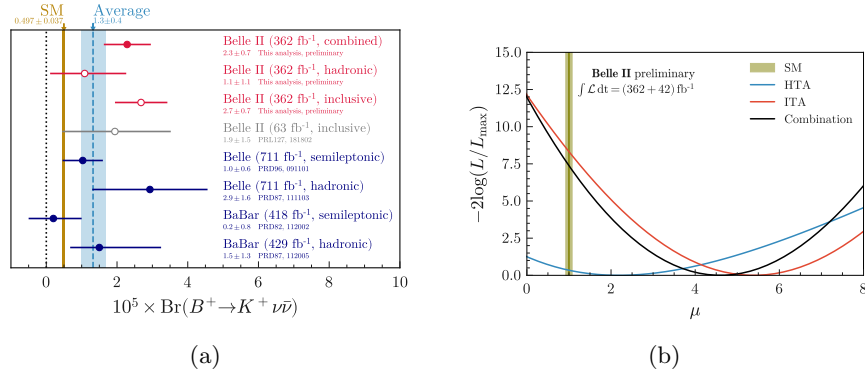

(a)

(b)

Figure 5: State of the art of the $B^+ \to K^+ \nu\bar{\nu}$ search. a shows the branching ratio of the decay for the inclusive reconstruction and FEI analyses done at Belle II compared to the previous results. b shows the negative log-likelihood of the FEI and inclusive analyses with respect to the *signal strength* μ, the ratio between the experimental and the standard model prediction of the branching ratio. [3]

3.2 Analysis strategy

In order to compare the performances of the GRAFEI model with the FEI and inclusive analyses, we need to define a similare analysis strategy, named *GTA*:

- **Reconstruction and preselection**: We select events where the LCAS contains the correct topology. We then ask that the event have a `BGeom` > 0.2, before applying a cut on the kaon likelihood of the signal kaon (> 0.9) and a cut on the cosine of the angle between the thrust axis of the signal B meson and the thrust axis of the rest of the event (< 0.9).

- **Training of a binary classifier**: The remaining background is further rejected using a BDT implemented with XGBoost. The BDT is trained using 10% of the dataset, further divided into an 80% for training and 20% for testing, and applied to the remaining 90% of the simulation (the loss in efficiency is corrected accordingly in the final result). The BDT is trained using 21 input variables.

- **Signal region cut**: The BDT distribution is subsequently flattened on signal events. A signal-enhanced region is defined by applying a cut on BDT > 0.8. The signal efficiency for this cut is about 18%.

3.3 Comparison

We start by defining the signal purity $\mathcal{P}^{\mathrm{sig}}$ as the ration between the number of signal events over the number of signal and background events after the whole strategy. The total efficiency of the technique, ε, will simply be the product of the efficiency at each step as described in the previous subsection. The results are summarized in the table 1.

	$\varepsilon[\%]$	$\mathcal{P}^{\mathrm{sig}}[\%]$
HTA	0.4	3.5
ITA	8	0.8
GTA	**2.7**	**1.3**

Table 1: Comparison of the signal purity and efficiency for the FEI (HTA), inclusive (ITA) and GRAFEI (GTA) analyses.

We also did the comparison of the likelihood for each analysis, similarly to figure 5b, but without considering any systematic uncertainty and normalizing the signal strength to 1 to have a purely statistical comparison of the methods, as one can see in figure 6.

4 Future of the graFEI

With this work, we demonstrated that the GRAFEI model is a powerful tool for the inclusive reconstruction of B meson decays at Belle II. The model is capable of reconstructing the $B^+ \to K^+ \nu \bar{\nu}$ decay without any prior assumption on the decay topology, leading to a higher efficiency than

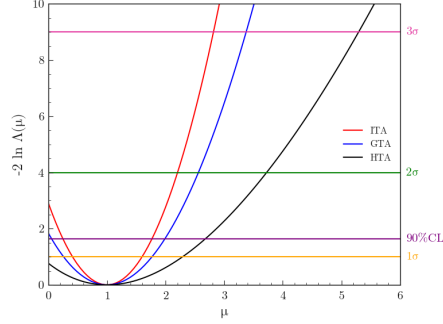Figure 6: Negative log-likelihood of the FEI, inclusive and GRAFEI analyses with respect to the signal strength μ.

the previous exclusive algorithm. Moreover, the knowledge of the decay topology, as infered by the GRAFEI, increase the signal purity with respect to the previous purely inclusive reconstruction technique. This overall underlines the ability of the model to classify rare signal events among numerous background events, similarly proved by the DFEI algorithm of the LHCb collaboration [14]. Nevertheless, the analysis strategy can still be improved, especially the binary classifier. Combined with a higher statistics, we expect the GRAFEI to be a powerful tool for the search of rare decays at Belle II.

Acknowledgements

References

[1] T. Abe et al., *Belle II Technical Design Report*, 2010, arXiv:1011. 0352.

[2] T. Keck et al., *The Full Event Interpretation*, Computing and Software for Big Science (2019), arXiv:1807.08680, https: //arxiv.org/abs/1807.08680.

[3] Belle-II, I. Adachi et al., *Evidence for $B+ \to K+\nu\bar{\nu}$ decays*, Phys. Rev. D **109**, 112006 (2024), arXiv:2311.14647.

[4] J. Kahn et al., *Learning Tree Structures from Leaves For Particle Decay Reconstruction*, en, Machine Learning: Science and Technology **3**, arXiv:2208.14924, 035012 (2022), arXiv:2208. 14924, http : / / arxiv . org / abs / 2208 . 14924 (visited on 04/17/2023).

[5] I. Tsaklidis, P. Goldenzweig, I. Ripp-Baudot, J. Kahn, and G. Dujany, *Demonstrating learned particle decay reconstruction using Graph Neural Networks at Belle II*, Presented on 19/06/2020, PhD thesis (Université de Strasbourg, Strasbourg and Karlsruhe, 2020).

[6] L. Reuter, *Full Event Interpretation using Graph Neural Networks*, MA thesis (Karlsruhe Institute of Technology (KIT), 2022).

[7] P. W. Battaglia et al., *Relational inductive biases, deep learning, and graph networks*, (2018), arXiv:1806.01261.

[8] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, (2015), arXiv:1512.03385.

[9] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, *Fast and accurate deep network learning by exponential linear units (elus)*, 2016, arXiv:1511.07289.

[10] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, *Improving neural networks by preventing co-adaptation of feature detectors*, 2012, arXiv:1207.0580.

[11] S. Ioffe and C. Szegedy, *Batch normalization: accelerating deep network training by reducing internal covariate shift*, (2015), arXiv:1502.03167.

[12] D. P. Kingma and J. Ba, *Adam: a method for stochastic optimization*, (2017), arXiv:1412.6980.

[13] *Cross-entropy Loss*, https://pytorch.org/docs/stable/ generated/torch.nn.CrossEntropyLoss.html.

[14] J. García Pardiñas, M. Calvi, J. Eschle, A. Mauri, S. Meloni, M. Mozzanica, and N. Serra, *Gnn for deep full event interpretation and hierarchical reconstruction of heavy-hadron decays in proton–proton collisions*, Computing and Software for Big Science **7**, 12 (2023), https://doi.org/10.1007/s41781- 023-00107-8.