Graph neural network for 3D classification of ambiguities and optical crosstalk in scintillator-based neutrino detectors

Saúl Alonso-Monsalve,1,2 Dana Douqa,3 César Jesús-Valls,4 Thorsten Lux,4 Sebastian Pina-Otey,4,5 Federico Sánchez,3 Davide Sgalaberna,6 and Leigh H. Whitehead7
1 CERN, The European Organization for Nuclear Research, 1211 Meyrin, Switzerland
2 Universidad Carlos III de Madrid, Av. de la Universidad, 30, 28911 Madrid, Spain
3 University of Geneva, Section de Physique, DPMC, 1205 Genève, Switzerland
4 IFAE, Institut de Física d’Altes Energies, Carrer de Can Magrans, 08193 Barcelona, Spain
5 Aplicaciones en Informática Avanzada (AIA), 08172 Sant Cugat del Vallès Barcelona, Spain
6 ETH Zurich, Institute for Particle Physics and Astrophysics, CH-8093 Zurich, Switzerland
7 Cavendish Laboratory, University of Cambridge, Cambridge, CB3 0HE, United Kingdom

Deep learning tools are being used extensively in high energy physics and are becoming central in the reconstruction of neutrino interactions in particle detectors. In this work, we report on the performance of a graph neural network in assisting with particle flow event reconstruction. The three-dimensional reconstruction of particle tracks produced in neutrino interactions can be subject to ambiguities due to high multiplicity signatures in the detector or leakage of signal between neighboring active detector volumes. Graph neural networks potentially have the capability of identifying all these features to boost the reconstruction performance. As an example case study, we tested a graph neural network, inspired by the GraphSAGE algorithm, on a novel 3D-granular plastic-scintillator detector, that will be used to upgrade the near detector of the T2K experiment. The developed neural network has been trained and tested on diverse neutrino interaction samples, showing very promising results: the classification of particle track voxels produced in the detector can be done with efficiencies and purities of 94-96% per event and most of the ambiguities can be identified and rejected, while being robust against systematic effects.

I. INTRODUCTION

Since 1999, a series of neutrino oscillation experiments have provided deep insight into the nature of neutrinos [1-8]. A number of these experiments are long-baseline neutrino oscillation experiments that use two detectors to characterize a beam of (anti-)neutrinos: a near detector, located a few hundred meters away from the target that measures the original beam composition, and a far detector, located several hundred kilometres away, that allows for the determination of the beam composition after neutrino flavor oscillations.

The energy of these beam neutrinos ranges from a few hundred MeV up to several GeV. Charged particles can be produced in neutrino interactions, and the energy that they deposit as they traverse the detector can be used to reconstruct the events. In general, the larger the energy transferred from the neutrino to the nucleus, the larger the number of particles and particle types produced in the final state. Modeling nuclear interactions in the target nuclei is highly complex, particularly for high energy transfers where the hadronic component of the interaction is more important. As a result, current long-baseline neutrino oscillation experiments mostly analyze interactions with low particle multiplicity. This situation, however, is expected to change in the coming years. On one hand, the statistical and systematic uncertainties of current experiments have decreased significantly over recent years such that neutrino-nucleus modeling is becoming a dominant source of uncertainty [8,9]. On the other hand, future experiments like DUNE [10] will use a broad-band energy neutrino beam, expecting a significant fraction of the neutrino interactions to have a high energy transfer to the nucleus.

As a result, in recent years, the neutrino physics community has turned its attention to measuring neutrino-nucleus interaction cross-sections for different ranges of energies and target materials [11] as a way to constrain the oscillation uncertainties while providing new measurements to further develop the interaction models. In parallel, a new generation of neutrino detectors are under development that aim to resolve and reliably identify short particle tracks even in very complex interactions. To achieve this, two main detector technologies stand out: one is based on Liquid Argon Time-Projection-Chambers (LArTPCs) [12] and the other is based on finely segmented plastic scintillators with three readout views [13] that will form part of the near detectors for T2K [14] and, possibly, DUNE [15].

For the latter, the detector response to a charged particle is read out into three orthogonal 2D projections. When reconstructing the 3D neutrino event, different types of hits are rebuilt, introducing non-physical entities that can hinder the reconstruction process. Due to the spatial disposition of such hits, an approach of utilizing Graph Neural Networks (GNNs) [15] is proposed to perform the classification of 3D hits to provide clean tracks for event reconstruction.

The article proceeds in the following way: Sec. II describes properly the motivation behind the methodology...
given the details of the detector technology. Section III introduces deep learning techniques and explains the specific GNN algorithm used. The simulated data samples and GNN training are discussed in Sec. IV. Results and a study of systematic uncertainties are given in Secs. V and VI respectively, followed by concluding remarks in Sec. VII.

II. MOTIVATION

A finely segmented scintillator detector consists of a 3D matrix of plastic scintillator cubes. The scintillation light produced by charged particles traversing the cubes is read out by three orthogonal wavelength-shifting (WLS) fibers that transport the scintillation light out of the detector where silicon photomultipliers (SiPMs) convert it into a certain number of photoelectrons (p.e.), as illustrated in Figs. 1 and 2.

![Figure 1: Geometry of a single SuperFGD element.](image1)

Here, we consider the Super Fine-Grained Detector (SuperFGD) [14] as a specific case-study. The detector contains 192×56×184 plastic scintillator cubes, each 1×1×1 cm³ in size, and provides three orthogonal 2D projections of particle tracks produced by a neutrino interaction, as depicted in Fig. 4a.

To reconstruct neutrino interactions in three dimensions, the light yield measurements in the three 2D views are matched together, as shown in Fig. 4b. The 3D objects, corresponding to the cubes where the energy deposition is reconstructed, are referred to as voxels. In addition to the cubes where a particle has passed and deposited energy, light-leakage between neighboring cubes can create additional crosstalk signals [17] [18], as depicted in Fig. 2. Moreover, ambiguities in the matching process can give rise to ghost voxels, shown in Fig. 3.

To accurately reconstruct neutrino interactions in these detectors, it is crucial to be able to classify each voxel as one of the three types:

- **Track**: a real energy deposit from a charged particle, henceforth referred to as track signals.
- **Crosstalk**: a real energy deposit from light-leakage between neighboring cubes.
- **Ghost**: fake signals coming from the ambiguity when matching the three 2D views into 3D.

Figure 4c shows the three types of voxels using truth information after 3D matching has been performed for an example neutrino interaction. Once these voxels are classified, the ghost voxels can be removed before the full event reconstruction proceeds, while simultaneously cleaning the particle tracks of crosstalk.

![Figure 2: Sketch of the signal generation, fiber transport and signal detection processes highlighting the production of optical crosstalk signals.](image2)

![Figure 3: Example of a ghost voxel appearance. When matching the coordinates of the fibers that recorded energy deposition, voxels may appear where no true signal exists, becoming non-physical ghost voxels.](image3)

In this article, we represent the voxels as nodes in a graph and classify the signals using a deep learning technique based on a GNN. The abstract data representation provided by graphs makes this method very versatile and applicable to any experiment where the output data from the detector elements can be represented as a list of features with arbitrary dimensionality.
(a) Projections of the observed neutrino interaction onto the three 2D detector views (XY, XZ, and YZ).

(b) 3D view of the neutrino interaction after the 3D matching of the three 2D views. The 3D voxels are shown as dark points. Projections of the observed neutrino interaction onto the three 2D detector views (XY, XZ, and YZ) are shown as shadow.

(c) 3D view of the neutrino interaction after the 3D matching of the three 2D views. The 3D voxels labelled as track (red), crosstalk (blue) and ghost (yellow) according to the truth information from the simulation. Projections of the observed neutrino interaction onto the three 2D detector views (XY, XZ, and YZ) are shown as shadows.

FIG. 4: Visualization of a neutrino interaction in a finely segmented 3D scintillator detector, demonstrating the relationship between the observed 2D projections onto the three orthogonal 2D views, the reconstructed 3D voxels and the true classification of the voxels.
III. BACKGROUND

A. Deep learning

Deep learning techniques are now commonly applied within the field of neutrino physics. In particular, Convolutional Neural Network (CNN) [19] algorithms that operate on two-dimensional images of the neutrino interactions have been very successful in a number of tasks, such as event classification [10, 20, 22] and pixel-level identification of track-like (linear) and shower-like (locally dense) energy deposits [23, 24]. However, images of neutrino interactions are typically very sparse as only those readout channels with a detected signal give rise to pixels with non-zero values, and in the case of the detector presented in Sec. II the average occupancy of the detector for a neutrino interaction is less than 0.02%. Thus, much of the computation time is spent unnecessarily applying convolutions to a large number of pixels with zero values.

The goal of this work is to classify 3D voxels as one of three categories (track, crosstalk or ghost), which is a naively three dimensional problem. To apply a 3D CNN-based algorithm to this detector would require two million voxels to avoid any downsampling or cropping of the input data, which is computationally prohibitive. We here investigate a sparse data representation, where voxels are represented as nodes in a graph. In computer science, a graph \( \mathcal{G} \) is a data structure that represents a mathematical concept consisting of vertices \( V \) (also known as nodes) and edges \( E \):

\[
\mathcal{G} = (V, E).
\]

A graph can be directed, where each edge has a starting and an ending vertex that define a direction, or undirected, where the edge simply connects two nodes without inducing a sense of direction. In our case, we use an undirected graph, since we are only interested in the spatial connections between vertices. Figure 3 shows a comparison of the 3D CNN and graph data structures, as well as the radial search method used for defining edges between nodes.

As mentioned above, each detector cube is represented as a node in a graph, and each node consists of a list of input variables called features that describe the physical properties of the detected signal (see Section IV and Appendix A). The deep learning algorithm that operates on graphs is the Graph Neural Network (GNN) [16, 25]. GNNs are used in many different fields [26, 27] and can operate on two-dimensional images of the neutrino interactions, outperforming a 3D CNN [34]. Other GNN-based studies have been performed for particle reconstruction in high energy physics detectors [31, 36]. To the best of our knowledge, the approach we present in this paper is the first attempt to use GNNs for node classification in neutrino experiments.

B. GraphSAGE

GraphSAGE [32] is a technique that leverages the features of graph nodes \( V \) - which can range from physical information to text attributes - to generate efficient representations on previously unseen samples by learning aggregator functions from training nodes. These aggregators can be simple functions (e.g., mean or maximum) or more complex ones, such as LSTM cells [37], and must be functions that take an arbitrary number of inputs without any given order. The model learns not only \( K \) aggregator functions that combine information from neighboring nodes but also a set of weight matrices \( W^k, \forall k \in \{1, ..., K\} \), which are used to propagate information through the \( K \) layers of the model and combines local information of the node with the aggregator information of its neighbors into an encoding vector (see Algorithm 1). The number of aggregator functions is also used to define the depth of the model, meaning that a GraphSAGE model has a depth of \( K \). Once trained, it can produce the embedding of a new node given its input features and neighborhood; this embedding is then used as the input of a multilayer perceptron (MLP) [38] that is responsible for predicting the label.

Since GraphSAGE learns from node features, it allows us to decide which physical information to use for each voxel. This means that the model can follow the particle flow, i.e., by predicting the label for each voxel based on
GENIE-G18.10b NIE datasets were created using were simulated following two approaches. Firstly, GEANT is a Monte Carlo based toolkit that provides realistic propagation of particles through matter. It outputs a list of energy deposits.

Algorithm 1: GraphSAGE embedding generation (i.e., forward propagation) algorithm (from [32])

| Input | Graph $G(V, E)$; input features $\{x_v, \forall v \in V\}$; depth $K$; weight matrices $W^k, \forall k \in \{1, ..., K\}$; non-linearity $\sigma$; differentiable aggregator functions $\text{AGGREGATE}_k, \forall k \in \{1, ..., K\}$; neighborhood function $N: v \rightarrow 2^V$
| Output: | Vector representations $z_v$ for all $v \in V$

1. $h^0_v \leftarrow x_v, \forall v \in V$
2. for $k = 1...K$ do
3. for $v \in V$ do
4. $h^k_{N(v)} \leftarrow \text{AGGREGATE}_k(h^{k-1}_v, \forall u \in N(v))$
5. $h^k_v \leftarrow \sigma(W^k \cdot \text{CONCAT}(h^{k-1}_v, h^k_{N(v)}))$
6. end
7. $h^k_v \leftarrow h^k_v/\|h^k_v\|_2, \forall v \in V$
8. end
9. $z_v \leftarrow h^K_v, \forall v \in V$

IV. METHODOLOGY

A. Data sample generation

In order to generate data sets of neutrino interactions with true labels that allow to train and benchmark the classification algorithm, the steps below are followed. For each neutrino interaction:

1. Initial particle types and initial kinematics are specified for all final-state particles produced in the interaction.
2. Initial particles are propagated through the detector geometry producing further particles and leaving signals in the form of energy deposits.
3. Using particle energy deposits, the detector response is simulated.
4. The information is stored as a list of voxels with a known true label.

Initial particle types and kinematics

The initial particle types and their associated kinematics were simulated following two approaches. Firstly, GENIE datasets were created using GENIE-G18.10b neutrino interaction software [39]. For a given neutrino flux and target geometry specification, it generates a list of realistic neutrino event interactions both in the number and type of outgoing particles, often referred to as event topologies, and in their individual initial kinematics. Secondly, particle-gun (Pgun) (1-track) and particle-bomb (P-Bomb) (multi-track) datasets were constructed to be as complementary as possible to the GENIE datasets by minimizing the modeling dependency. The number of initial particles and their types were fixed in each of these datasets and their input kinematics were chosen to be randomly and uniformly distributed in the range 10-1000 MeV/c. A summary regarding the number of events and voxels in the two datasets, as well as of the class distribution is presented in Tab. I.

| GENIE dataset | Training | Validation | Testing |
|---------------|----------|------------|---------|
| # Events      | 6k       | 2k         | 11.5k   |
| # Voxels      | 1.83M    | 606.7k     | 3.58M   |
| Track         | 43%      | 37%        | 20%     |
| Crosstalk     | 49%      | 36%        | 13%     |
| Ghost         | 1%       | 3%         | 10%     |

| P-Bomb dataset | Training | Validation | Testing |
|----------------|----------|------------|---------|
| # Events       | 6k       | 2k         | 39.5k   |
| # Voxels       | 1.84M    | 618k       | 12.3M   |
| Track          | 49%      | 38%        | 13%     |
| Crosstalk      | 49%      | 38%        | 13%     |
| Ghost          | 1%       | 3%         | 10%     |

TABLE I: Descriptions of both GENIE and P-Bomb datasets, displaying the number of events and number of voxels used for training, validating and testing the models. Additionally the fractions of the different classes of voxels are shown.

Particle propagation simulation in the detector

The SuperFGD detector geometry was simulated as described in Ref. [14]. The particle propagation and physics simulation is done by means of GEANT v4-10.6.1 [41]. GEANT is a Monte Carlo based toolkit that provides realistic propagation of particles through matter. It outputs a list of energy deposits.

All energy deposits occurring in the same detector cube, including the effect of Birks’ quenching [42], are summed to form the list of track voxels. To simulate imperfect cube light-tightness, the 3D voxelized energy is then randomly shared ($\mu = 2.7\%$) with the neighboring cubes, creating a new set of voxels that originally had no energy deposits, the crosstalk voxels (see Figure 2). Then, the 3D voxelized energy of both track and crosstalk voxels is projected onto its three orthogonal planes where the detector 2D signals are simulated, converting the continuous energy deposit into discretized photons, weighted by distance-dependent attenuation factors, which are detected with 35% probability. To mimic a minimum threshold detection sensitivity, only 2D hits with three or more detected photons are kept. Then, the 2D hits are matched into 3D reconstructed voxels only if the same

---

1 We used the T2K flux, which peaks at 600 MeV/c, see Ref. [40].

11 Only signals in the first 100 ns are considered. Further delayed signals, such as decays, can be treated as independent graphs.
XYZ coordinate combination can be made using two different combinations of 2D planes. In this process, due to ambiguities some extra voxels are created, the ghost voxels (see Fig. 3). Finally, those track and crosstalk voxels not reconstructed after the 3D matching are discarded from the original lists. An example of the 2D to 3D reconstruction is shown in Figures 4a and 4b.

Simulation output
The resulting output from the simulation is a list of voxels and their associated energy deposits in the three planes, each with one of the following three labels that we want to classify, as described in Sec. II: track, crosstalk or ghost voxel.

B. Network architecture
Each graph in GraphSAGE is constructed using the proximity of two voxels in that graph. If both voxels are spatially located within a radius of 1.75 cm\( \text{iii} \) then we consider them to be connected in the graph by an edge; we repeat the same procedure for each pair of voxels\( \text{iv} \). Additionally, we consider a neighborhood depth of three, i.e., to produce the embedding of a voxel, we use the voxel features together with its first neighbors’ features, the features of the neighbors of its neighbors, i.e., second neighbors’ features, and the features of the neighbors of the neighbors of its neighbors, i.e., third neighbors’ features. The aggregator used to combine the feature of the neighbors is the mean aggregator, which produces the average of the neighbors’ values. This final embedding is then passed to an MLP consisting of two fully connected layers - each followed by a LeakyReLU activation function, and a final output layer followed by a softmax activation function. Figure 6 illustrates the GraphSAGE-based approach used, while Tab. II shows the architectural parameters chosen. Categorical cross-entropy is chosen as the loss function to minimize during training, as it is considered the standard one for multi-class classification problems:

\[
J = - \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{c} y_j^{(i)} \log \hat{y}_j^{(i)},
\]

where:

- \( \hat{y}_j^{(k)} \): predicted values corresponding to the \( k^{th} \) training example.
- \( y_j^{(k)} \): true values corresponding to the \( k^{th} \) training example.
- \( m \): number of training examples.
- \( c \): number of classes/neurons corresponding to the output. In this case, the three classes are: track, crosstalk, and ghost.

\| Parameter \| value \\
\hline
Encoding size 128 \\
Depth 3 \\
Aggregator mean \\
Fully Connected Layer 1 128 neurons \\
Fully Connected Layer 2 128 neurons \\
Fully Connected Layer 3 (output) 3 neurons \\
\hline

TABLE II: Architectural parameters; for more information about the meaning of the parameters, see Sec. III B.

The output layer of the model consists of three neurons, one for each of the three classes, with values \( v_i \) for \( i = 1, 2, 3 \). The sum of neuron values is given by \( \sum_{i=1}^{3} v_i = 1 \) such that each neuron value gives a fractional score that can be used to classify voxels. In other words, the model returns scores for each voxel to be one of the three desired outputs, which can be interpreted as the probability: track-like, crosstalk-like, or ghost-like.

C. Training
The network was trained for 50 epochs\( \text{v} \) using Python 3.6.9 and PyTorch 1.3.0\( \text{vi} \) as the deep learning framework, on an NVIDIA RTX 2080 Ti GPU. Adam\( \text{vii} \) is used as the optimizer, with a mini-batch size of 32, and an initial learning rate of 0.001 (divided by 10 when the error plateaus, as suggested in\( \text{viii} \)). The model has a total of 105,347 parameters. Figure 7 shows the validation results during the training process, measured by the \( F_1 \)-score metric:

\[
F_1 = 2 \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}},
\]

The precision and recall are defined as:

\[
\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}, \quad \text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{true negatives}},
\]

\( \text{iii} \) To link only those voxels within the 3×3×3 cube of voxels centred on the target voxel (the maximum diagonal distance from the center of this cube is \( \sqrt{1^2 + 1^2 + 1^2} \approx 1.75 \)).

\( \text{iv} \) If a voxel has no neighbors, it is discarded from the graph and cannot be classified; this happens for less than 0.6% of the total number of voxels.

\( \text{v} \) Epoch: one forward pass and one backward pass of all the training examples. In other words, an epoch is one pass over the entire dataset.
where the labels are compared as one class vs. all the others. The model used later for inference on new data is the one that maximizes the $F_1$-score for the validation set, as it has the best generalization for unseen data.

The efficiencies and purities of these predictions are calculated by two methods: per voxel and per event. The latter method evaluates the correctness of predictions for individual events, while the former does an overall evaluation of the efficiencies and purities for all events in all training sets. The results of both methods for four sets of training/testing samples are shown in Tab. III, giving nearly identical performance that is independent of the dataset used to train and test the GNN.

As an example, Fig. 8 shows the voxel prediction results from the GNN when applied to the event shown in Fig. 4, a GENIE event that features a track almost completely composed of ghost voxels. Figure 8a shows the class predicted for each voxel, while Fig. 8b displays which voxels were correctly/incorrectly classified.

A more in-depth analysis of the GNN performance can be carried out by studying the effects of different event properties on the efficiencies and purities of the predictions. For these studies, the results of the GNN trained and tested on the GENIE dataset are used.

One of the factors expected to affect these predictions is the number of voxels in the event. Figure 9 shows the relationship between the mean efficiency and purity per event for each type of voxel as a function of the total number of voxels in the event. The figure also shows the mean number of events in each bin (in light blue). It is clear that both the efficiencies and purities of the three types of voxels decrease as the number of voxels in the event increases. This decrease is coupled with an increase of the fraction of ghost voxels as the total number of voxels increases, which are the hardest for the GNN to classify.

The number of tracks in the event is an estimate of the complexity of its topology. According to Fig. 10, the classification efficiencies and purities drop as the number of tracks increases. This behavior is also correlated with the increasing fraction of ghost voxels in the events.

The region around the interaction vertex is of particular interest in the event. It is expected that a high spatial density of voxels within a certain volume of the detector may pose a challenge for the GNN to correctly identify the voxel type. This can be observed by studying the efficiencies and purities as a function of the distance to the interaction vertex, as shown in Fig. 11. At the interaction vertex itself, it is clear that there are only track voxels and the GNN can identify them with over 96% efficiency and 100% purity. The following 2 cm exhibit only a small fraction of ghost voxels, mainly due to the
TABLE III: Mean efficiencies and purities of voxel classification, calculated for the whole sample (per voxel) and as a mean of the event-by-event efficiencies and purities (per event).

|                        | GENIE Training | P-Bomb Training |
|------------------------|----------------|-----------------|
|                        | Per Voxel      | Per Event       |"
| GENIE Testing          | Track Crosstalk Ghost | Track Crosstalk Ghost |
| Efficiency             | 93% 90% 84%    | Efficiency 93% 89% 80% |
| Purity                 | 93% 87% 91%    | Purity 91% 86% 89% |
| Per Event              | Track Crosstalk Ghost | Track Crosstalk Ghost |
| Efficiency             | 94% 94% 88%    | Efficiency 94% 93% 88% |
| Purity                 | 96% 91% 92%    | Purity 95% 91% 91% |
| P-Bomb Testing         | Track Crosstalk Ghost | Track Crosstalk Ghost |
| Per Voxel              | Efficiency 94% 93% 87% | Efficiency 95% 91% 92% |
| Purity                 | 96% 90% 92%    | Purity 96% 91% 92% |
| Per Event              | Efficiency 94% 94% 87% | Efficiency 95% 93% 88% |
| Purity                 | 96% 90% 92%    | Purity 96% 91% 92% |

TABLE IV: Mean efficiencies and purities of voxel classification for the GNN and a simple charge cut.

|                        | GNN | Charge Cut |
|------------------------|-----|------------|
|                        | Track Other | Track Other |
| Efficiency             | 94% 96%    | Efficiency 93% 80% |
| Purity                 | 96% 95%    | Purity 80% 91% |

As we go further from the vertex, the efficiencies and purities increase up to 10 cm, after which we expect the density of voxels to decrease allowing for more ghost voxels. Therefore, at large distances we observe that the efficiencies and purities tend to decrease, most notably the efficiencies of crosstalk and ghost voxels and the purity of crosstalk voxels.

As the main goal of this GNN is to identify ghost voxels in order to eliminate them from the events, it is important to make sure that true track and crosstalk voxels are not lost in the process. According to the tests performed, only 1.1% of all true track voxels and 3.3% of crosstalk voxels are incorrectly classified as ghost voxels by the GNN. In addition, it is important not to miss ghost voxels: the GNN correctly identified 84.5% of all ghost voxels, where 72.1% of those classified incorrectly were predicted as crosstalk. Therefore, although not ideal, this issue is not critical as crosstalk voxels have a smaller influence on future studies than track voxels.

Lastly, we compare the results of the GNN against a conventional method of voxel classification which relies on a charge cut. As described in Appendix A, each voxel has three charges that correspond to the signals from the three fibers passing through it. Since other voxels along the same fiber may have signals causing a larger amplitude to be recorded, we consider the smallest of these three charges to be the most accurate estimation of the true voxel charge. Hence, this minimum charge is used for the purposes of this charge cut. Since, by definition, we expect higher energy deposition in track voxels compared to crosstalk and ghost voxels, we set a lower limit for the minimum charge in a voxel such that any voxels with a higher minimum charge than the threshold are classified as track voxels. Fig. 12 shows the distribution of the minimum voxel charge for the three types of voxels. From this figure, it is clear that it is not possible to separate ghost from crosstalk voxels. Thus, this classification is only binary such that we have two categories: track or other. We decide to place this cut at 12 p.e., where the track and non-track voxel PDFs intersect.

To compare the results of this cut with those of our GNN, we combine the predictions of the crosstalk and ghost categories. Table IV shows the efficiency and purity of the classifications for the two methods. It is evident that using only a charge cut can still yield a comparable track voxel classification efficiency to the GNN. However, it struggles to correctly classify non-track voxels which, in turn, reduces the purity of the predicted track voxels.

Another improvement given by GNN over the charge cut is the capability of rejecting “fake” tracks, i.e. a cluster of ghost voxels that closely resembles the structure of a real particle track. Since fake tracks are usually produced by the shadowing of real tracks, the corresponding number of p.e. measured in the three readout views is higher than 12 p.e., hence the charge cut cannot reject them easily. The ability of the GNN to reject ghost tracks is shown in Appendix C for a number of neutrino interactions and compared to the charge cut method.

Figure 13 shows the advantage of the three-fold classification of the GNN over the binary classification of the charge cut when comparing the fraction of true total deposited energy obtained using each method. In the case of the GNN, the total deposited energy in an event is the sum of the true energy deposited in all non-ghost voxels. For the charge cut, only the energy deposited in track voxels is used. This causes an average energy loss of 5% per event when using a method that also excludes the crosstalk voxels, compared to less than 1% when using
(a) Prediction: voxels are colored based on the GNN predictions.

(b) Accuracy: voxels correctly classified by the GNN are shown in green.

FIG. 8: Example GNN prediction results for the interaction shown in Fig. 4.
the GNN that can isolate ghost voxels.

VI. SYSTEMATIC UNCERTAINTY CONSIDERATIONS

The results presented in Sec. V show that the GNN is a very powerful technique for removing ghost voxels and identifying optical crosstalk in 3D-reconstructed neutrino interactions. It is important to demonstrate that this technique is robust and does not introduce new systematic uncertainties, potentially given by a sub-optimal choice of the training sample.

One of the main limitations in the measurement of the neutrino oscillation parameters in long-baseline experiments comes from uncertainties in the modeling of neutrino interactions, not yet fully constrained by data and partially incomplete for describing all the details of the interaction final state. For example, the modeling of hadron multiplicity and kinematics may considerably change the image of the neutrino interaction, particularly near the neutrino vertex, or the total energy deposited by all the particles produced by the neutrino interaction. Hence, it is hard to obtain a data-driven control sample to train a neural network without making any prior assumptions. Since the GNN is trained only on a subset of the parameter space, the results could be biased if the detected neutrino interactions belong to a region of the parameter space not well covered by the MC generator. Thus, there must be careful studies of the systematic uncertainties given by a sub-optimal choice of the training sample.

FIG. 9: Efficiency and purity as a function of the number of voxels in the event for a sample trained and tested on GENIE simulated data.

FIG. 10: Efficiency and purity as a function of the number of tracks in the event for a sample trained and tested on GENIE simulated data.
uncertainties in order to account for a potentially incomplete sampling of the parameter space. In this section, we investigate potential sources of systematic uncertainty.

As described in Sec. V, different training samples, (GENIE or P-Bomb) were generated and the results were summarised in Tab. III, demonstrating that the performance is still very good even when the samples used for training and testing were different.

A few comments about the training sample generation are necessary. Whilst the GENIE sample belongs to a particular choice of the neutrino interaction model, the P-Bomb sample aims, in principle, to be as model-independent as possible. However, generalizing the training sample enough to contain all possible final states of the neutrino interaction is computationally prohibitive.

The training sample will always have some limitations from the subjective decisions about which neutrino interaction topologies are sufficiently improbable to be omitted. Thus, it is necessary to adopt a figure of merit to quantitatively evaluate how two training samples differ in terms of sampling of the parameter space. Following the prescription described in Ref. [46], we computed the distance between the correlation matrices of the GNN input variables of the different training samples (GENIE and P-Bomb), defined as

$$d(C_1, C_2) = 1 - \frac{\text{tr} \{C_1 C_2\}}{\|C_1\| \|C_2\|}$$

where $C_1$ and $C_2$ are the correlation matrices obtained from two different training samples. In the text we will refer to $d(C_1, C_2)$ simply as to the distance between training samples. Although this method is an approximation to a multivariate Gaussian distribution of the probability density function and does not take into account the scales and means of the variables, it still provides a quick way to understand how similar the training samples are. Nevertheless, other heavily computational methods, such as those involving the difference of probability density func-
The robustness of the GNN against model dependencies can be verified by training different neural networks on different event samples and applying them to the same set of neutrino interactions. A difference in the observables used in the physics measurement, such as particle momenta, energy deposit, etc., obtained by the different training can be assigned as a systematic uncertainty introduced by the method.

A study was performed to evaluate the impact of the method on the total true energy deposited in the detector. The difference between the total energy deposit computed after rejecting the voxels classified as ghosts for both network trainings was computed. Fig. 14 shows the distribution of the total true deposited energy before and after discarding the voxels classified as ghosts. Both GENIE- and P-Bomb-trained GNNs give very similar results over the full range of total deposited energy. The total true deposited energy computed with and without ghost rejection differ on average by less than 1 MeV. Hence, it is expected to be improved by increasing the statistics of the training samples. The total difference between GENIE- and P-Bomb-trained GNNs is found to be less than 1 MeV with a standard deviation of approximately 5.5 MeV, mainly due to a few outlier entries, and 68% of the events with a difference better than 0.192 MeV, as shown in Fig. 15.

This corresponds to less than 2% of the mean total deposited energy per event. In Fig. 16 the impact of the different training sample is shown as a function of the total deposited energy. The fractional standard deviation, defined as the standard deviation of the difference between deposited energy computed from different GNN trainings and divided by the true deposited energy, shown in the bottom panel, is less than 2% and almost constant as a function of the deposited energy. This means that the performance of the method is about the same irrespective of the total deposited energy. This study confirms that GNN can be used for classifying 3D voxels potentially with limited systematic uncertainties in the deposited energy, while drastically improving the tracking capability.

### TABLE V: Distance as defined in Eq. 6 between the correlation matrices obtained from the different training samples. Details of the generation of the GENIE and P-Bomb samples is described in Sec. IV A. The Alternative sample was built with a particle bomb similar to the P-Bomb sample but with additional event topologies.

| Sample     | GENIE | P-Bomb | Alternative |
|------------|-------|--------|-------------|
| GENIE      | 0.020075 | 0.020803 | -           |
| P-Bomb     | 0.020803 | 0.000136 | -           |
| Alternative| 0.020075 | 0.000136 | -           |

**FIG. 14:** Distribution of the total true deposited energy after rejecting the ghost voxels classified either with GENIE- (dashed orange) or P-Bomb- (dotted green) trained GNNs and without any ghost rejection (solid blue). The mean total deposited energy per event is about 288 MeV.

**FIG. 15:** Difference between the total true deposited energy computed after rejecting the ghost voxels classified with GENIE- and P-Bomb-trained GNNs. The mean is 0.78 MeV while the standard deviation is 5.5 MeV. About 40% of events show no difference between P-Bomb and GENIE, 68% have a difference within ±0.192 MeV, while only 5% of the events have a difference outside the range ±6.35 MeV.

Another potential issue could be given by a mismodeling of the amount of crosstalk. In addition to the nominal optical crosstalk (2.7%), two further datasets were sim-
A graph neural network inspired by GraphSAGE was developed and tested on simulated neutrino interactions in a 3D voxelized fine-granularity plastic-scintillator detector with three 2D readout views. The advantage of this neural network is that, the graph data structures provide a natural representation of the neutrino interactions.

The neural network was able to identify ambiguities and scintillation light leakage between neighboring active scintillator detector volumes as well as real signatures left by particles with efficiencies and purities in the range of 94-96% per event, with a clear improvement with respect to less sophisticated methods. In particular, it can reject fake tracks produced by the shadowing of real tracks observed in the 2D readout views. The performance was tested for neutrino events with different number of voxels, number of tracks and voxels at different distances from the vertex, variables that could hint to interaction model dependencies of the method. Efficiencies and purities were found to be relatively stable and the trends were consistent with the expectation. The robustness of the neural network against possible systematic uncertainties introduced by the method was tested. The results were obtained using neural networks trained on different samples, produced either with the GENIE event generator or by randomizing the number of final state particles and relative momentum to obtain a more generic sample that does not belong to any particular theoretical model. It was found that the bias introduced on the total deposited energy of the event by arbitrarily choosing a different training sample is, on average, less than 1 MeV. The impact of potential mismodeling of the light leakage between neighboring scintillator volumes was tested. Results show that the performance of the neural network is robust to expected changes in the crosstalk modeling.

To conclude, we showed that a graph neural network has great potential in assisting a 3D particle-flow reconstruction of neutrino interactions. Similar results may be expected for other types of detectors that aim to a 3D reconstruction of the neutrino event from 2D projections and that share analogous features like ambiguities and leakage of signal between detector voxels.

### Acknowledgments

D. Douqa and F. Sánchez acknowledge the Swiss National Foundation Grant No. 200021_85012. C. Jesús-Valls and T. Lux acknowledge funding from the Spanish Ministerio de Economía y Competitividad (SEIDI-MINECO) under Grants No. FPA2016-77347-C2-2-P and SEV-2016-0588. S. Pina-Otey acknowledges the support of the Industrial Doctorates Plan of the Secretariat of Universities and Research of the Department of Business and Knowledge of the Generalitat of Catalonia. The authors also thank T. Jiang, T. Zhao and D. Wang for their assistance.
implementation of GraphSAGE\textsuperscript{[1]} on which the software of this paper was based.

This work was initiated in the framework of the T2K Near Detector upgrade project, fruitful discussions in this context with our colleagues are gratefully acknowledged. The authors acknowledge the T2K Collaboration for providing the neutrino interaction and detector simulation software.

**Appendix A: Input variables**

The list of variables used as features for the graph nodes is given below. Each node is placed at XYZ coordinates matching the center of a cube, however, these center coordinates are not node variables by themselves since the detector response is isotropic. The numbers in front of each variable match those in Fig 17.

- **0-2:** peXY, peXZ, peYZ
  Number of photons detected in the XY, XZ or YZ-fiber intersecting the cube under consideration corrected by the expected attenuation.

- **3-5:** mXY, mXZ, mYZ
  Number of active voxels intersected by the fiber associated to peXY, peXZ or peYZ.

- **6:** pewav
  Average number of detected photons peXY, peXZ, peYZ, each weighted by the fiber multiplicity mXY, mXZ, mYZ.

\[
\text{pewav} = \frac{\text{peXY}}{\text{mXY}} + \frac{\text{peXZ}}{\text{mXZ}} + \frac{\text{peYZ}}{\text{mYZ}}
\]

- **7-9:** pullX, pullY, pullZ
  Relative difference between the light measured in two different 2D planes.

\[
\begin{align*}
\text{pullX} &= \frac{\text{peXY} - \text{peXZ}}{\text{peXY} + \text{peXZ}} \\
\text{pullY} &= \frac{\text{peXY} - \text{peYZ}}{\text{peXY} + \text{peYZ}} \\
\text{pullZ} &= \frac{\text{peXZ} - \text{peYZ}}{\text{peXZ} + \text{peYZ}}
\end{align*}
\]

- **10:** residual
  Similarity of the light yield measured in the three 2D planes, measured as the squared distance from each peXY, peXZ, peYZ to the average, weighted by the squared average.

\[
\mu = \frac{\text{peXY} + \text{peXZ} + \text{peYZ}}{3}
\]

\[
\text{residual} = \frac{(\text{peXY} - \mu)^2 + (\text{peXZ} - \mu)^2 + (\text{peYZ} - \mu)^2}{\mu^2}
\]

- **11:** pullXYZ
  Similarity of the light yield measured in the three 2D planes, measured as a combination of 2D pulls \((a_1, a_2, a_3)\) weighted by pewav.

\[
\begin{align*}
a_1 &= \frac{\text{peXY}}{\text{mXY}} - \frac{\text{peXZ}}{\text{mXZ}} \\
\frac{\text{peXY}}{\text{mXY}} + \frac{\text{peXZ}}{\text{mXZ}}
\\
a_2 &= \frac{\text{peXY}}{\text{mXY}} - \frac{\text{peYZ}}{\text{mYZ}} \\
\frac{\text{peXY}}{\text{mXY}} + \frac{\text{peYZ}}{\text{mYZ}}
\\
a_3 &= \frac{\text{peXZ}}{\text{mXZ}} - \frac{\text{peYZ}}{\text{mYZ}} \\
\frac{\text{peXZ}}{\text{mXZ}} + \frac{\text{peYZ}}{\text{mYZ}}
\end{align*}
\]

\[
\text{pullXYZ} = \frac{a_1a_2 + a_1a_3 + a_2a_3}{\text{pewav}}
\]

- **12:** ratioMQ
  Ratio between the average voxel multiplicity in the three fibers and pewav.

\[
\text{ratioMQ} = \frac{\text{mXY} + \text{mXZ} + \text{mYZ}}{3}
\]

- **13-14:** R1, R3
  Number of active neighbor voxels in a sphere of certain radius.
  \(\Rightarrow R1, r=1\ cm.\)
  \(\Rightarrow R2, r=2\ cm.\)
  \(\Rightarrow R3, r=5\ cm.\)
  R2 was not used as a variable due to the high correlation with R1, but is used to compute RR.

- **15-20:** x+, x-, y+, y-, z+, z-
  Boolean variables representing the existence of immediate neighbors in each of the 6 surrounding cubes.

- **21:** orthogonal_neighbor
  It is 1 if any of x+, x-, y+, y-, z+, z- is 1.

- **22:** RR
  Ratio between the number of close and far voxels. The \(\epsilon = 10^{-7}\) prevents numerical problems when R3=0.

\[
\text{RR} = \frac{R2}{R3 + \epsilon}
\]

\[\text{https://github.com/tujiang/graphSAGE-pytorch}\]
• 23: ratioDQ
  Ratio between the average voxel distance $\text{aveDist}$ around the voxel and the weighted average light yield $pewav$.

  \[
  \text{ratioDQ} = \frac{\text{aveDist}}{pewav}
  \]

• 24: aveDist
  Average distance from the voxel center $C$ to all fired voxel centers ($C_i$) within a sphere of radius 2.5 cm.

  \[
  \text{aveDist} = \frac{1}{N} \sum_{i=1}^{N} \text{EuclidianDist}(C, C_i)
  \]

A number of these variables are calculated from the same underlying properties of the energy deposits. In theory, an infinitely deep GNN trained on an infinite amount of training data would be able to extract all of the information required for classification from the few underlying properties. In practice, we use a larger number of derived variables to guide the GNN to allow it to more easily extract information from the data and to converge quickly in the training process. Global position was intentionally not used as a variable to avoid the GNN to learn neutrino modelling specific behaviours.

Appendix B: Comparison of GENIE and P-Bomb simulated data samples

Figure 17 shows the correlations of the input variables defined in Appendix A for the GENIE and P-Bomb data samples. Differences between the two matrices arise from the different topologies of interactions produced by the two generator methods.

Appendix C: Event Gallery

This section contains a number of visualizations to show the classification performance of the GNN for a number of neutrino interactions with different complexity and topology. Displays are shown for different events in Figs 18-23; all voxels with their true classification, only the true track voxels, the classified track voxels using the charge cut method, and the classified track voxels using the GNN. The interactions shown here are examples of interactions containing many ghost voxels in order to showcase the GNN performance.

The track voxel classification ability of the charge cut and GNN methods can be seen by comparing subfigures (c) and (d) with (b), respectively, for each interaction. The GNN is able to reject ghost voxels very well, as shown in Figs 19, 21, 22 and 23 where ghost tracks remain using the charge cut method. In general, the performance improvement from the GNN increases with the complexity of the interactions. For simple interactions with only a single muon in the final state both methods perform similarly.
FIG. 18: 3D visualization of a neutrino interaction in a finely segmented 3D scintillator detector after the 3D matching of the three 2D views. The GNN cut is able to almost entirely reject the fake track traveling on the XZ plane and stopping near to the vertex at X≈160 mm and Z≈70 mm, while the charge cut cannot.
(a) The 3D voxels labelled as track (red), crosstalk (blue) and ghost (yellow) according to the truth information from the simulation are shown.

(b) Only the 3D voxels labelled as track according to the truth information from the simulation are shown.

(c) The 3D voxels labelled as track according to the charge cut classification are shown.

(d) The 3D voxels labelled as track according to the GNN classification are shown.

FIG. 19: 3D visualization of a neutrino interaction in a finely segmented 3D scintillator detector after the 3D matching of the three 2D views. The charge cut is not able to reject two fake tracks, one coming from a vertex a X≈50 mm Z≈50 mm traveling on the XZ plane and stopping near to the vertex at X≈160 and Z≈70. Moreover, the charge cut leave a bump of ghost voxels around the vertex that could mimic the interaction of a few low-energy protons, an effect that could bias the reconstruction of the neutrino energy.
(a) The 3D voxels labelled as track (red), crosstalk (blue) and ghost (yellow) according to the truth information from the simulation are shown.

(b) Only the 3D voxels labelled as track according to the truth information from the simulation are shown.

(c) The 3D voxels labelled as track according to the charge cut classification are shown.

(d) The 3D voxels labelled as track according to the GNN classification are shown.

FIG. 20: 3D visualization of a neutrino interaction in a finely segmented 3D scintillator detector after the 3D matching of the three 2D views. In this even the performance of GNN and the charge cut is quite similar because the ghost voxels are mainly given by the overlap of crosstalk hits in the 2D readout views.
FIG. 21: 3D visualization of a neutrino interaction in a finely segmented 3D scintillator detector after the 3D matching of the three 2D views. This neutrino event has a quite high multiplicity and tracks are quite close each other. This produce relatively big clusters of ghost voxels that produce at least two fake tracks even after the charge cut. Instead GNN allows to classify ghosts more precisely and correctly visualize the correct number of tracks. Moreover, the charge cut makes true tracks more fat making their separation harder and, potentially, less precise the particle momentum reconstruction.
(a) The 3D voxels labelled as track (red), crosstalk (blue) and ghost (yellow) according to the truth information from the simulation are shown.

(b) Only the 3D voxels labelled as track according to the truth information from the simulation are shown.

(c) The 3D voxels labelled as track according to the charge cut classification are shown.

(d) The 3D voxels labelled as track according to the GNN classification are shown.

FIG. 22: 3D visualization of a neutrino interaction in a finely segmented 3D scintillator detector after the 3D matching of the three 2D views. Although this is a relatively simple neutrino event, the charge cut is not able to reject a fake track stopping near the neutrino interaction vertex while GNN can provide a much cleaner reconstruction.
(a) The 3D voxels labelled as track (red), crosstalk (blue) and ghost (yellow) according to the truth information from the simulation are shown.

(b) Only the 3D voxels labelled as track according to the truth information from the simulation are shown.

(c) The 3D voxels labelled as track according to the charge cut classification are shown.

(d) The 3D voxels labelled as track according to the GNN classification are shown.

FIG. 23: 3D visualization of a neutrino interaction in a finely segmented 3D scintillator detector after the 3D matching of the three 2D views. In the neutrino event GNN can easily reject the relatively big cluster of ghost voxels that would make difficult a proper reconstruction of the number of tracks and corresponding energy, in particular near to the interaction vertex.
[43] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, in Advances in Neural Information Processing Systems 32 (Curran Associates, Inc., 2019) pp. 8024–8035.

[44] D. P. Kingma and J. Ba, CoRR abs/1412.6980 (2014). arXiv:1412.6980

[45] K. He, X. Zhang, S. Ren, and J. Sun, CoRR abs/1512.03385 (2015) arXiv:1512.03385

[46] M. Herdin, N. Czink, H. Ozcelik, and E. Bonek, in 2005 IEEE 61st Vehicular Technology Conference, Vol. 1 (2005) pp. 136–140 Vol. 1.