Graph neural network for 3D node classification 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]**, Leigh H. Whitehead**[7]
[1]CERN, The European Organization for Nuclear Research
[2]Universidad Carlos III de Madrid
[3]University of Geneva
[4]IFAE, Institut de Física d'Altes Energies
[5]Aplicaciones en Informática Avanzada (AIA)
[6]ETH Zurich
[7]University of Cambridge

Abstract

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 graph neural networks in assisting with particle flow event reconstruction. As an example case study, we tested a graph neural network, inspired by the GraphSAGE algorithm, on a novel 3D-granular plastic-scintillator detector. The results are promising: 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.

1 Introduction

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 [1]. 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) [2] and the other is based on finely segmented plastic scintillators with three readout views [3] that will form part of the near detectors for T2K [4] and, possibly, DUNE [5].

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 deep learning is proposed to perform the classification of 3D hits to provide clean tracks for event reconstruction.

Deep learning techniques are now commonly applied within the field of neutrino physics. In particular, convolutional neural network (CNN) [6] algorithms that operate on two-dimensional images of the neutrino interactions have been very successful in a number of tasks [7, 8, 9, 10, 11, 12]. 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. Thus, other algorithms such as the graph neural network (GNN) [13, 14] or the submanifold sparse convolutional network (SSCN) [15] must be considered since much of the computation time is spent unnecessarily applying convolutions to a large number of pixels with zero values.

* saul.alonso.monsalve@cern.ch

Third Workshop on Machine Learning and the Physical Sciences (NeurIPS 2020), Vancouver, Canada.

We here investigate a sparse data representation, where hits are represented as nodes in a graph. Due to the spatial disposition of such hits, an approach of utilizing a GNN is proposed to perform the classification of 3D hits to provide clean tracks for event reconstruction.

2 Case study

We consider the Super Fine-Grained Detector (SuperFGD) [4], which will be used to upgrade the near detector of the T2K experiment, as a specific case-study. It will have 2 million plastic scintillator cubes, each $1\times1\times1\,\mathrm{cm}^3$ in size, and provides three orthogonal 2D projections of particle tracks produced by a neutrino interaction. To reconstruct neutrino interactions in three dimensions, the light yield measurements in the three 2D views are matched together. The 3D objects, corresponding to the cubes where the energy deposition is reconstructed, are referred to as *voxels*.

To accurately reconstruct neutrino interactions in these detectors, it is crucial to be able to classify each voxel as one of the three types: (1) *track*, a real energy deposit from a charged particle; (2) *crosstalk*, a real energy deposit from light-leakage between neighboring cubes [16, 17]; (3) *ghost*, fake signals coming from the ambiguity when matching the three 2D views into 3D. Figure 1 shows the three 2D views detector read-out and the three types of voxels for an example neutrino interaction.

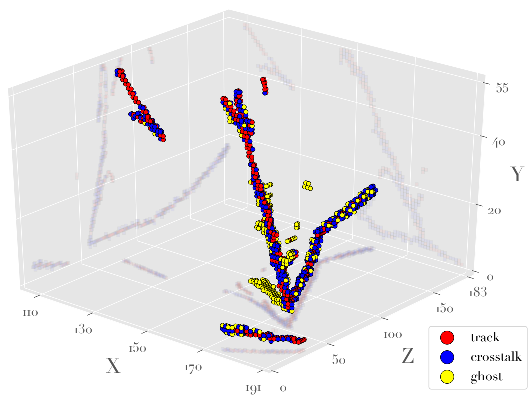


Figure 1: 3D view of the neutrino interaction after the 3D matching of the three 2D views. The 3D voxels labelled as track (red), crosstalk (blue) and ghost (yellow) according to the truth information from the simulation. Projections of the observed neutrino interaction onto the three 2D detector views (XY, XZ, and YZ) are shown as shadows. The axes are in cm.

We generated two datasets for this study. A summary regarding the number of events and voxels in the two datasets, as well as of the class distribution are presented in Table 1.

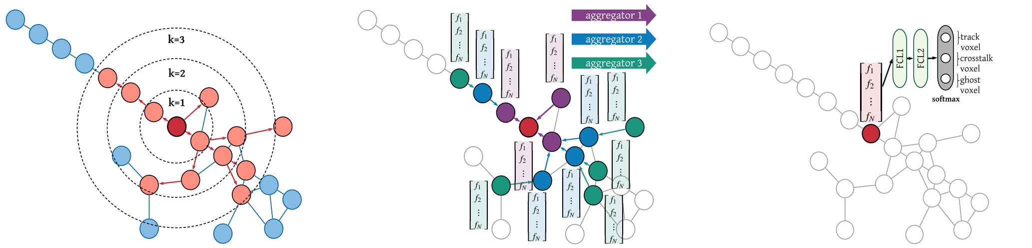
GENIE dataset		**Training**	**Validation**	**Testing**	**P-Bomb dataset**		**Training**	**Validation**	**Testing**
	# Events	6k	2k	11.5k		# Events	6k	2k	39.5k
	# Voxels	1.83M	606.7k	3.58M		# Voxels	1.84M	618k	12.3M
		Track	**Crosstalk**	**Ghost**			**Track**	**Crosstalk**	**Ghost**
	Fraction	43%	37%	20%		Fraction	49%	38%	13%

Table 1: 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.

2.1 Network architecture

As mentioned above, each detector voxel 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. The deep learning algorithm that operates on graphs is the graph neural network (GNN) [13, 14]. GNNs are used in many different fields [18, 19, 20, 21, 22, 23, 24]. In this work, a GNN inspired by the GraphSAGE algorithm [24] is used to classify individual voxels in SuperFGD events. The application of GNNs to data from neutrino experiments has been recently demonstrated by the IceCube experiment in order to identify entire events as atmospheric neutrino interactions [25]. To the best of our knowledge, the approach we present in this work is one of the first attempts of using GNNs for node classification in neutrino experiments.

GraphSAGE [24] is a technique that leverages the features of graph nodes $\mathcal{V}$ - 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 [26], and must be functions that take an arbitrary number of inputs without any given order. In our case, each graph is constructed using the proximity of two voxels in that graph. If both voxels are spatially located within a radius of 1.75 cm[2], then we consider them to be connected in the graph by an edge; we repeat the same procedure for each pair of voxels[3]. Additionally, we consider a neighborhood depth of three. The aggregator used to combine the feature of the neighbors is the mean aggregator, which produces the average of the neighbors' values. The final embedding is passed to an MLP consisting of two fully connected layers - followed by a LeakyReLU activation function - and a final output layer followed by a softmax activation function. Figure 2 illustrates the GraphSAGE-based approach used.



(a) Sample neighborhood.

(b) Aggregate feature information from neighbors.

(c) Use aggregated information as input for the fully connected layers and predict the label.

Figure 2: Visual illustration of the GraphSAGE sample and aggregate approach with a depth of three [24].

2.2 Training

The network was trained for 50 epochs using Python 3.6.9 and PyTorch 1.3.0 [27] as the deep learning framework, on an NVIDIA RTX 2080 Ti GPU. Adam [28] 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 [29]). The model has a total of 105,347 parameters. The model used later for inference on new data is the one that maximizes the F_1-score for the validation set (achieving a score of 0.90 and 0.93 for the GENIE and P-Bomb validation sets, respectively) as it has the best generalization for unseen data.

2.3 Results

The GNN voxel-type predictions are compared against the true labels to evaluate the network performance and identify possible areas of improvement. 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 on an event-by-event basis, while the former does an overall calculation of the efficiencies and purities for all voxels in all events of the sample. The results of both methods for four sets of training/testing samples are shown in Table 2, giving nearly identical performance that is independent of the dataset used to train and test the GNN.

To compare the results of a conventional charge cut with those of our GNN, we combine the predictions of the crosstalk and ghost categories. Table 3 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.

[2] 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 $\sqrt{1^2+1^2+1^2} \approx 1.75$).

[3] 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.

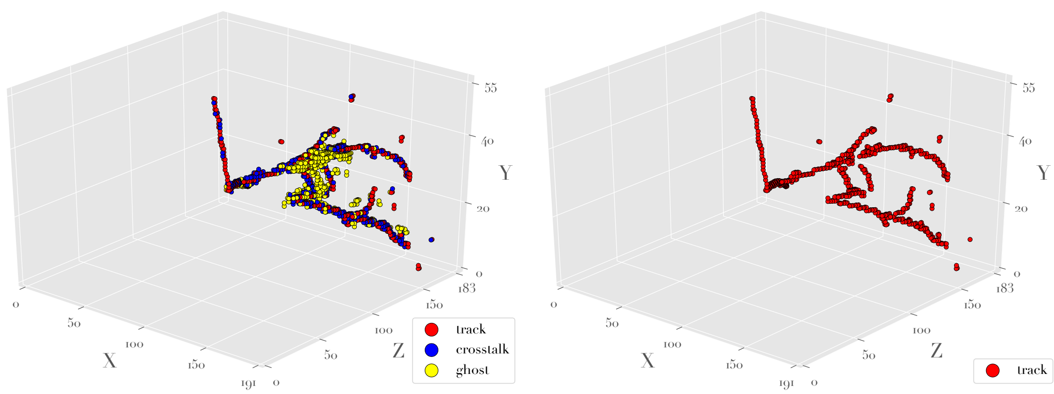
		GENIE Training				P-Bomb Training			
			Track	Crosstalk	Ghost		Track	Crosstalk	Ghost
GENIE Testing	**Per Voxel**	Efficiency	93%	90%	84%	Efficiency	93%	89%	80%
		Purity	93%	87%	91%	Purity	91%	86%	89%
			Track	Crosstalk	Ghost		Track	Crosstalk	Ghost
	Per Event	Efficiency	94%	94%	88%	Efficiency	94%	93%	88%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

Table 2: 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).

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

Table 3: Mean efficiencies and purities of voxel classification for the GNN and a simple charge cut.

Figure 3 shows a neutrino event that has a high multiplicity and tracks are quite close each other. The GNN allows us to classify ghosts more precisely and visualize the correct number of tracks (Figure 3a). 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 3b).



(a) The 3D voxels labelled as track (red), crosstalk (blue) and ghost (yellow) according to the GNN classification are shown.

(b) The 3D voxels labelled only as track according to the GNN classification are shown (clean event).

Figure 3: 3D visualization of a neutrino interaction in a finely segmented 3D scintillator detector after the 3D matching of the three 2D views.

3 Conclusion

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 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. Efficiencies and purities were found to be relatively stable and the trends were consistent with the expectation.

Broader Impact

In this work, we showed that a graph neural network has great potential in assisting a 3D particle-flow reconstruction of neutrino interactions. One advantage of this technique is that the graph data structures provide a natural representation of the 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. This method is promising also for any scintillator detector that shares features similar to the ones described in this work, even outside the field of high energy physics. Disadvantages from this research and consequences of failure of the system are not applicable here.

References

[1] M. Tanabashi *et al.*, "Review of Particle Physics," *Phys. Rev. D*, vol. 98, no. 3, p. 030001, 2018.

[2] C. Rubbia, "The liquid-argon time projection chamber: a new concept for neutrino detectors," Tech. Rep. CERN-EP-INT-77-8, CERN, Geneva, 1977.

[3] A. Blondel, F. Cadoux, S. Fedotov, M. Khabibullin, A. Khotjantsev, A. Korzenev, A. Kostin, Y. Kudenko, A. Longhin, A. Mefodiev, P. Mermod, O. Mineev, E. Noah, D. Sgalaberna, A. Smirnov, and N. Yershov, "A fully-active fine-grained detector with three readout views," *Journal of Instrumentation*, vol. 13, pp. P02006–P02006, feb 2018.

[4] K. Abe *et al.*, "T2K ND280 Upgrade - Technical Design Report," 2019.

[5] G. Yang, "3D Projection Scintillator Tracker in the DUNE Near Detector," *PoS*, vol. ICHEP2018, p. 868, 2019.

[6] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, pp. 541–551, Dec 1989.

[7] A. Aurisano *et al.*, "A convolutional neural network neutrino event classifier," *Journal of Instrumentation*, vol. 11, no. 09, p. P09001, 2016.

[8] R. Acciarri *et al.*, "Convolutional Neural Networks Applied to Neutrino Events in a Liquid Argon Time Projection Chamber," *JINST*, vol. 12, no. 03, p. P03011, 2017.

[9] B. Abi, R. Acciarri, M. Acero, G. Adamov, D. Adams, M. Adinolfi, Z. Ahmad, J. Ahmed, T. Alion, S. Alonso Monsalve, *et al.*, "Far detector technical design report, volume II: DUNE physics," *arXiv*, 2020.

[10] B. Abi, R. Acciarri, M. Acero, G. Adamov, D. Adams, M. Adinolfi, Z. Ahmad, J. Ahmed, T. Alion, S. Alonso Monsalve, *et al.*, "Neutrino interaction classification with a convolutional neural network in the DUNE far detector," *Physical Review D*, vol. 102, Nov 2020.

[11] B. Abi, R. Acciarri, M. A. Acero, M. Adamowski, C. Adams, D. L. Adams, P. Adamson, M. Adinolfi, Z. Ahmad, C. H. Albright, *et al.*, "The Single-Phase ProtoDUNE Technical Design Report," *arXiv*, 2017.

[12] C. Adams *et al.*, "Deep neural network for pixel-level electromagnetic particle identification in the microboone liquid argon time projection chamber," *Phys. Rev. D*, vol. 99, p. 092001, May 2019.

[13] A. Sperduti and A. Starita, "Supervised neural networks for the classification of structures," *IEEE Transactions on Neural Networks*, no. 3, pp. 714–735, 1997.

[14] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," 2018.

[15] B. Graham and L. van der Maaten, "Submanifold sparse convolutional networks," 2017.

[16] A. Blondel *et al.*, "The SuperFGD Prototype Charged Particle Beam Tests," 2020.

[17] O. Mineev *et al.*, "Beam test results of 3D fine-grained scintillator detector prototype for a T2K ND280 neutrino active target," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 923, pp. 134–138, 2019.

[18] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional Networks on Graphs for Learning Molecular Fingerprints," in *Advances in Neural Information Processing Systems 28* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 2224–2232, Curran Associates, Inc., 2015.

[19] Y. Shen, H. Li, S. Yi, D. Chen, and X. Wang, "Person Re-identification with Deep Similarity-Guided Graph Neural Network," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[20] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic Graph CNN for Learning on Point Clouds," 2018.

[21] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical Graph Representation Learning with Differentiable Pooling," in *Advances in Neural Information Processing Systems 31* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), pp. 4800–4810, Curran Associates, Inc., 2018.

[22] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online Learning of Social Representations," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, (New York, NY, USA), p. 701–710, Association for Computing Machinery, 2014.

[23] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," 2016.

[24] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," 2017.

[25] N. Choma, F. Monti, L. Gerhardt, T. Palczewski, Z. Ronaghi, P. Prabhat, W. Bhimji, M. M. Bronstein, S. R. Klein, and J. Bruna, "Graph neural networks for icecube signal classification," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 386–391, 2018.

[26] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[27] 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, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, Curran Associates, Inc., 2019.

[28] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *CoRR*, vol. abs/1412.6980, 2014.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *CoRR*, vol. abs/1512.03385, 2015.