# NetXplain: Real-time explainability of Graph Neural Networks applied to computer networks

David Pujol-Perich<sup>1</sup>, José Suárez-Varela<sup>1</sup>, Shihan Xiao<sup>2</sup>, Bo Wu<sup>2</sup>, Albert Cabellos-Aparicio<sup>1</sup>, Pere Barlet-Ros<sup>1</sup>

# 1. Introduction

#### **D** Motivation:

- Graph Neural Networks (GNN) are typically treated as black-boxes (their internal behavior is not understandable by humans).
- As all Machine Learning (ML) models, they provide statistical guarantees, which is not sufficient for applications in critical systems (e.g., computer networks).

### Background:

#### **Explainability mask** $\bigcirc$

The explainability mask quantifies ([0,1]) the relevance of a node or an edge of the input graph to the output labels of the GNN.



#### State-of-the-art (SOTA) $\bigcirc$

#### Explainability Ο

- Explainability techniques aim to **identify the critical** elements of input graphs (e.g., nodes, edges) that mostly affect the GNN (represented with the explainability mask).
- These techniques enable to interpret the inner behavior of the GNN given an input sample.
- Existing solutions [1,2] are based on costly iterative optimization algorithms, which must be applied over each input sample.
- The computational cost of SOTA solutions is often **prohibitive for** real-time applications.

# 2. NetXplain

### **Proposed solution:**

- Train a GNN to explain the behavior of a target GNN (previously trained for some task).
- One GNN execution vs SOTA iterative algorithms



### **Workflow of NetXplain:**

- 1. Generate a small explainability dataset by running a (costly) SOTA algorithm over the target model (original GNN).
- 2. Train an independent GNN over the explainability dataset (generated in step 1) to produce the output explainability masks.
- Given an input graph, the resulting **NetXplain's GNN produces** 3. explainability masks equivalent to the SOTA algorithm used for training.

### **Real-time applications:**

**NetXplain** enables to perform explainability in **real-time applications**.



- We focus on computer networks, as they are critical infrastructures that can greatly benefit from real-time explainability solutions:
  - 1. Test and troubleshooting: Given an input network scenario, identify the elements (e.g., links) that mostly affect the output decision of the GNN.
  - 2. Network optimization: Enhance the exploration strategy of optimizers (e.g., DRL agents) considering the critical elements reported by NetXplain.

# 3. Evaluation

✓ We train a NetXplain model to produce explainability masks on RouteNet [3] (a GNN model that predicts the per-source-destination packet delay in networks).

✓ Tested in three datasets with real-world network topologies (NSFNet, GBN and GEANT).

#### **Comparison:** NetXplain vs SOTA algorithms

- > NetXplain produces equivalent explainability masks to the SOTA solution (METIS [2]).
- > NetXplain makes accurate predictions even over a new network topology **unseen** during training (GBN).





### References

[1] Ying, R., et al. 2019. GNNExplainer: A tool for post-hoc explanation of graph neural networks. arXiv.

[2] Meng, Z., et al. 2020. Interpreting Deep Learning-Based Networking Systems. In Proceedings of ACM SIGCOMM.

[3] Krzysztof Rusek et al. 2019. Unveiling the potential of GNN for network modeling and optimization in SDN. In ACM SOSR.



NetXplain achieves a speed-up of 7200x compared to the SOTA solution.

NSFNet	Benchmark (Metis)	98.139	2.455	
	NetXplain	0.012	0.001	
GBN	Benchmark (Metis)	150.83	1.79	
	NetXplain	0.0214	0.005	
GEANT2	Benchmark (Metis)	191.46	2.76	
	NetXplain	0.029	0.002	

#### Take-home messages:

- NetXplain produces explainability representations equivalent to costly SOTA algorithms.
- Thanks to the use of GNN, it generalizes over new network scenarios unseen during training.
- Its low execution cost enables to integrate it in real-time networking applications.

This work has received funding from the European Union's Horizon 2020 research and innovation programme within the framework of the NGI-POINTER Project funded under grant agreement No. 871528. This paper reflects only the authors' view; the European Commission is not responsible for any use that may be made of the information it contains. This work was also supported by the Spanish MINECO under contract TEC2017-90034-C2-1-R ALLIANCE and the Catalan Institution for Research and Advanced Studies (ICREA).

<sup>1</sup> Barcelona Neural Networking Center, Universitat Politècnica de Catalunya, Spain <sup>2</sup> Network Technology Lab., Huawei Technologies Co., Ltd.

CONTACT US: <u>contactus@bnn.upc.edu</u>



