UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Departament d'Arquitectura de Computadors

Fast prototyping of complex Graph Neural Networks for Networking

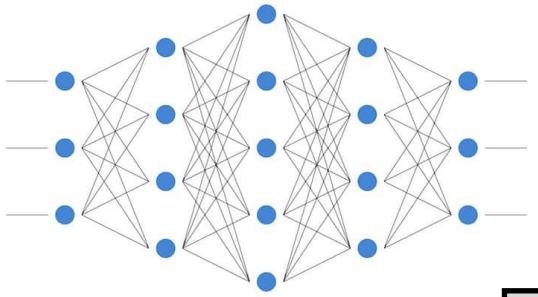
José Suárez-Varela (jsuarezv@ac.upc.edu)

Join work with: David Pujol Perich, Miquel Ferriol Galmés, Albert López Brescó, Pere Barlet Ros, and Albert Cabellos Aparicio

How can we design commercializable AI solutions for networking?

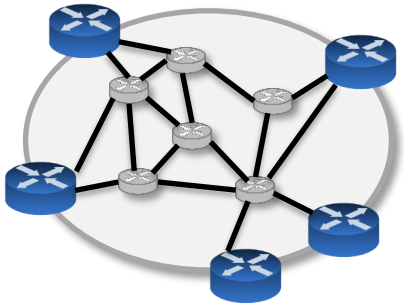
VENDOR'S LAB

AI model (e.g., neural network)



+

Controlled testbed

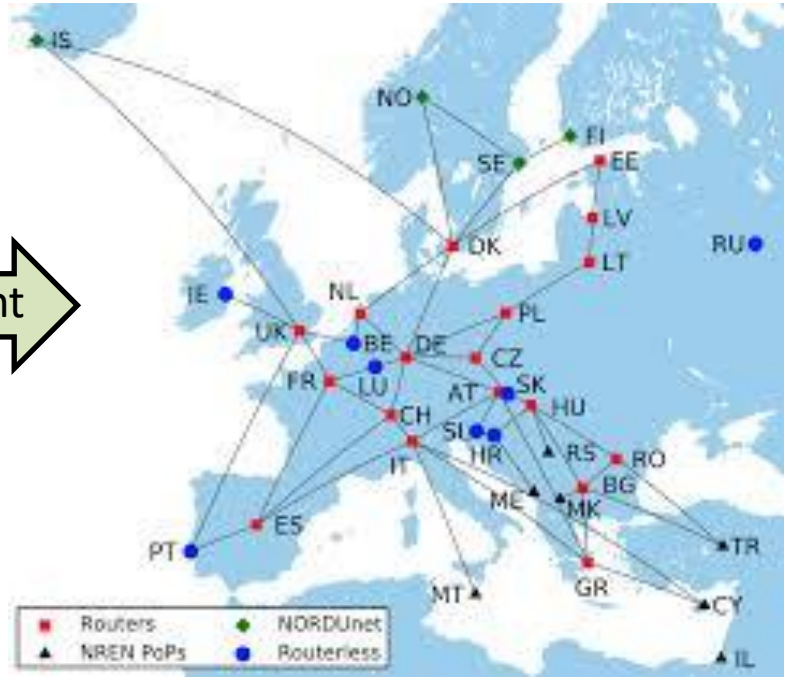


Training

FINAL PRODUCT

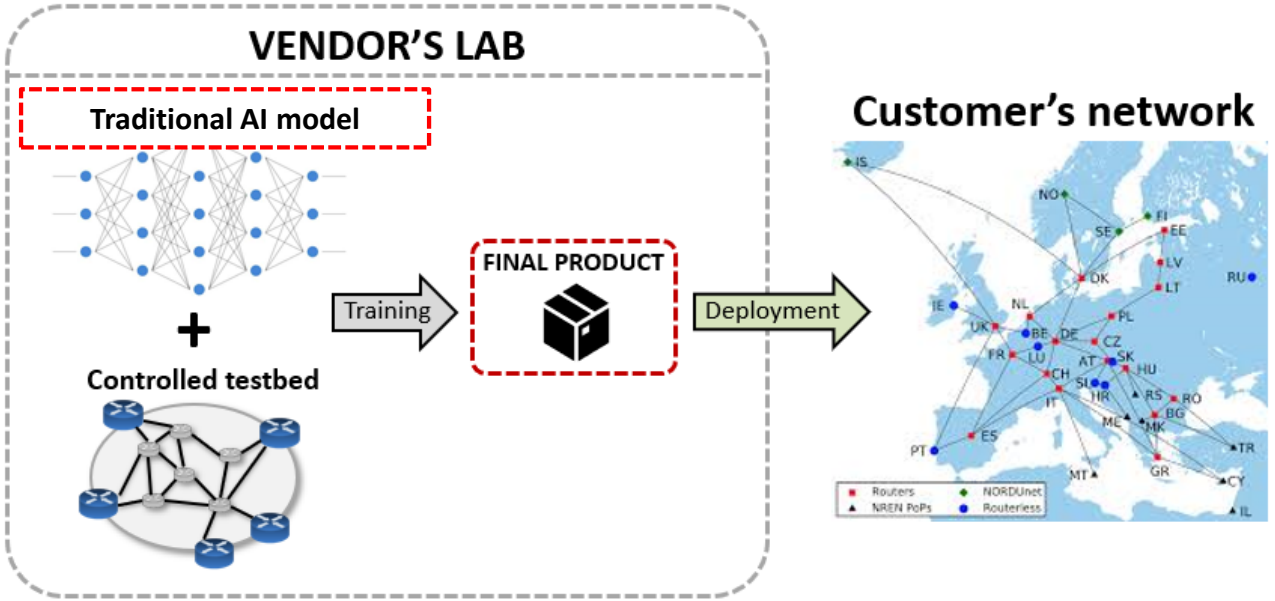
Deployment

Customer's network



Commercialization problem of traditional AI solutions for networks

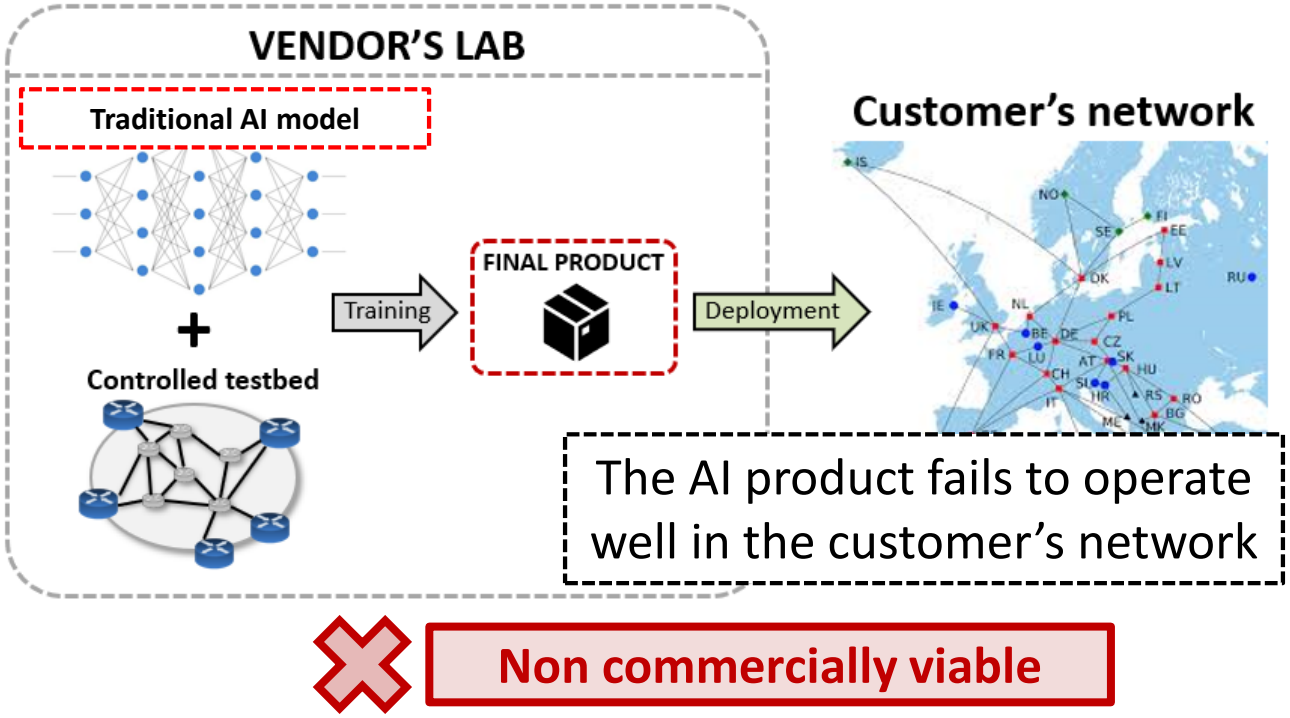
- Main barrier to achieve commercializable AI products:
  - **Traditional AI-based solutions do not generalize to other networks**



Commercialization problem of traditional AI solutions for networks

- Main barrier to achieve commercializable AI products:
  - **Traditional AI-based solutions do not generalize to other networks**
  - It is unfeasible to train AI tools for networking on the **customer's network**:

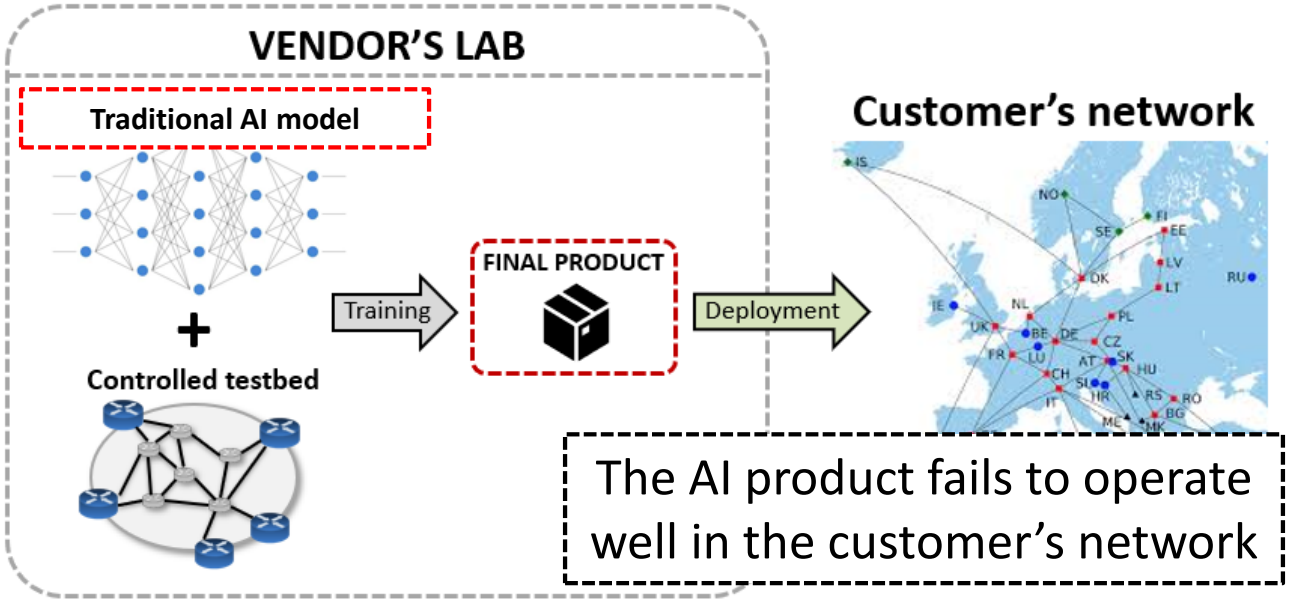
    It would require network instrumentation and may cause service disruption due to possible wrong configurations!
  - Difficult to replicate the customer's network in a networking lab to train the AI product



Commercialization problem of traditional AI solutions for networks

- Main barrier to achieve commercializable AI products:
  - **Traditional AI-based solutions do not generalize to other networks**
  - It is unfeasible to train AI tools for networking on the **customer's network**:

    It would require network instrumentation and may cause service disruption due to possible wrong configurations!
  - Difficult to replicate the customer's network in a networking lab to train the AI product



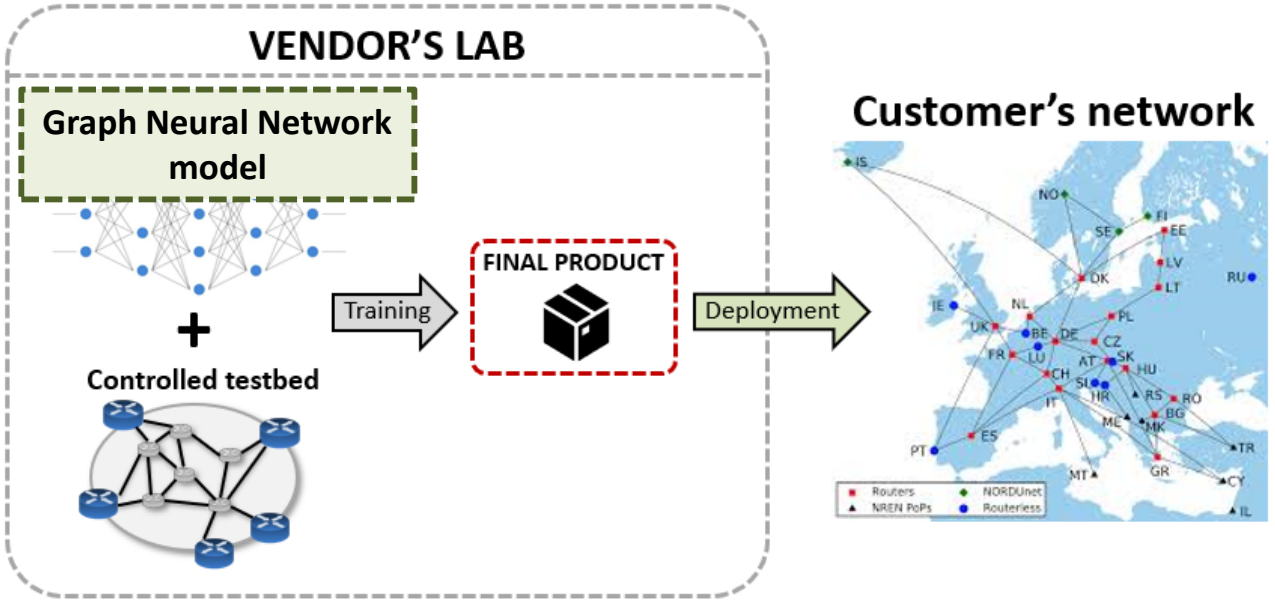
Non commercially viable

Need for AI models able to generalize to other networks unseen during training

GNN as a commercializable AI solution for networking

So far, **Graph Neural Networks (GNN) are the only AI-based models that can generalize to other networks** (not seen in advance):

- Topology
- Routing configuration
- Traffic
- …



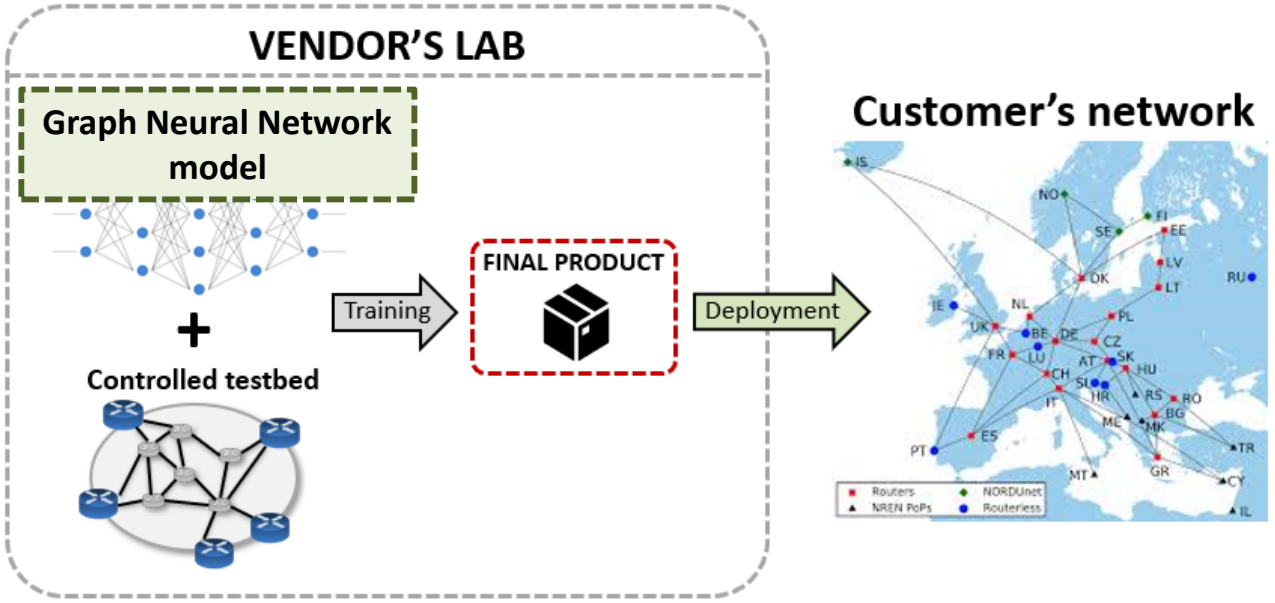
GNN as a commercializable AI solution for networking

So far, **Graph Neural Networks (GNN) are the only AI-based models that can generalize to other networks** (not seen in advance):

- Topology
- Routing configuration
- Traffic
- …

Vendor's lab:

- Offline training on controlled testbeds with synthetic topologies and configurations



GNN as a commercializable AI solution for networking

So far, **Graph Neural Networks (GNN) are the only AI-based models that can generalize to other networks** (not seen in advance):

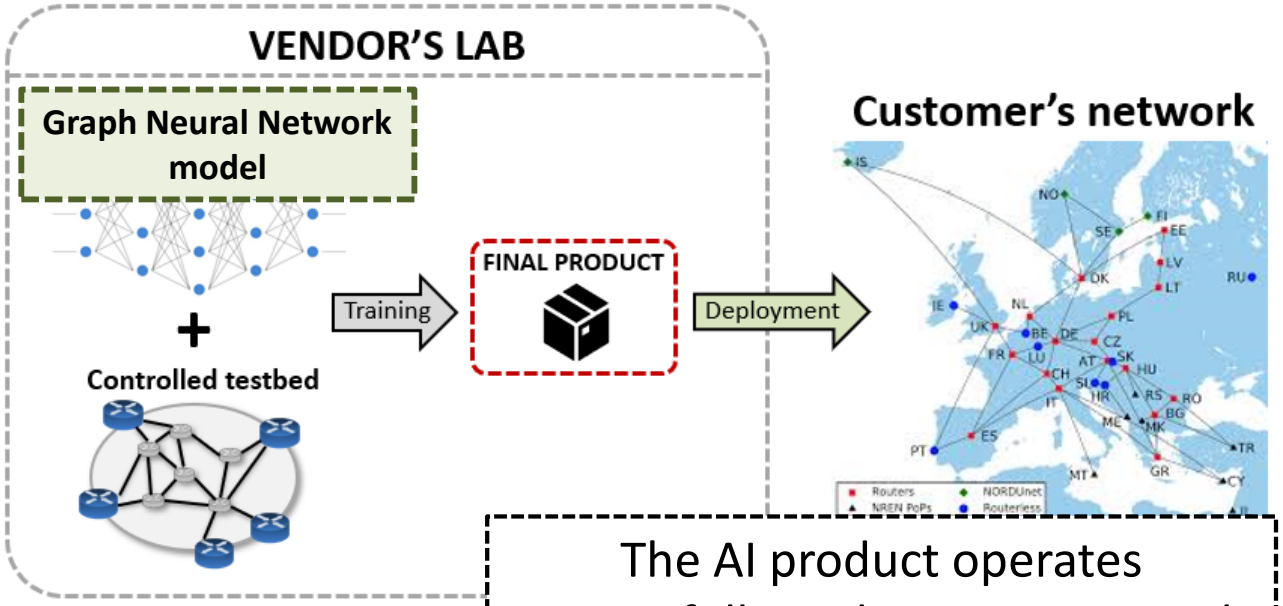
- Topology
- Routing configuration
- Traffic
- …

Vendor's lab:

- Offline training on controlled testbeds with synthetic topologies and configurations

Deployment on the customer's network:

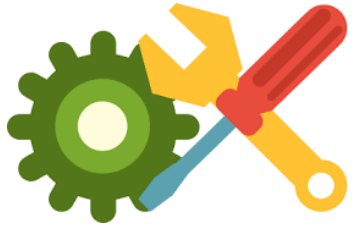
- **One final product** that can operate on any customer network



GNN enables commercially viable AI solutions

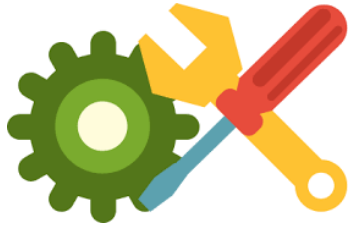
Design of GNN-based solutions for networking

- Necessity of **custom GNN designs** for different networking use cases:
  - QoS-aware configuration optimization (e.g., routing)
  - Optical Networks (e.g., routing, modulation, spectrum assignment)
  - VNF placement
  - …

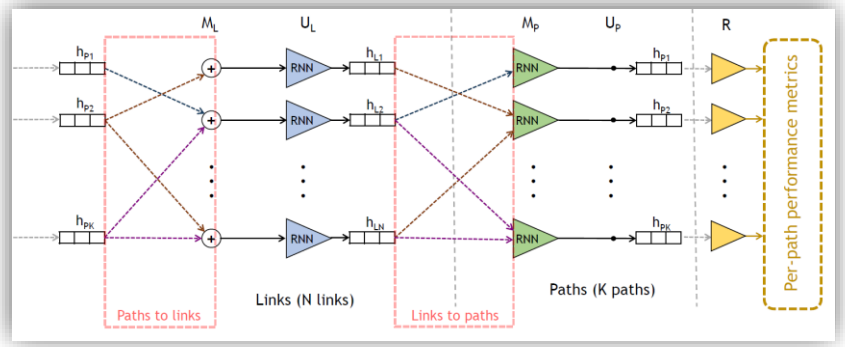


Design of GNN-based solutions for networking

- Necessity of **custom GNN designs** for different networking use cases:
  - QoS-aware configuration optimization (e.g., routing)
  - Optical Networks (e.g., routing, modulation, spectrum assignment)
  - VNF placement
  - …



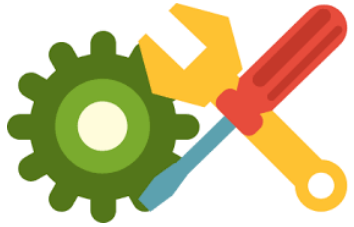
- Each use case requires a mathematical formulation to represent the different network elements involved in the form of graphs:
  - E.g., topology, routing, traffic, security policy…



- Need of ML experts with high skills on neural network programming (e.g. TensorFlow)

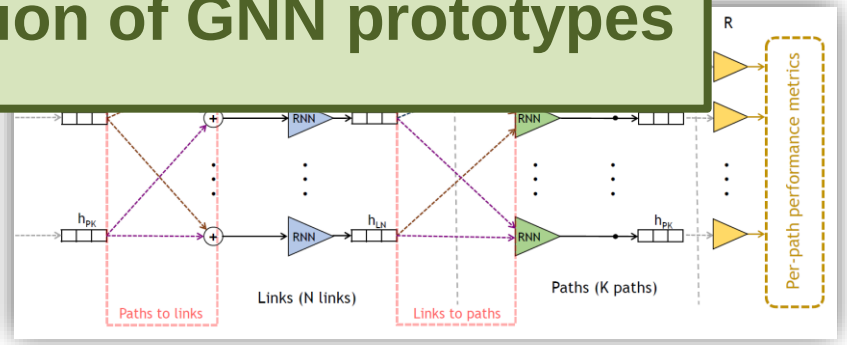
Design of GNN-based solutions for networking

- Necessity of **custom GNN designs** for different networking use cases:
  - QoS-aware configuration optimization (e.g., routing)
  - Optical Networks (e.g., routing, modulation, spectrum assignment)
  - VNF placement
  - …



- Ea work
  ele
  -

Motivation: **To boost the adoption of GNN for networks it is essential to simplify the implementation of GNN prototypes**



- Need of ML experts with high skills on neural network programming (e.g. TensorFlow)

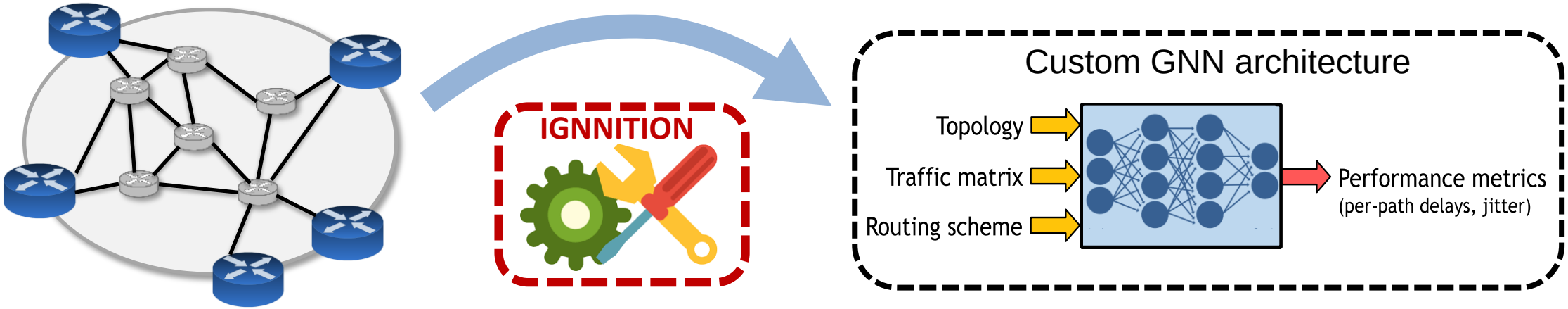
Barcelona Neural Networking Center

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Departament d'Arquitectura de Computadors

IGNNITION: A framework for fast prototyping of GNN models for network AI

What is IGNNITION?

- Generic framework for GNN applied to networking



IGNNITION is an **easy-to-use GNN toolbox** for **networking** researchers/practitioners

Why this framework?

- Current situation:
  - Programs in Tensor-based languages

TensorFlow

PyTorch

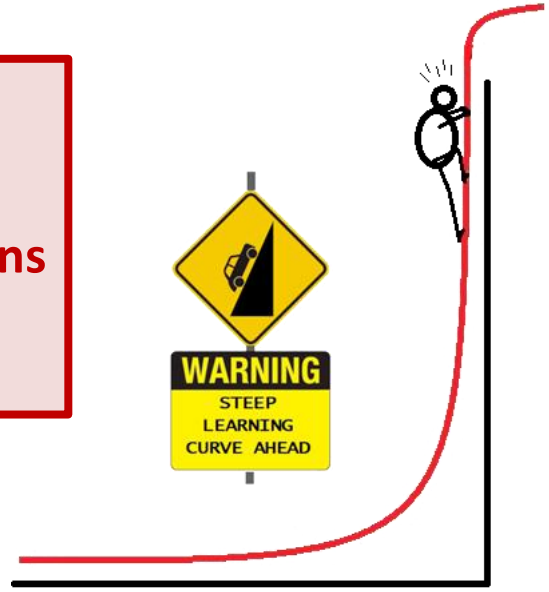
Huawei Mindspore

```
h_tild = tf.gather(link_state,links)

ids=tf.stack([paths, seqs], axis=1)
max_len = tf.reduce_max(seqs)+1
shape = tf.stack([f_['n_paths'], max_len, self.hparams.link_state_dim])
lens = tf.math.segment_sum(data=tf.ones_like(paths),
                           segment_ids=paths)

link_inputs = tf.scatter_nd(ids, h_tild, shape)
outputs, path_state = tf.nn.dynamic_rnn(self.path_update,
                                        link_inputs,
                                        sequence_length=lens,
                                        initial_state = path_state,
                                        dtype=tf.float32)
                                  ⋮
```

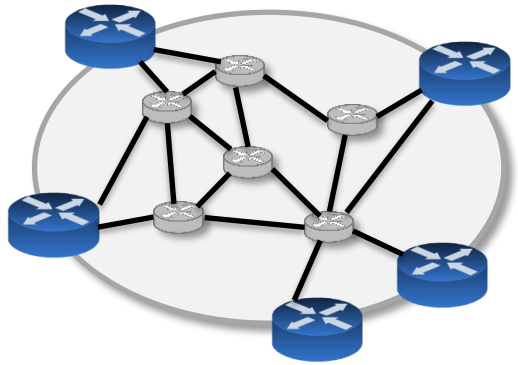
Main difficulties:

1) Convert GNN designs into complex tensor-wise operations

2) Very complex to debug!



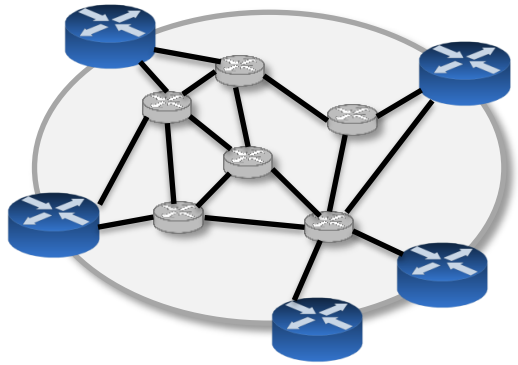
How it works?



Networking use case:

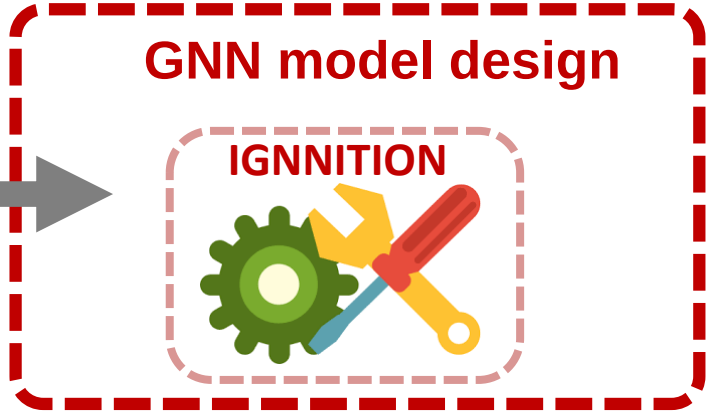
- Routing optimization
- Security policy
- Network Function Virtualization
- Target metrics (e.g., performance, anomaly detection)
- . . .

How it works?

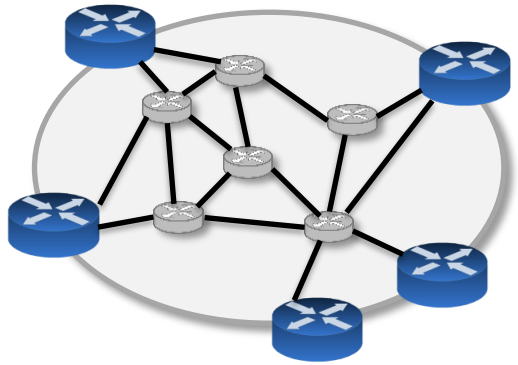


Networking use case:

- Routing optimization
- Security policy
- Network Function Virtualization
- Target metrics (e.g., performance, anomaly detection)
- . . .



How it works?

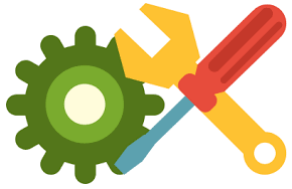


Networking use case:

- Routing optimization
- Security policy
- Network Function Virtualization
- Target metrics (e.g., performance, anomaly detection)
- . . .

GNN model design

IGNNITION



Automatic code generation

GNN model implementation

```
for _ in range(self.hparams.T):

    h_tild = tf.gather(link_state,links)

    ids=tf.stack([paths, seqs], axis=1)
    max_len = tf.reduce_max(seqs)+1
    shape = tf.stack([f_['n_paths'], max_len, self.hparams.link_state_dim])
    lens = tf.math.segment_sum(data=tf.ones_like(paths),
                               segment_ids=paths)

    link_inputs = tf.scatter_nd(ids, h_tild, shape)
    outputs, path_state = tf.nn.dynamic_rnn(self.path_update,
                                            link_inputs,
                                            sequence_length=lens,
                                            initial_state = path_state,
                                            dtype=tf.float32)
    . . .
```

User's workflow

Develop you own GNN solution in three simple steps

Step 1: GNN model description

Network elements

```
"entities": [
    {
        "name": "link",
        "hidden_state_dimension": 32,
        "features":[
            {
                "name": "link_capacity",
                "size": 1
            }
        ]
    },
    {
        "name": "path",
        "hidden_state_dimension": 32,
        "features":[
            {
                "name": "traffic",
                "size": 1
            }
        ]
    }
],
```

Relations between elements

```
[
    {
    "step1":[
        {
            "type": "individual",
            "source_entity": "link",
            "destination_entity":"path",
            "agregation": "ordered",
            "update": "recurrent",
            "adj_vector": "adj_links_paths"
        }]
    },
    {
    "step2":[
        {
            "type": "individual",
            "source_entity": "path",
            "destination_entity":"link",
            "agregation": "sum",
            "update": "recurrent",
            "adj_vector": "adj_paths_links"
        }]
    }
]
```

The GNN model is defined via a JSON file

Network abstraction:
Descriptive representation of the network elements involved in the networking use case

We provide a template and documentation to fill the JSON file

Step 2: Migrate your dataset to JSON

- Standard JSON interface to easily feed the GNN model with any dataset

Step 3: Training/evaluation execution

- Execution with just few lines of code
  - Training:

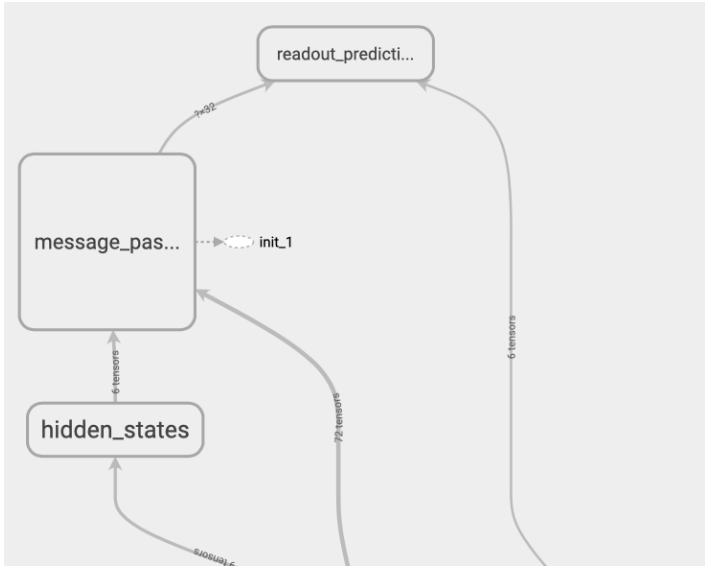
```
gnn_model = load_model(json_file)
train_and_evaluate(gnn_model, path_to_dataset, training_params_file)
```

  - Execute a trained model:

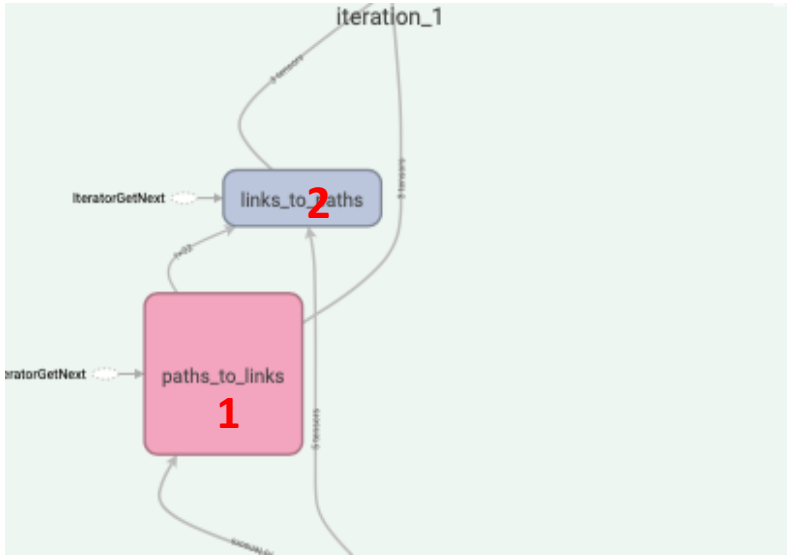
```
gnn_model = load_trained_model(path_to_model)
predict(gnn_model, dataset)
```

Easy to debug

- Visualization of GNN models automatically generated (interactive graph):



General overview of RouteNet*



One message-passing iteration in RouteNet*

The framework identifies potential errors and assists users to correct them

*K. Rusek, J. Suárez-Varela, A. Mestres, P. Barlet-Ros, A. Cabellos-Aparicio A, "Unveiling the potential of Graph Neural Networks for network modeling and optimization in SDN," In Proceedings of ACM SOSR, pp. 140-151, 2019.

Main advantages

- **Fast GNN prototyping** for networking practitioners/researchers:

  Months to create a GNN prototype

  Few hours for a GNN prototype

- It provides support to design **GNN protypes for any networking use case**
- **Easy debugging**

Bridge the gap between the networking and AI communities

Main advantages

- Public repository with state-of-the-art GNN models applied to networks:
  - Standard Message-passing on networks
  - RouteNet*
  - …

Progress and education in AI cannot occur without public implementations and datasets

- Open source software (v0.1)

https://github.com/knowledgedefinednetworking/ignnition

*K. Rusek, J. Suárez-Varela, A. Mestres, P. Barlet-Ros, A. Cabellos-Aparicio, “Unveiling the potential of Graph Neural Networks for network modeling and optimization in SDN,” In Proc. of ACM SOSR, 2019.

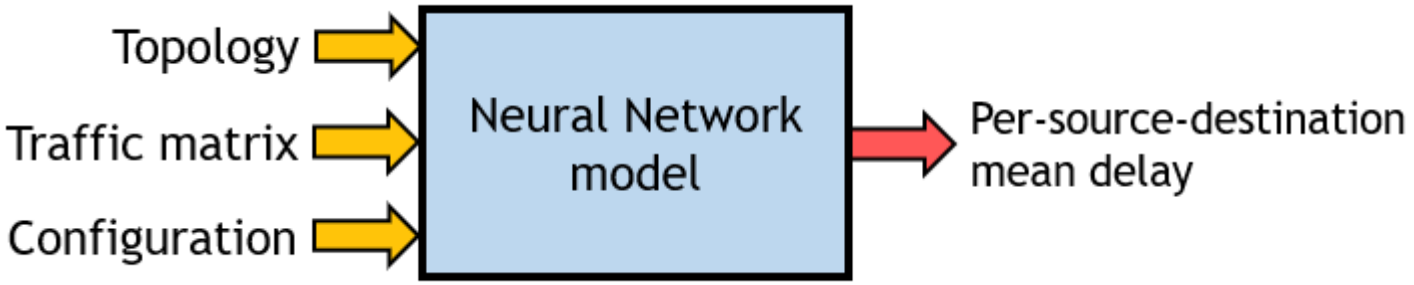
Barcelona Neural Networking Center

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Departament d'Arquitectura de Computadors

Updates on the…

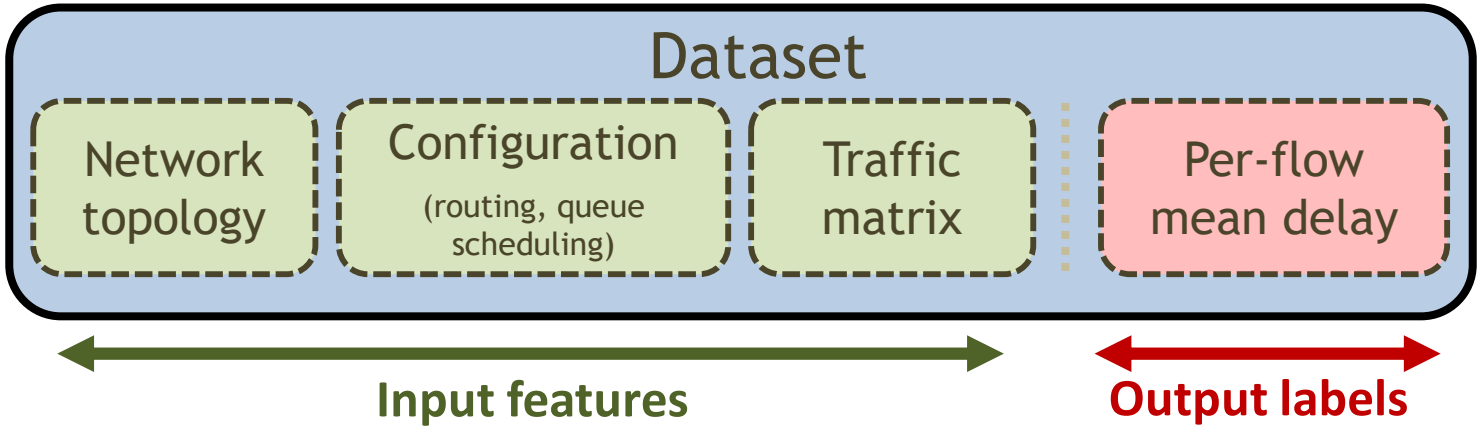
https://bnn.upc.edu/challenge2020

Problem overview:



- Inputs:
  - Network topology
  - Source-destination traffic matrix
  - Network configuration:
    - Routing
    - Queue scheduling policy (e.g., Weighted Fair Queueing, Deficit Round Robin)
- Output:
  - Mean per-packet delay on each source-destination path

Datasets:



- Simulated with OMNet++
- Several topologies, hundreds of combinations of routing + queue scheduling + traffic
- Three different datasets:
  - Training and evaluation → Include output labels
  - Test set → Unlabeled (used to evaluate proposed solutions)

Evaluation:

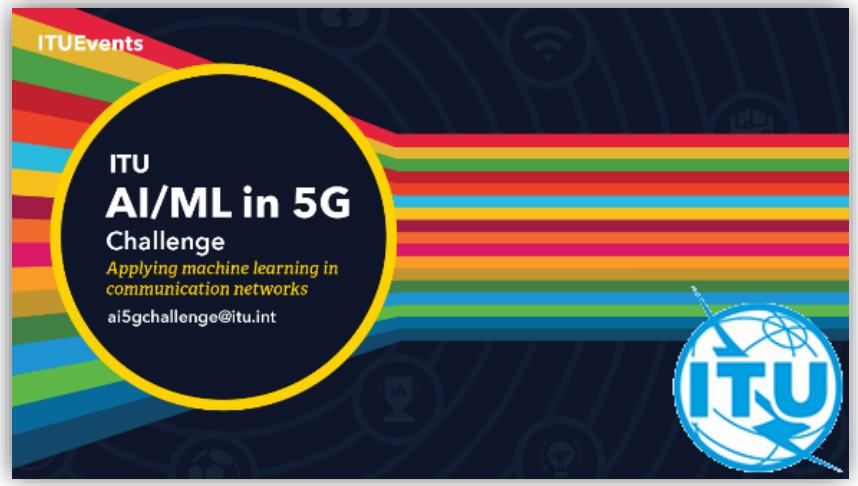
- **Test the generalization capabilities of neural network solutions:**
  - Training dataset → Samples simulated in two network topologies
  - Validation and Test datasets → Samples simulated in a third topology

- We will test the capability of the proposed solutions to make good delay predictions in the third network (unseen during training)

Graph Neural Networking challenge 2020

- **Target audience:**
  - Networking community
  - AI community (GNN is a hot topic!)

- **Resources:**
  - Baseline model and tutorial → RouteNet*
  - API to easily read and process the datasets
  - Mailing list to engage participants



Check a list with all the challenges (e.g., China telecom, China Unicom, Vodafone, Lenovo, ZTE, etc) at: https://www.itu.int/en/ITU-T/AI/challenge/2020/Pages/default.aspx

- **Organized as part of the ITU AI/ML in 5G Challenge**

*K. Rusek, J. Suárez-Varela, A. Mestres, P. Barlet-Ros, A. Cabellos-Aparicio, "Unveiling the potential of Graph Neural Networks for network modeling and optimization in SDN," In Proc. of ACM SOSR, 2019.

Graph Neural Networking challenge 2020

See all the details at:

https://bnn.upc.edu/challenge2020

- **Open to all participants around the world!** (teams up to 4 members)
- **Timeline** → May 22nd-Nov 15th (≈6-month duration)
- **Final conference and awards** → Nov-Dec 2020

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Departament d'Arquitectura de Computadors

Thank you for your attention!

José Suárez-Varela (jsuarezv@ac.upc.edu)

David Pujol Perich

Miquel Ferriol Galmés

Albert López-Brescó

Pere Barlet Ros

Albert Cabellos Aparicio