IGNNITION: A framework for fast prototyping of Graph Neural Networks

David Pujol-Perich[1], José Suárez-Varela[1] , Miquel Ferriol-Galmés[1], Shihan Xiao[2], Bo Wu[2], Albert Cabellos-Aparicio[1], Pere Barlet-Ros[1]

1. Motivation

❑ **Graph Neural Networks (GNN):**

- GNNs have outstanding applications in many fields where **data is structured as graphs** (e.g., computer networks, chemistry, biology, physics, recommender systems) [1, 2].
- Nowadays, **GNN is a hot topic** in the Machine Learning (ML) field, and new applications are continuously emerging.
- **GNNs are particularly difficult to implement** (even for experts in neural network programming).

❑ **Main barriers for exploring new applications:**

1) **Experts in the potential fields of application often lack the needed programming skills** (e.g., TensorFlow, PyTorch).

2) GNN-based applications **often require to create custom GNN architectures adapted to the problem.** (e.g., non-standard message-passing schemes)

2. Existing GNN programming frameworks

❑ Frameworks with support for GNN face a complex trade-off between **flexibility** and **usability**.

❑ Two main categories can be distinguished:

1. **Frameworks with full flexibility:** (e.g., DGL, PyTorch Geometric, DeepMind's Graph Nets)
   - ✓ Make no assumptions on the possible GNN architectures.
   - X Require tensor-based implementations of critical parts of the GNN (e.g., NN layers, message function).
2. **Frameworks for quick implementation:** (e.g., Spektral, Graph Gym)
   - X Introduce important limitations on the possible GNN architectures (e.g., they mainly support well-known GNN models).
   - ✓ Provide high-level GNN abstractions (or even codeless programming interfaces)

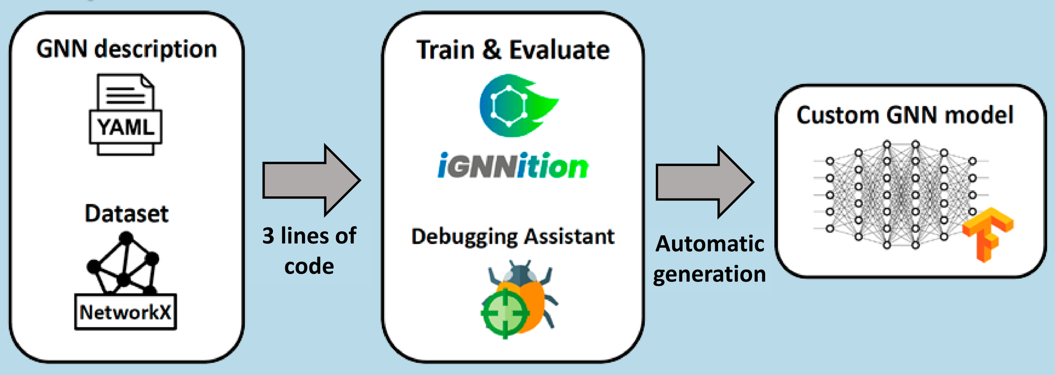
3. IGNNITION

❑ Main features:

1. **High-level abstraction:** Novel message-passing abstraction (MSMP graph).
2. **Codeless interface:** Users can define their GNNs in a YAML file.
3. **Flexibility:** Support for standard models (e.g., GCN, GAT, MPNN, Gated Neural Networks) [3] and **custom GNNs** with non-standard message-passing schemes (e.g., GNN models applied to computer networks [1, 4]).
4. **Debugging assistant:** Visual representations and advanced error-checking.
5. **High performance:** Equivalent to native TensorFlow implementations.

More information at: https://ignnition.net

❑ IGNNITION produces an efficient **TensorFlow implementation** given a GNN model description (in YAML) and a dataset processed with NetworkX.



Use case: Implementation of RouteNet [1]

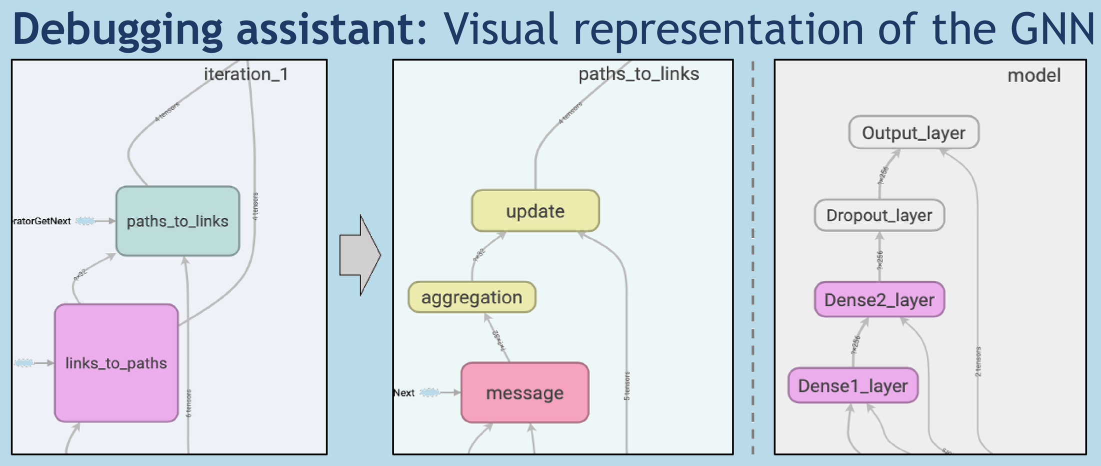
Fragment of the YAML file: Message passing definition

```yaml
- destination_entity: link
  source_entities:
    - name: path
      message:
          - type: direct_assignment

  aggregation:
        - type: sum

  update:
        type: neural_network
        nn_name: recurrent_1
```

Debugging assistant: Visual representation of the GNN



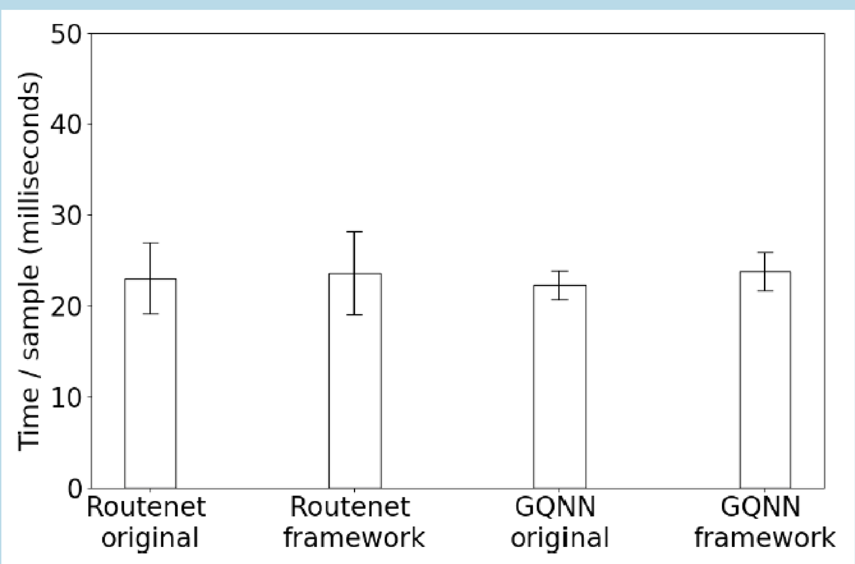
4. Evaluation

❑ We implement two models applied to computer networks with non-standard message-passing architectures: RouteNet [1], and GQNN [4].

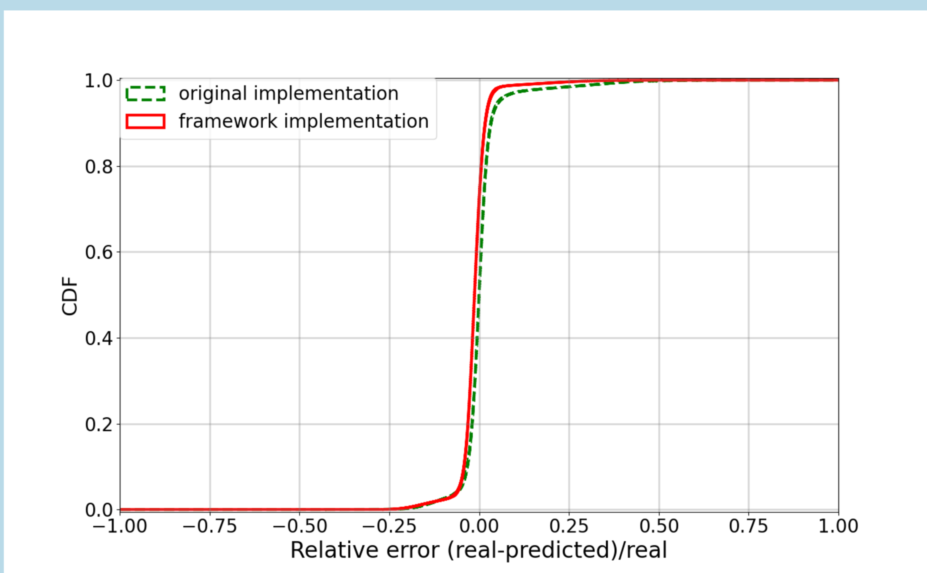
❑ Compare the performance of our implementation w.r.t. the original implementations in TensorFlow.

❑ **Experimental evaluation (IGNNITION vs Native TensorFlow):**

Execution cost (during training)



Accuracy (Mean Relative Error)



Results: The resulting IGNNITION implementations are equivalent in execution cost and accuracy (after training) to the original implementations directly coded in TensorFlow.

Take-home messages:

- **IGNNITION provides a codeless interface for fast prototyping of GNNs**, being completely oblivious to the underlying complex tensor-based operations.
- **Flexibility to implement custom GNNs** with non-standard message-passing schemes.
- **No performance loss compared to native TensorFlow implementations.**

References

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

[2] Jie Zhou, et al., 2018. Graph neural networks: A review of methods and applications. arXiv preprint arXiv:1812.08434.

[3] Wu, Zonghan, et al. "A comprehensive survey on graph neural networks." *IEEE transactions on neural networks and learning systems* (2020).

[4] Fabien Geyer et. al. 2018. Learning and generating distributed routing protocols using graph-based deep learning. In ACM BigDAMA workshop.

Acknowledgment

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).

[1] Barcelona Neural Networking Center, Universitat Politècnica de Catalunya, Spain
[2] Network Technology Lab., Huawei Technologies Co.,Ltd.

CONTACT US: contactus@bnn.upc.edu

Barcelona Neural Networking Center