Half-Hop: A Graph Upsampling Approach for Slowing Down Message Passing

Mehdi Azabou[1], Venkataramana Ganesh[1], Shantanu Thakoor[2], Chi-Heng Lin[1], Lakshmi Sathidevi[1], Ran Liu[1],
Michal Valko[2], Petar Veličković[2], Eva L. Dyer[1]

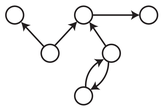
[1] Georgia Tech
[2] Google DeepMind

tl;dr

- ➔ This work introduces Half-Hop, a **plug-and-play** graph augmentation for message passing neural networks.
- ➔ Half-Hop works well across a **wide range of applications** and across diverse sets of graph benchmarks, with impressive boosts in supervised and self-supervised settings.
- ➔ Half-Hop significantly improves the performance of MPNNs on **heterophilic graphs**.
- ➔ In our analysis of the dynamics of message passing, we find that Half-Hop helps to **avoid oversmoothing** and **improve generalization**.

Motivation

Graph augmentations are key in the graph learning pipeline, and are used to:
- improve the generalization of the model by generating new samples.
- improve the connectivity and information flow in a graph.



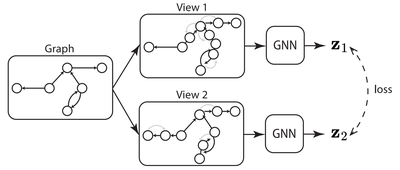
$$h_i^{(l)} = \sigma\left(\sum_{j \in \mathcal{N}(i)} \mathbf{W}_l h_j^{(l-1)}\right)$$

Message Passing Update

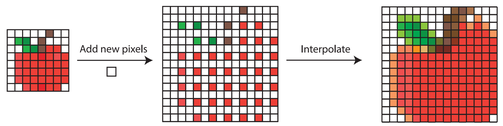
$h_i^{(0)} = x_i$: Node features
$\mathbf{W}_l$: Learnable weights

Graph augmentations are also **critical** in the self-supervised learning pipeline:

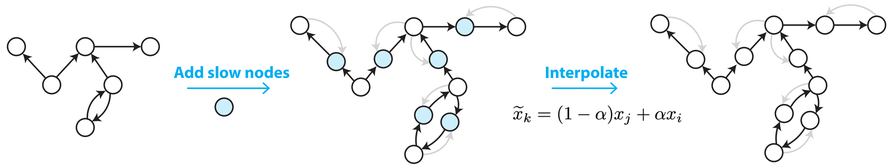
- Augmentations are used to perturb the graph, mainly by removing edges or masking node features.
- **Finding augmentations** for graphs that work well in diverse settings is difficult.



Can upsampling improve message passing?



Proposed Method



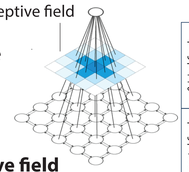
Half-Hop is **plug-and-play**, and works with a wide range of datasets, architectures, and learning objectives!

When we apply it **probabilistically**, Half-Hop introduces variable path lengths between nodes

Code

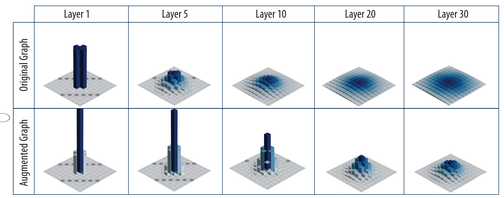
```
#apply augmentation
transform = HalfHop(alpha=0.5)
data = transform(data)

#feedforward
y = model(data)

#get rid of slow nodes
y = y[~data.slow_node_mask]
```

Half-Hop Improves Message Passing

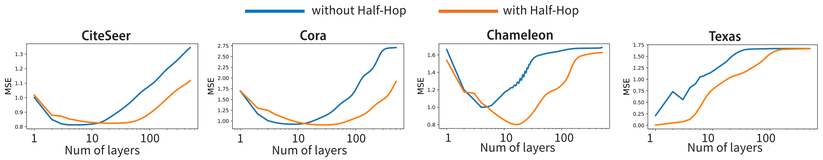


Experiment. We analyse the message passing dynamics for a simple linear GNN.

Half-Hop **alters the receptive field** and allows for **deeper models**



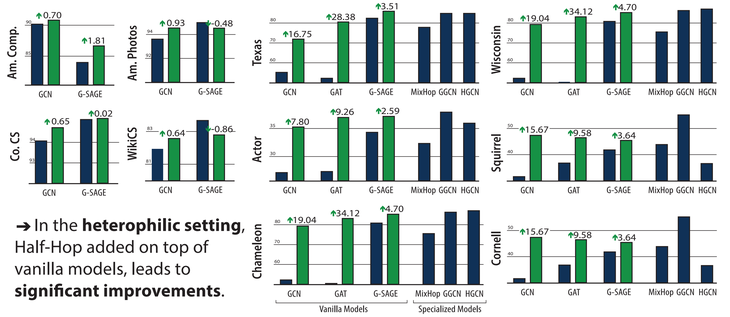
Experiment. We report the test mean squared error (MSE) for a node regression task as a function of the number of message passing steps, when using a linear GNN.



➔ **More rounds of message passing** can be performed before we observe over-smoothing.
➔ The basin of low error solutions is widened, and in some cases, Half-Hop **improves generalization**.

Results: Supervised

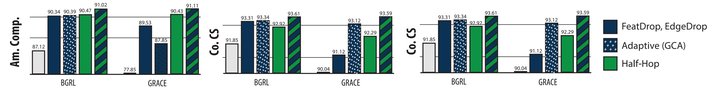
Half-Hop boosts the performance of different models, across various benchmarks.



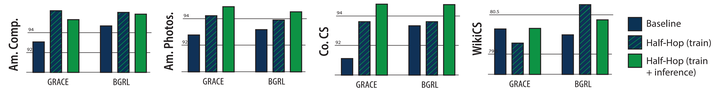
➔ In the **heterophilic setting**, Half-Hop added on top of vanilla models, leads to **significant improvements**.

Results: Self-Supervised

When used as an **augmentation**, Half-Hop creates views where the number of hops between nodes varies.

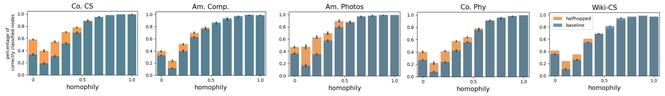


Half-Hop can also be used at **inference time**!



➔ Half-Hop can be used as the sole augmentation, or combined with existing augmentations to unlock new **state-of-the-art performance** on self-supervised learning benchmarks.

Where are the benefits of Half-Hop coming from?



➔ We find that improvements are most significant for **heterophilic nodes**.