Lipschitz Normalization for Self-Attention Layers with Application to Graph Neural Networks

George Dasoulas Kevin Scaman Aladin Virmaux

Attention

- A *soft-selection* method.
- *Partial* focus of structured input.

Attention mechanism

Let $X \in \mathbb{R}^{d \times n}$ be an input matrix and a **score function** $g : \mathbb{R}^{d \times n} \to \mathbb{R}^{m \times n}$:

$$\mathrm{Att}(X) = h(X)\,\mathrm{softmax}(g(X))^\top .$$

Standard attention

$\mathrm{Att}(X) = X\,\mathrm{softmax}(Q^\top X)^\top .$

Transformer

$\mathrm{Att}(X) = V\mathrm{softmax}\left(\frac{Q^\top K}{\sqrt{d}}\right)^\top$ with $X = (Q||K||V)$

Attention

- A *soft-selection* method.
- *Partial* focus of structured input.

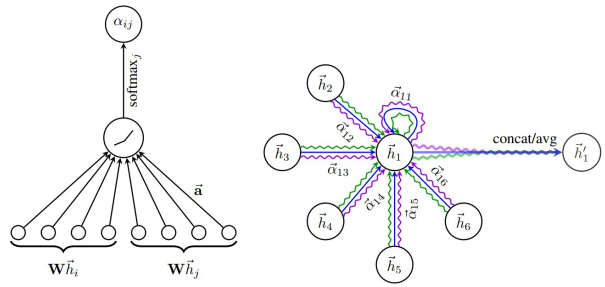
Attention mechanism

Let $X \in \mathbb{R}^{d \times n}$ be an input matrix and a **score function** $g : \mathbb{R}^{d \times n} \to \mathbb{R}^{m \times n}$:

$$\mathrm{Att}(X) = h(X)\,\mathrm{softmax}(g(X))^\top .$$

Graph Learning and Attention

- **Neighbor-wise** softmax function.



Graph attention network (Veličković et al, 2018)

Standard attention

$$\mathrm{Att}(X) = X\,\mathrm{softmax}(Q^\top X)^\top .$$

Transformer

$$\mathrm{Att}(X) = V\mathrm{softmax}\left(\frac{Q^\top K}{\sqrt{d}}\right)^\top \text{ with}$$

$X = (Q||K||V)$

Attention

- A *soft-selection* method.
- *Partial* focus of structured input.

Attention mechanism

Let $X \in \mathbb{R}^{d \times n}$ be an input matrix and a **score function** $g : \mathbb{R}^{d \times n} \to \mathbb{R}^{m \times n}$:

$$\mathrm{Att}(X) = h(X)\,\mathrm{softmax}(g(X))^\top .$$

Standard attention

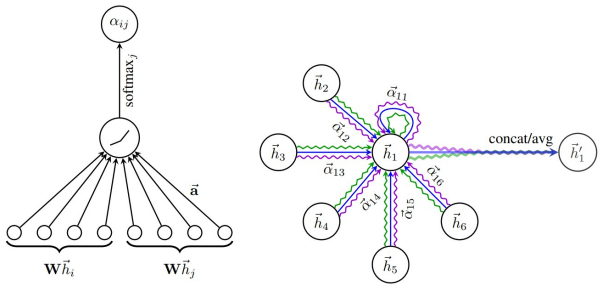
$\mathrm{Att}(X) = X\,\mathrm{softmax}(Q^\top X)^\top$.

Transformer

$\mathrm{Att}(X) = V\mathrm{softmax}\left(\frac{Q^\top K}{\sqrt{d}}\right)^\top$ with $X = (Q||K||V)$

Graph Learning and Attention

- **Neighbor-wise** softmax function.



Graph attention network (Veličković et al, 2018)

GNN training and model depth

- **Deep** attention models suffer from convergence inability (Alon et al, 2020).
  - Oversmoothing, oversquashing.
  - Gradient explosion/vanishing not explored!
- Gradient flow and Lipschitz continuity.

Overall message and contributions

How to enforce Lipschitz continuity to attention layers?

Main contributions

1. We derive **general bounds** for the Lipschitz constant of attention layers.
2. We propose a **novel normalization** for attention layers that ensures Lipschitz continuity.
3. We apply this normalization to **graph attention networks**.

Is attention Lipschitz continuous ?

- Large scores tend to create large gradients!

Lemma

For any $X \in \mathbb{R}^{d \times n}$, the norm of the derivative of attention is upper bounded by:

$$|||\mathrm{D}\mathrm{Att}_X|||_F \leqslant \|\mathrm{softmax}(g(X))\|_F + \sqrt{2}\|X^\top\|_{(\infty,2)}|||\mathrm{D}g_X|||_{F,(2,\infty)} \,.$$

Is attention Lipschitz continuous ?

- Large scores tend to create large gradients!

Lemma

For any $X \in \mathbb{R}^{d\times n}$, the norm of the derivative of attention is upper bounded by:

$$\|\|\mathrm{D}\mathrm{Att}_X\|\|_F \leqslant \|\mathrm{softmax}(g(X))\|_F + \sqrt{2}\|X^\top\|_{(\infty,2)}\|\|\mathrm{D}g_X\|\|_{F,(2,\infty)} \cdot$$

Impact of a scalar normalization:

Normalize the score function by a scalar function $c : \mathbb{R}^{d\times n} \to \mathbb{R}_+$: $g(X) = \frac{\tilde{g}(X)}{c(X)}$

Is attention Lipschitz continuous ?

- Large scores tend to create large gradients!

Lemma

For any $X \in \mathbb{R}^{d\times n}$, the norm of the derivative of attention is upper bounded by:

$$|||\mathrm{D}\mathrm{Att}_X|||_F \leqslant \|\mathrm{softmax}(g(X))\|_F + \sqrt{2}\|X^\top\|_{(\infty,2)}|||\mathrm{D}g_X|||_{F,(2,\infty)}\,.$$

Impact of a scalar normalization:

Normalize the score function by a scalar function $c : \mathbb{R}^{d\times n} \to \mathbb{R}_+$: $g(X) = \frac{\tilde{g}(X)}{c(X)}$

Theorem

Let $\alpha \geqslant 0$. If, for all $X \in \mathbb{R}^{d\times n}$, we have

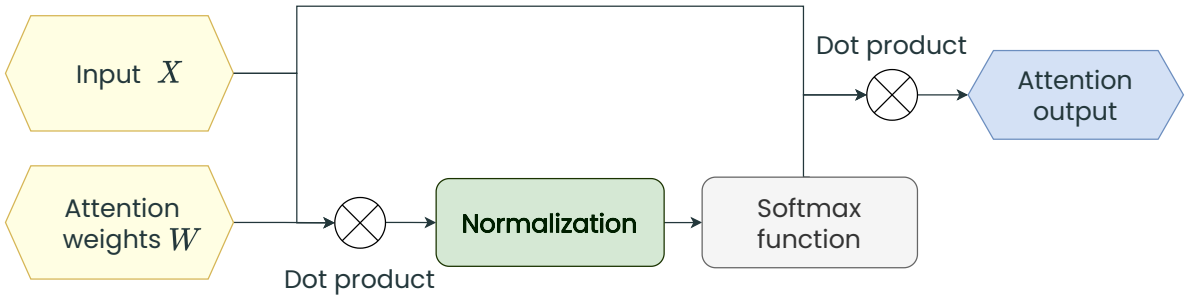
(1) $\|\tilde{g}(X)\|_\infty \leqslant \alpha c(X)$,
(2) $\|X^\top\|_{(\infty,2)}|||\mathrm{D}\tilde{g}_X|||_{F,(2,\infty)} \leqslant \alpha c(X)$,
(3) $\|X^\top\|_{(\infty,2)}|||\mathrm{D}c_X|||_{F,1}\|\tilde{g}(X)\|_{(2,\infty)} \leqslant \alpha c(X)^2$,

then attention models with $g(X) = \tilde{g}(X)/c(X)$ are Lipschitz continuous and

$$L_F(\mathrm{Att}) \leqslant e^\alpha\sqrt{\frac{m}{n}} + \alpha\sqrt{8}\,.$$

The *LipschitzNorm* normalization

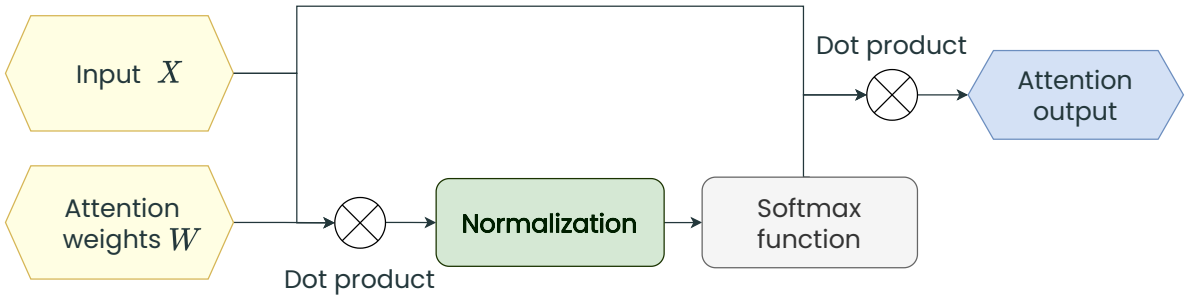
Core Idea: Replace original $\tilde{g}(X)$ with normalized $g(X)$ in the attention model!



Pipeline of LipschitzNorm.

The *LipschitzNorm* normalization

Core Idea: Replace original $\tilde{g}(X)$ with normalized $g(X)$ in the attention model!



Pipeline of LipschitzNorm.

Linear score function $g(X)$

$$g(X) = \frac{Q^\top X}{\|Q\|_F \|X^\top\|_{(\infty,2)}}$$

$L_F(\mathrm{Att}) \leqslant e^1 \sqrt{\frac{m}{n}} + \sqrt{8}.$

Quadratic score function $g(X)$

$$g(X) = \frac{Q^\top K}{\max\{uv, uw, vw\}},$$

where $u = \|Q\|_F, v = \|K^\top\|_{(\infty,2)}, w = \|V^\top\|_{(\infty,2)}$

$L_F(\mathrm{Att}) \leqslant e^{\sqrt{3}} \sqrt{\frac{m}{n}} + 2\sqrt{6}.$

Gradient Explosion

For **deep attention** models, there is a tight connection between: efficient training and Lipschitz continuity.

Lipschitz constant

Given M attention layers $\mathrm{Att}_m(\cdot)$, $f = \mathrm{Att}_1 \circ \mathrm{Att}_2 \circ \ldots \circ \mathrm{Att}_{m-1} \circ \mathrm{Att}_m$ is Lipschitz continuous:

$$L_F(f) \leqslant \prod_{m=1}^{M} L_F(\mathrm{Att}_m).$$

- Multiplicative effect on the gradient flow.
- Enforcing Lipschitz continuity can alleviate gradient explosion.

Gradient Explosion

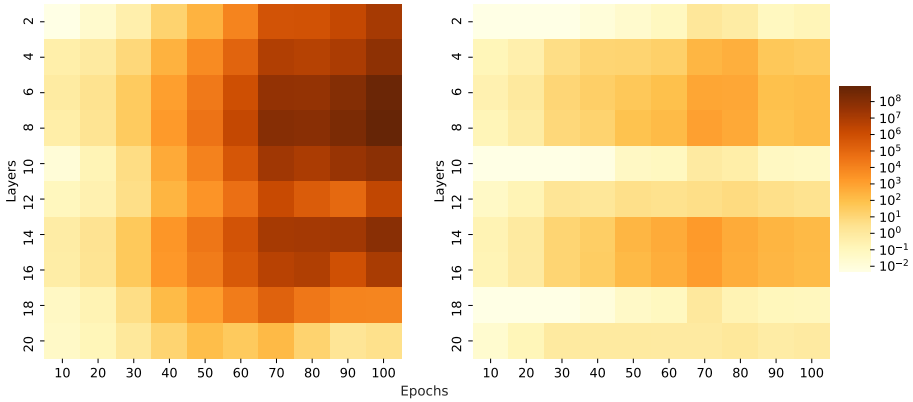
For **deep attention** models, there is a tight connection between: efficient training and Lipschitz continuity.

Lipschitz constant

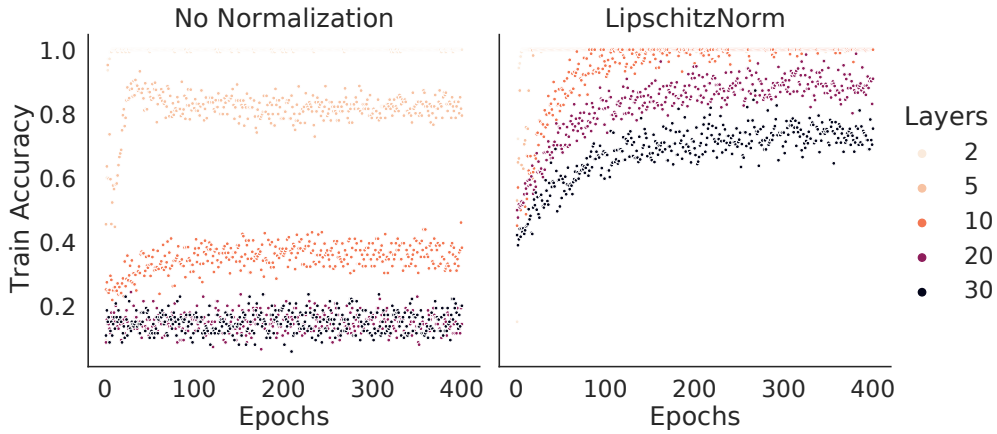
Given M attention layers $\mathrm{Att}_m(\cdot)$, $f = \mathrm{Att}_1 \circ \mathrm{Att}_2 \circ \ldots \circ \mathrm{Att}_{m-1} \circ \mathrm{Att}_m$ is Lipschitz continuous:

$$L_F(f) \leqslant \prod_{m=1}^{M} L_F(\mathrm{Att}_m).$$

- **Multiplicative effect** on the gradient flow.
- Enforcing Lipschitz continuity can alleviate gradient explosion.



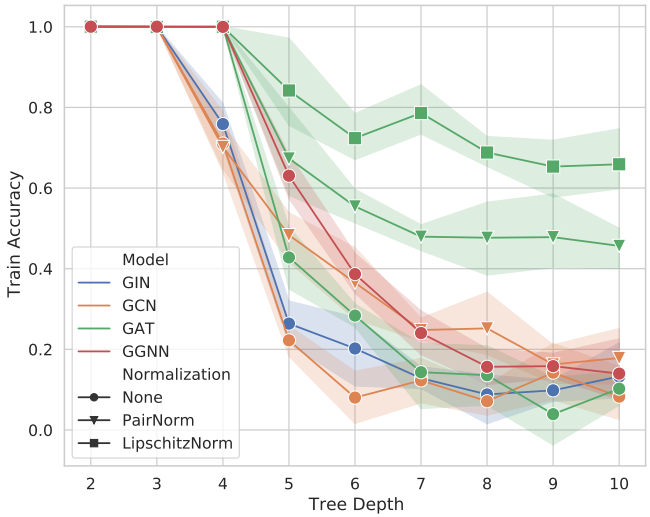
Gradient evolution of 25-layer GAT without/with LipschitzNorm



Model convergence w.r.t. model depth

A. Trees with increasing depth

- Synthetic graph benchmark (Alon et al, 2020).
- Simulation of *oversquashing*.

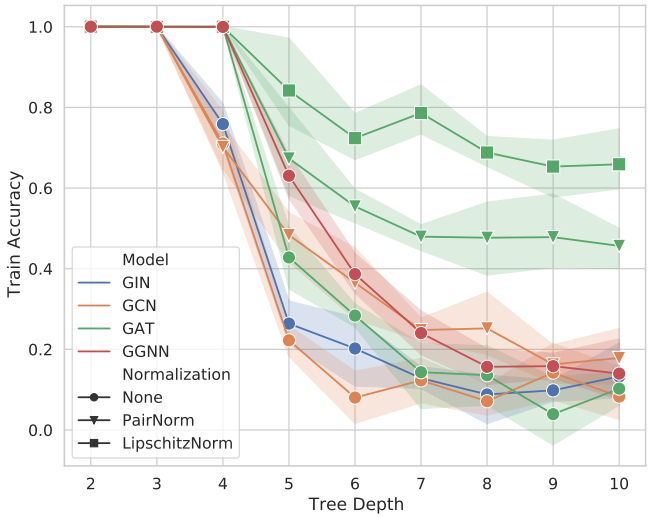


Train accuracies of four GNN models.

Synthetic Study

A. Trees with increasing depth

- Synthetic graph benchmark (Alon et al, 2020).
- Simulation of *oversquashing*.



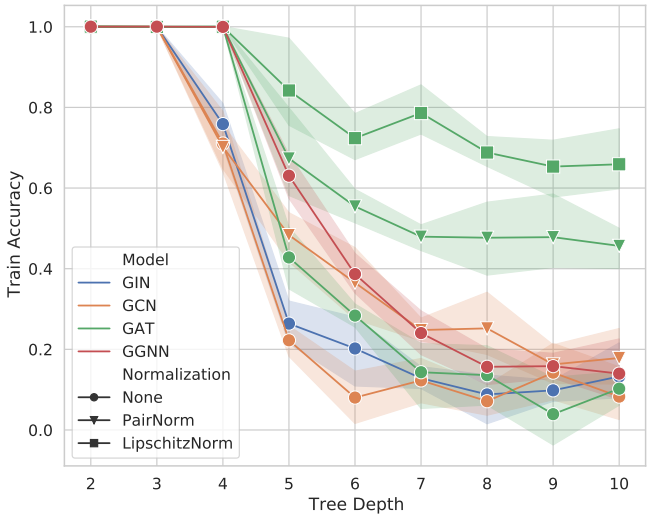
Train accuracies of four GNN models.

* Normalization always helps 😎
* LipschitzNorm for the win!

Synthetic Study

A. Trees with increasing depth

- Synthetic graph benchmark (Alon et al, 2020).
- Simulation of *oversquashing*.



Train accuracies of four GNN models.

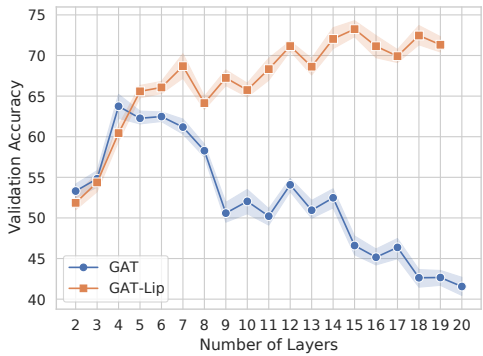
* Normalization always helps 😎

* LipschitzNorm for the win!

B. Node Classification with Missing Features

(Zhao and Akoglou, 2019)

	Cora		CiteSeer		Pubmed	
	0%	100%	0%	100%	0%	100%
GCN	82.5± 1.2 (2)	58.8 ± 3.5 (2)	**69.5 ± 2.1 (2)**	31.3 ± 2.7 (2)	77.9 ± 1.4 (2)	44.9 ± 4.4 (2)
GGNN	81.8 ± 2.0 (2)	68.2 ± 2.5 (6)	68.5 ± 1.9 (3)	40.5 ± 1.4 (5)	78.4 ± 2.1 (4)	56.6 ± 1.9 (4)
GAT	82.3 ± 2.3 (2)	65.3 ± 2.1 (4)	69.3 ± 1.6 (2)	42.8 ± 1.6 (4)	77.4 ± 0.5 (6)	63.1 ± 0.7 (4)
GAT-Pn	78.8 ± 0.6 (4)	73.8 ± 1.2 (12)	67.2 ± 0.8 (4)	**51.7 ± 1.1 (10)**	77.6 ± 1.6 (8)	70.4 ± 1.1 (12)
GAT-Lip	**83.1± 0.5 (5)**	**75.3± 0.9 (11)**	69.1 ± 1.5 (3)	50.9 ± 1.9 (9)	**78.9 ± 1.3 (5)**	**73.3 ± 1.4 (15)**

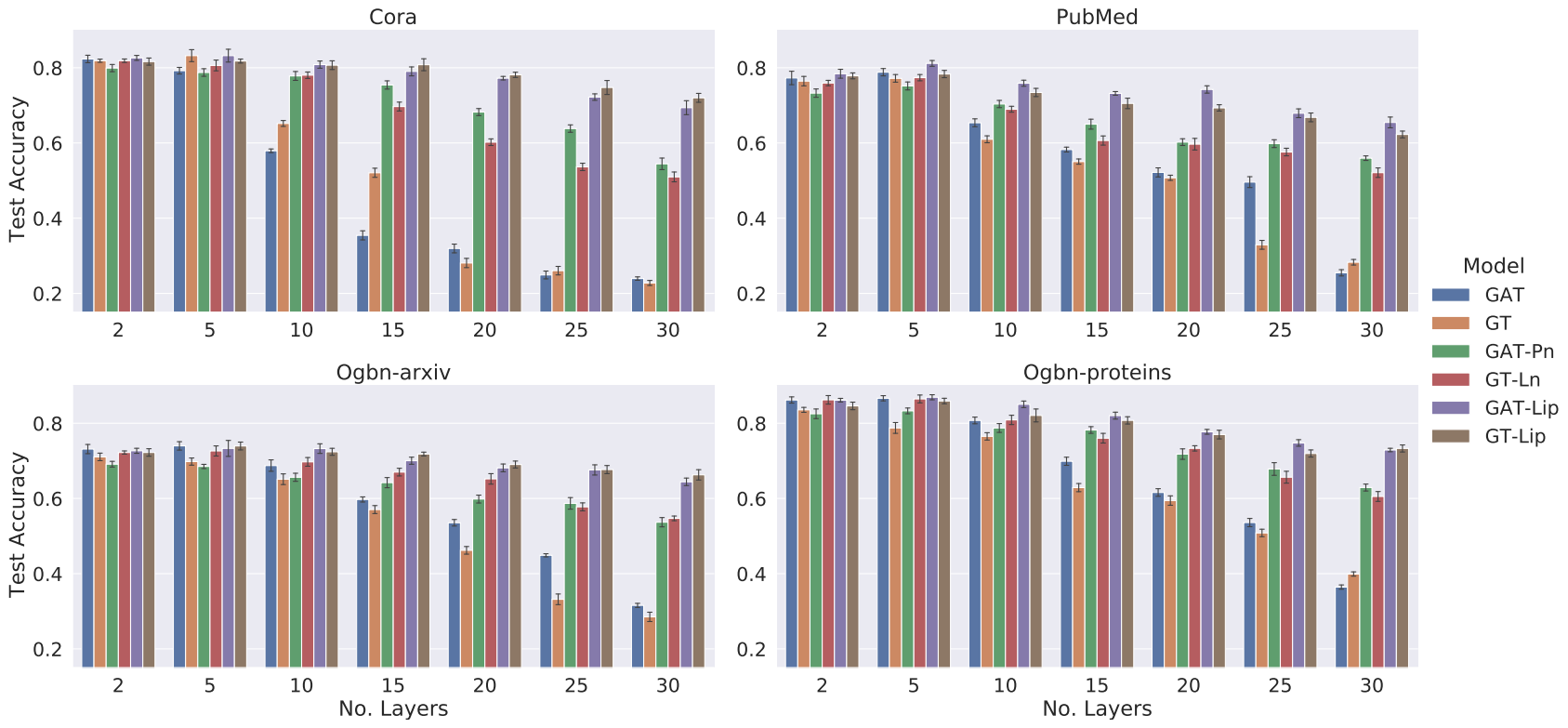


Accuracy of GAT for 100%-PubMed

* GAT-Lip: strong across shallow and deep settings!

Model depth in real-world datasets

- We evaluate the performance w.r.t. increasing number of layers.



Test accuracies of a GAT and a Graph Transformer (GT). By '-Lip' we denote the application of *LipschitzNorm*, by '-Ln' the *LayerNorm* and by '-Pn' the *PairNorm*.

Thank you for your normalized attention!

References

P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio,“Graph Attention Networks” *ICLR*, 2018.

L.Zhao & L.Akoglou,“PairNorm:Tackling Oversmoothing in GNNs” *ICLR*, 2019.

U. Alon & E. Yahav, “On the Bottleneck of Graph Neural Networks and its Practical Implications,” *ICLR*, 2021.

W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren,B. Liu, M. Catasta & J. Leskovec, “Open Graph Benchmark: Datasets for Machine Learning on Graphs,” *CoRR*, 2020.