
A APPENDIX

A.1 EXPERIMENTAL DETAILS

Dataset summary An overview of the characteristics of the datasets we used for our experiments is given in Table 1.

Table 1: Summary of the datasets used in our experiments.

	Cora	Citeseer	Pubmed	Actor	Cornell	Texas	Wisconsin
# Nodes(# Graphs)	2708 (1)	3327 (1)	19717 (1)	7600 (1)	183 (1)	183 (1)	251 (1)
# Edges	5429	4732	44338	33544	295	309	499
# Features/Node	1433	3703	500	931	1703	1703	1703
# Classes	7	6	3	5	5	5	5
# Training Nodes	140	120	60	60%	60%	60%	60%
# Validation Nodes	500	500	500	20%	20%	20%	20%
# Test Nodes	1000	1000	1000	20%	20%	20%	20%

Backbone architecture Here we specify setups of the backbone architectures.

MemGAT We mostly adopted the original architecture of GAT. Specifically, we used a two-layer GAT across all tasks. For all but the PubMed dataset, the first layer consists of 2 attention heads and 256 features each. The second layer is used for classification with a single attention head that computes C features (with C being number of classes), followed by a softmax activation. For Pubmed dataset, we used 8 output attention heads as in Veličković et al. (2018). For the edge weights, we picked $\psi_{ij} = S_{ij}$ where S is the edge weight matrix, and $\phi^{(l)}(h, h') = \text{LeakyRelu}(S_{ij}\langle a, [W^{(l)}h \parallel W^{(l)}h'] \rangle)$. The nonlinear function $\sigma^{(l)}$ was chosen identically across all models and all layers as the ELU function Clevert et al. (2015), with the exception that we did not use nonlinear transform of the input features.

MemGCN We used a two-layer GCN across all tasks with 256 hidden features.

Training We applied L_2 regularization with tuning parameter 0.001 and a dropout Srivastava et al. (2014) operation of probability 0.6 to each layer’s inputs for all transductive tasks. We applied entropy regularization Grandvalet & Bengio (2005) with tuning parameter 0.6 for Cora and Citeseer datasets (but not for the ablation study) and 0.1 for Pubmed dataset. All models used Glorot initialization Glorot & Bengio (2010) and cross entropy loss optimized using Adam Kingma & Ba (2014) with an initial learning rate of 0.01 for Pubmed dataset, and 0.005 for all the other datasets. We used an early stopping strategy on both the cross-entropy loss and accuracy on the validation nodes, with a patience of 100 epochs.

Optimal hyperparameters We report the optimal choice of hyperparameters in table 2.

Table 2: Optimal choice of hyperparameters, namely the incorporation of skip connection, the number of random walk steps K , and the type of transition matrix T . We tune K over the set $\{0, 1, 2, 3, 4, 5, 10, 20, 30\}$.

MemGAT				MemGCN		
Dataset	Cora	Citeseer	Pubmed	Cora	Citeseer	Pubmed
skip connection	✓	✓	✓	✗	✗	✗
K	30	30	0	3	1	30
T	A_{sym}	A_{sym}	A_{sym}	A_{sym}	A_{sym}	A_{sym}

MemGAT					MemGCN			
Dataset	Actor	Cornell	Texas	Wisconsin	Actor	Cornell	Texas	Wisconsin
skip connection	✓	✓	✓	✓	✓	✓	✓	✓
K	3	3	3	3	3	3	3	3
T	A_{sym}	A_{sym}	A_{sym}	A_{sym}	A_{sym}	A_{sym}	A_{sym}	A_{sym}

Implementation We implemented MemGAT based on the open source PyTorch Paszke et al. (2019) implementation of GAT Veličković et al. (2018) at <https://github.com/PetarV-/GAT>, and MemGCN based on the open source PyTorch implementation of GCN (Kipf & Welling, 2016) at <https://github.com/tkipf/pygcn>,

A.2 PROOF OF THEOREMS

Proof of theorem 1. We show by induction on l , for $l = 0$ it follows trivially since $X_v = X'_{f(v)}, \forall v \in V$, suppose for $l = L$ we have $h_v^{(L)} = h_{f(v)}^{(L)'}, \forall v \in V$, for $l = L + 1$, consider any $v \in V$, since $\text{STAR}(v)$ is isomorphic to $\text{STAR}(f(v))$ and the map f is surjective, it follows that the multiset representation of the feature vector $X_{\mathcal{N}_v}$ is identical to $X'_{\mathcal{N}_{f(v)}}$, thus an unordered aggregation function would produce $\tilde{h}_v^{(L)} = \tilde{h}_{f(v)}^{(L)'}$, we conclude that $h_v^{(L+1)} = \text{COMBINE} \left(h_v^{(L)}, \tilde{h}_v^{(L)} \right) = \text{COMBINE} \left(h_{f(v)}^{(L)'}, \tilde{h}_{f(v)}^{(L)'} \right) = h_{f(v)}^{(L+1)'}$. \square

Proof of lemma 1. For part (i), under condition C_1 there exists a parameter (hereafter referred to as identifier) $\vartheta_0^* = \Theta(\mathcal{X})$ that identifies every bounded subset (with subset in the sense of multiset) of \mathcal{X} up to distributional equivalence. The map Θ hence maps feature set to an "identifier" parameter. Since \mathcal{X} is a countable subset of some euclidean space, it's easy to find an element $\mathbf{m}_0 \notin \mathcal{X}$, and we let $\tilde{\mathcal{X}} = \mathcal{X} \cup \{\mathbf{m}_0\}$, it follows immediately that if we augment every $\{X\}$ into $\{X \cup \{\mathbf{m}_0\}\}$, the identifier $\tilde{\vartheta}_0^* = \Theta(\tilde{\mathcal{X}})$ over $\tilde{\mathcal{X}}$ identifies $\{X \cup \{\mathbf{m}_0\}\}$ and $\{X' \cup \{\mathbf{m}_0\}\}$ for any $\{X\} \neq \{X'\}$, since \mathbf{m}_0 always has a multiplicity of one and is distinct from all elements in \mathcal{X} .

The injectivity is therefore defined in the following sense: for any multiset $\{\tilde{X}\}$ satisfying:

- (i) Its underlying set \tilde{X} represented as $\tilde{X} = \{\mathbf{m}_0\} \cup X$ where X is a subset of \mathcal{X} with bounded size.
- (ii) The multiplicity of \mathbf{m}_0 is restricted to be one, and the multiplicities of other elements are uniformly bounded from above.

Then under identifier $\tilde{\vartheta}_0^*$, $\text{GNN}_{\tilde{\vartheta}_0^*}(\{\tilde{X}\}) = \text{GNN}_{\tilde{\vartheta}_0^*}(\{\tilde{X}'\})$ if and only if $\{\tilde{X}\} = \{\tilde{X}'\}$, which is equivalent to $\{X\} = \{X'\}$. Applying the previous argument iteratively, we obtain identifier for each layer $\tilde{\vartheta}_l^*, l \in \mathbb{N}$ and injectivity could be defined in similar ways. Part (ii) is a consequence of part (i) in that we choose X and X' to be the corresponding graph feature of G and G' . \square

Proof of theorem 2. By lemma 1, the output of the second MemGAT layer would be different for $h_v^{(2)}, h_{f(v)}^{(2)'}$, $\forall v \in V$, since the underlying feature space is countable, there exists a nonlinear function σ satisfying $\sigma(h_v^{(2)}) < 0.5, \sigma(h_{f(v)}^{(2)'}) \geq 0.5, \forall v \in V$. Picking the readout function as σ finishes the proof. Note also that this function could be approximated by universal approximators like multi layer perceptrons. \square

Proof of corollary 1. We first show for GCN. Note that GCN has several different definitions, we will follow the general definition in Dehmamy et al. (2019) without bias term:

$$H^{(l+1)} = \sigma\left(\tau(A)H^{(l)}W\right) \quad (1)$$

where $H^{(l)}$ is the matrix stacked by hidden representations of each node in the l th layer, and $\tau : \mathbb{R}^{N \times N} \mapsto \mathbb{R}^{N \times N}$ is a matrix transformation operation. With $D = \text{diag}(AI_N)$, two popular forms of GCN are defined as

Kipf & Welling (2016) uses $\tau_1(A) = (D + I_n)^{-1/2}(A + I_N)(D + I_N)^{-1/2}$, and σ is RELU.

Xu et al. (2019) uses $\tau_2(A) = (D + I_n)^{-1}(A + I_N)$ which reduces to mean pooling, and σ is RELU.

The fact that GCN formulated by τ_2 satisfied condition C_1 is directly implied by Xu et al. (2019, Corollary 8). But for GCN induced by τ_1 , the identifiability result need not hold since the aggregation process of each node $v \in V$ is determined not solely by its neighborhood information, but also by the degree of its neighborhoods which could be arbitrary. Nevertheless, we could still gain insights from this (more popular) design by noting that with respect to regular graphs, the identifiability issue of both formulations are the same, and are mitigated via memory augmentation.

For GAT, consider the worst case of two multisets with their underlying set identical with a single element but different in multiplicities. In this case, regardless of the attention mechanism, GAT is identical to mean pooling. Hence it suffices to choose the identifier obtained from (Xu et al., 2019, Corollary 8) over mean pooling that makes GAT identify multisets up to distributional equivalence. \square

A.3 ON THE EFFECT OF NUMBER OF MEMORY NODES

In this section we present a study on training MemGAT on the Cora dataset using different number of memory nodes M . The tuning range is $\{1, 3, 5, 7, 9, 11, 13, 15, 17\}$. The rest of the hyperparameters are the same as the one reported in table ???. The results are reported in table 3. The result shows little performance difference in using different number of memory nodes.

A.4 ARCHITECTURES UNSUITABLE FOR MEMORY AUGMENTATION

GNN architectures that utilize max pooling for aggregating operation may not identify distributionally equivalent instances (Xu et al., 2019, Corollary 9), hence the max-pooling version of GraphSAGE (Hamilton et al., 2017) is not a proper backbone architecture for memory augmentation. GIN (Xu et al., 2019) uses sum pooling that is strictly more expressive than mean pooling hence satisfies condition C_1 . However, summing up feature vectors of the whole graph increases numerical instability and is empirically found hard to train.

A.5 COMPARISONS WITH OTHER MESSAGE PASSING VARIANTS

Comparison with CPCGNN CPCGNN Sato et al. (2019) utilizes a *consistent port numbering* that numbers the neighbors of each node v by an integer $i \in [\text{degree}(v)]$, according to a *port numbering function* p such that $p(v, i) = (u, j)$ identifies the neighboring node u labeled i and a port number $j \in [\text{degree}(u)]$. The port numbering rule is said to be *consistent* if $p(p(v, i)) = (v, i)$ for any valid (v, i) pairs. CPCGNN allows node v sending messages to node u depending on both its own feature and the port number of u , thus forms a certain kind of locally ordered message passing framework that is strictly more expressive than locally unordered GNNs. However was shown in

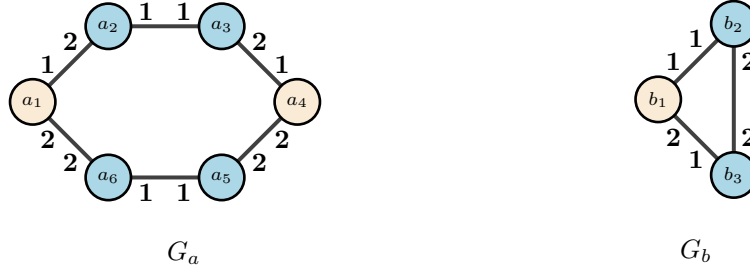


Figure 1: Consistent port numberings for G_a and G_b that makes them locally distinguishable

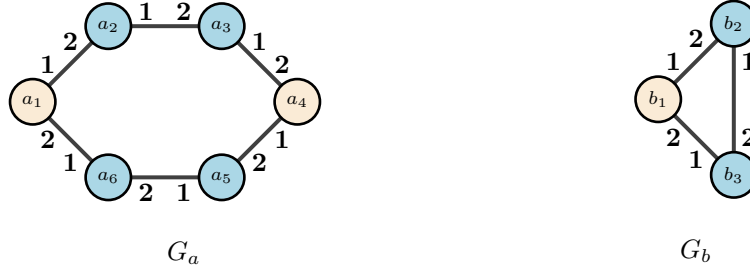


Figure 2: Consistent port numbering for G_a and G_b that fails to distinguish a_1, a_4 and b_1

Garg et al. (2020) that since consistent port numbering functions are *non-unique*, there exists some port numbering functions that does not strengthen expressiveness, we illustrate this phenomenon using the construction in figure 1 and figure 2. Figure 1 shows a port numbering that makes a_1, a_4 receiving different messages with that of b_1 . Meanwhile in figure 2 the port numbering can not distinguish a_1, a_4 and b_1 . Finding a consistent port numbering that succeeds in distinguishing local structures is yet another challenging task, MemGAT thus offers an easier choice when the two graphs have different global features.

Comparison with DimeNet DimeNet Klicpera et al. (2020) is a *directional* message passing model that exploits the relative layout of local neighborhood through angles. Specifically DimeNet computes node embedding $h_v^{(l)}$ as the summation of its incoming message embeddings $h_v^{(l)} = \sum_{u \in \mathcal{N}_v} m_{uv}^{(l)}$, and the update rule is defined as

$$m_{uv}^{(l)} = f_{\text{update}} \left(m_{uv}^{(l-1)}, \sum_{w \in \mathcal{N}_v \setminus u} f_{\text{integrate}} \left(m_{wv}^{(l-1)}, e^{(uv)}, a^{(wu, uv)} \right) \right) \quad (2)$$

where $f_{\text{integrate}}$ and f_{update} are analogs of aggregate and combine as in LUMP protocol, $e^{(uv)}$ is a representation vector measuring the distance from u to v , and $a^{(wu, uv)}$ combines $\angle wuv$ with the distance from w to u . The choice of metric is problem dependent, and we presume a suitable one exists. Consider the following construction:

Table 3: Study on number of memory nodes on the Cora dataset using MemGAT model. Results (%mean \pm %standard deviation test set accuracy) are computed over 100 trials

M	Performance
1	84.60 \pm 0.58
3	84.62 \pm 0.58
5	84.70 \pm 0.52
7	84.64 \pm 0.52
9	84.71 \pm 0.54
11	84.69 \pm 0.67
13	84.78 \pm 0.59
15	84.78 \pm 0.59
17	84.76 \pm 0.60

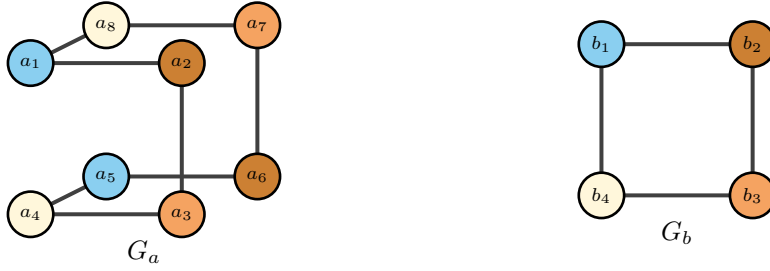


Figure 3: Two graphs $G_a = (V_a, E_a)$ and $G_b = (V_b, E_b)$ that DimeNet cannot distinguish locally: for any $V \in \{V_a, V_b\}$ and all $w, u, v \in V$ satisfying $(u, w) \in E, (u, v) \in E$, the graph is constructed such that $\angle wuv = \pi/2$ and for any $(u, v) \in E$, the distance between u and v is identical.

Figure 3 shows a construction that DimeNet fails to distinguish with the local isomorphism map defined as $f(a_1) = f(a_5) = b_1, f(a_2) = f(a_6) = b_2, f(a_3) = f(a_7) = b_3, f(a_4) = f(a_8) = b_4$, while MemGAT is able to distinguish them.

The above contrived examples suggest that the optimal choice of GNN architecture shall be problem dependent.

REFERENCES

- Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint 1511.07289*, 2015.
- Nima Dehmamy, Albert-László Barabási, and Rose Yu. Understanding the representation power of graph neural networks in learning graph topology. In *Advances in Neural Information Processing Systems*, pp. 15413–15423, 2019.
- Vikas K Garg, Stefanie Jegelka, and Tommi Jaakkola. Generalization and representational limits of graph neural networks. *arXiv preprint arXiv:2002.06157*, 2020.
- Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterton (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL <http://proceedings.mlr.press/v9/glorot10a.html>.
- Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pp. 529–536, 2005.
- Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pp. 1024–1034, 2017.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- Johannes Klicpera, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, 2020. URL <https://openreview.net/forum?id=BlEWbxStPH>.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL <http://papers.neurips.cc/paper/>

9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Ryoma Sato, Makoto Yamada, and Hisashi Kashima. Approximation ratios of graph neural networks for combinatorial problems. In *Advances in Neural Information Processing Systems*, pp. 4083–4092, 2019.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL <http://jmlr.org/papers/v15/srivastava14a.html>.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL <https://openreview.net/forum?id=rJXMpikCZ>.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=ryGs6iA5Km>.