Enhancing Heterophilic Graph Representations in Structure-Aware Transformers

Robin Renggli [*1] **Anisha Mohamed Sahabdeen** [*1] **Lukas Münzel** [*1] **Matin Urdu** [*1]

1. Introduction

Recently, efforts have been made to generalize transformer models (Vaswani et al., 2017) to the graph domain, following their success in modeling sequential data in diverse fields, ranging from natural language processing to biological sequence generation. Leveraging the original self-attention mechanism (Vaswani et al., 2017), transformers are able to aggregate information globally and adaptively, capturing the interaction between any pair of nodes. Nonetheless, due to the mechanism's independence from the graph's edges, the self-attention module produces node features which lack any meaningful information about the graph underlying structure (Ying et al., 2021). To overcome this limitation, most existing approaches adopt a *positional* encoding method (Shaw et al., 2018), which captures the relative position or distances between nodes on the graph, albeit disregarding some edge-related knowledge. More promising work (Chen et al., 2022; Zhang et al., 2023) proposes to incorporate local *structural* information into the node embeddings, using specific Graph Neural Networks (GNNs) to extract subgraph representations centered around each node. Formally, given an arbitrary k-layer GNN model $GNN^{(k)}$ applied to a graph $\mathcal{G}$ with node features $\mathbf{X}$, we define the structure extractor φ representing the k-hop subtree rooted at node u as follows:

$$\varphi(u, \mathbf{X}, \mathcal{G}) = (GNN^{(k)}(\mathbf{X}, \mathcal{G}))_u \ . \tag{1}$$

However, the performance of GNNs usually degrades with an increase in the number of layers (Cai & Wang, 2020). One of the key reasons why this occurs is a phenomenon known as over-smoothing, which describes the exponential convergence of node features to the node features of their neighbourhood (Oono & Suzuki, 2019; Rusch et al., 2023). In particular, over-smoothing poses a significant problem in *heterophilic* graphs, where the loss of dissimilarity between neighboring nodes destroys crucial features in the graph representation (Rusch et al., 2023). Hence, the number of layers of $GNN^{(k)}$ must be limited because of its susceptibility to over-smoothing. This constrains the size of the subgraph that φ is able to encode. However, considering more extensive node-centered subgraphs could improve the expressive power of the model's node embeddings.

Contributions. Motivated by recent research suggesting the need for specialized methods in order to achieve strong performance on graphs with low homophily (Platonov et al., 2023), in this work, we leverage the generalization power of Structure-Aware Transformers (SATs) (Chen et al., 2022) in heterophilic settings. In particular, our primary focus is directed towards the analysis and enhancement of the k-hop structure extractor φ: (i) we investigate whether increasing the number of hops is beneficial to the model's performance while addressing potential over-smoothing challenges; (ii) we apply novel GNN techniques to generate more informative embeddings. We empirically validate our SAT-based models, achieving competitive or state-of-the-art results on heterophilic benchmarks.

2. Models and Methods

In this section, we introduce the main mechanisms of SATs and explore diverse methods to enhance the model's embeddings.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E} \subseteq \mathcal{V} \times \mathcal{V})$ be an undirected graph with $|\mathcal{V}| = v$ nodes. Each node u is represented by a feature vector $\mathbf{x}_u \in \mathbb{R}^m$. These $m-$dimensional feature vectors can be gathered into one matrix $\mathbf{X} \in \mathbb{R}^{v \times m}$. New hidden representations $\mathbf{X}^n \in \mathbb{R}^{v \times m}$ are generated iteratively as follows:

$$\mathbf{X}^0 = \mathbf{X},$$
$$\mathbf{X}^n = \sigma\left(\mathbf{F}_{\theta^n}\left(\mathbf{X}^{n-1}, \mathcal{G}\right)\right), \quad \text{for } n = 1, \ldots, N \ ,$$

where each $\mathbf{F}_{\theta^n}$ is a learnable function with parameters θ^n and σ is an element-wise non-linear activation function (Rusch et al., 2023).

2.1. Structure-Aware Graph Transformer

Each SAT layer is composed of two main blocks: a self-attention module followed by a feed-forward neural network (FFN).

We define the trainable parameters $\mathbf{W_Q}, \mathbf{W_K}, \mathbf{W_V} \in$

[*]Equal contribution [1]ETH Zürich, Zürich, Switzerland.

$\mathbb{R}^{m \times d_{out}}$ which linearly project an input to the query, key and value matrices. The structure-aware self-attention mechanism, introduced by Chen et al. (2022), is defined as:

$$\text{SA-attn}(u;\varphi) := \sum_{v \in \mathcal{V}} \frac{\kappa_{\exp}(\hat{\mathbf{x}}_u, \hat{\mathbf{x}}_v)}{\sum_{w \in \mathcal{V}} \kappa_{\exp}(\hat{\mathbf{x}}_u, \hat{\mathbf{x}}_w)} \mathbf{W_V} \hat{\mathbf{x}}_v, \forall u \in \mathcal{V} ,$$

where $\hat{\mathbf{x}}_\circ := \varphi(\circ, \mathbf{X}, \mathcal{G})$, $\varphi : \mathbb{R}^m \to \mathbb{R}^m$ is a structure extractor as defined in Eq. 1 and $\kappa_{\exp}$ is a non-symmetric exponential kernel on $\mathbb{R}^m \times \mathbb{R}^m$ parameterized by $\mathbf{W_Q}$ and $\mathbf{W_K}$. Specifically, we define:

$$\kappa_{\exp}(\mathbf{x}, \mathbf{x}') := \exp\left(\langle \mathbf{x}\mathbf{W_Q}, \mathbf{x}'\mathbf{W_K} \rangle / \sqrt{d_{\text{out}}}\right).$$

Chen et al. (2022) extend both the self-attention mechanism and the FFN with post-processing blocks, $\text{Post-Process}_{attn}$ and $\text{Post-Process}_{\text{FFN}}$ respectively, each including a modified skip-connection (Mialon et al., 2021), a feed-forward neural network and two normalization layers.

With a slight overload of notation on SA-attn, the n-th transformer layer can be computed as follows:

$$\begin{aligned} \mathbf{X}' &= \text{Post-Process}_{attn}(\text{SA-attn}(\mathbf{X}^{n-1};\varphi)) ; \\ \mathbf{X}^n &= \text{Post-Process}_{\text{FFN}}(\text{FFN}(\mathbf{X}')) \\ &= \text{Post-Process}_{\text{FFN}}(\text{ReLU}(\mathbf{X}'W_1)W_2) , \end{aligned}$$

where W_1 and W_2 are trainable matrices.

2.2. Methods for improving the structure extraction

We propose to apply the following techniques directly to φ (Eq. 1) in order to enhance the stage of structure extraction of the SAT.

Gradient Gates. Gradient Gating (Rusch et al., 2022) is a novel gating-based framework inspired by residual message-passing GNN updates that has been shown to theoretically and empirically alleviate over-smoothing. The graph gradient is used to implement a learnable early stopping mechanism for each node. Given another learnable 1-neighborhood coupling function $\hat{\mathbf{F}}_{\hat{\theta}}$, Rusch et al. (2022) define the gradient framework as

$$\begin{aligned} \hat{\boldsymbol{\tau}}^n &= \sigma\left(\hat{\mathbf{F}}_\theta\left(\mathbf{X}^{n-1}, \mathcal{G}\right)\right) , \\ \boldsymbol{\tau}^n_{ik} &= \tanh\left(\sum_{j \in \mathcal{N}_i} \left|\hat{\boldsymbol{\tau}}^n_{jk} - \hat{\boldsymbol{\tau}}^n_{ik}\right|^p\right) , \\ \mathbf{X}^n &= (1 - \boldsymbol{\tau}^n) \odot \mathbf{X}^{n-1} + \boldsymbol{\tau}^n \odot \sigma\left(\mathbf{F}_\theta\left(\mathbf{X}^{n-1}, \mathcal{G}\right)\right) , \end{aligned}$$

where $\boldsymbol{\tau}^n, \hat{\boldsymbol{\tau}}^n \in \mathbb{R}^{v \times m}$ and $p \in \mathbb{R}^+_0$ is an hyperparameter. Empirically, increasing p corresponds to an earlier application of the early stopping mechanism.

Directed GNNs (Dir-GNNs) The Dir-GNN framework extends GNNs to directed graphs by performing two separate aggregations for each node $i \in V$, with respect to the in-neighbours $(j \to i)$ and the out-neighbours $(i \to j)$ respectively (Rossi et al., 2023). Formally, we can write

$$\begin{aligned} \mathbf{X}^n_{i,\leftarrow} &= \mathbf{F}_{\theta^n,\leftarrow}\left(\left\{\left(\mathbf{X}^{n-1}_j, \mathbf{X}^{n-1}_i\right) : (j,i) \in E\right\}\right) \\ \mathbf{X}^n_{i,\rightarrow} &= \mathbf{F}_{\hat{\theta}^n,\rightarrow}\left(\left\{\left(\mathbf{X}^{n-1}_j, \mathbf{X}^{n-1}_i\right) : (i,j) \in E\right\}\right) \\ \mathbf{X}^n_i &= \alpha \cdot \mathbf{X}^n_{i,\leftarrow} + (1-\alpha) \cdot \mathbf{X}^n_{i,\rightarrow}, \end{aligned}$$

where $\mathbf{F}_{\theta^n,\leftarrow}$ is the one directional aggregation over in-neighbours $(j \to i)$ with learnable parameters θ^n and $\mathbf{F}_{\hat{\theta}^n,\leftarrow}$ is the one directional aggregation with learnable parameters $\hat{\theta}^n$ over out-neighbours $(i \to j)$. The learnable parameter α weights the influence of the aggregation of in-neighbours and out-neighbours respectively.

2.3. Methods for improving the final node embeddings

Jumping Knowledge Networks (JKNs). The goal of JKNs is to address limitations in many commonly used neighbourhood aggregation schemes in graph representation learning by aggregating the node features of every layer at the final layer (Xu et al., 2018).

$$\mathbf{X}_{out} = \text{AGGR}(\mathbf{X}^0, \ \ldots \ , \mathbf{X}^N)$$

Depending on the aggregation scheme that is used at the final layer, the model learns to select which node features from the intermediate representations can "jump" to the final layer (Xu et al., 2018).

3. Results

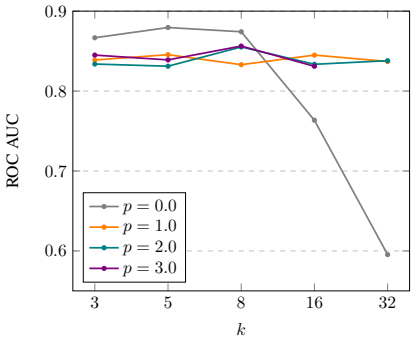
3.1. Heterophilic Datasets

In this section, we present the findings of our investigation on the heterophilic datasets Roman-Empire and Minesweeper (Platonov et al., 2023) for node classification tasks. We intentionally use these datasets due to their high heterophily, task-relevant graph structure, and suitable size for statistically significant results within hardware and time constraints. Additionally, the two datasets originate from different domains and possess distinct structural qualities: their inherent diversity allows for a better generalization of our results.

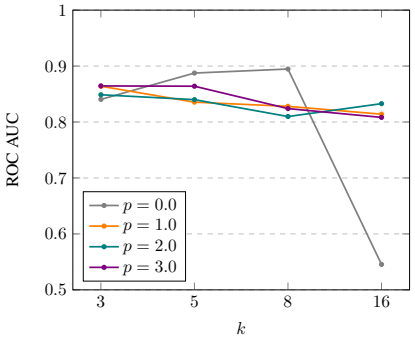
We investigate the potential performance improvements resulting from applying the techniques outlined in Section 2 to a baseline SAT. These includes:

(i) the application of the Dir-GNN framework to the structure extractor;

(ii) the increase in the number of hops in the structure extractor alongside the application of the gradient gating mechanism;

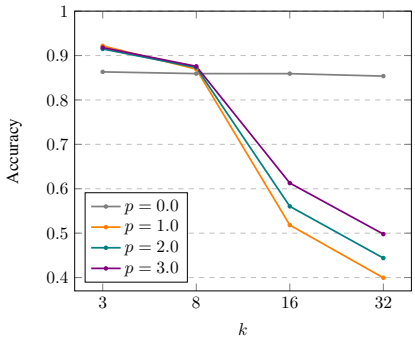


(a) Enhanced φ-undirected SAT

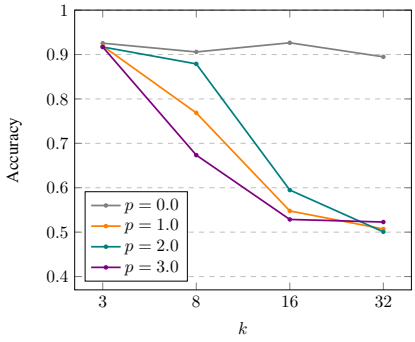


(b) Enhanced φ-directed SAT

Figure 1. k-p study on our enhanced SAT models on Minesweeper. Models used for the k-p study were intentionally kept smaller to stay within the time and hardware constraints (see Table 3).



(a) Enhanced φ-undirected SAT



(b) Enhanced φ-directed SAT

Figure 2. k-p study on our best enhanced SAT models on Roman Empire.

(iii) the incorporation of a JKN to aggregate the outputs of the transformer blocks.

In addition to comparing our enhanced transformer to its standard SAT version, we also assess its performance against methods specifically designed for node classification under heterophily, namely the GBK-GNN (Du et al., 2022), which uses a bi-kernel for feature transformation, and the FSGNN (Maurya et al., 2021), which performs feature selection.

3.1.1. Minesweeper

This graph is a regular 100×100 grid with each node being connected to all eight neighbours. The node features describe the number of neighbouring mines, inspired by the game of the same name. The task is to predict which nodes are mines.

An evaluation of the effect of gating and directionality applied to Minesweeper can be seen in Figure 1. While the best performing models do not use gating, we observe an increased stability of the models performance across different number of hops when gating is applied. Gated models achieved ROC AUC scores of over 80% even up to 32 hops, while ungated high-hop models showed poor performance. Further, the directional GNN outperforms the undirected one in most k-p settings.

Based on the k-p study results in Figure 1, we ran an ablation study on deeper models (Table 5), which produced ROC AUC scores outperforming all baselines Table 1, highlighting the strengths of our model in the heterophilic setting. We observed the largest gains in model accuracy from using a directional structure extractor in combination with jumping knowledge networks with LSTM aggregation, whereas increasing the number of hops and the additions of gates did not improve the results. This could be due to the absence of long range dependencies in the Minesweeper graph.

3.1.2. Roman-Empire

The dataset is based on the Roman Empire Wikipedia article. Nodes denote words, connected if they follow sequentially or share a dependency tree link. Node classes correspond to their syntactic role.

Table 1 summarizes the performance of the best SAT model and enhanced versions. We successfully show that our best model, using a directed structure extractor and jumping knowledge with LSTM aggregation on top of the standard SAT, improves the baseline with an accuracy gain of 10.46%. To the best of our knowledge, this result outperforms the cur-

	MINESWEEPER	ROMAN-EMPIRE
# NODES	$10,000$	$22,662$
# EDGES	$39,402$	$32,927$
# FEATURES	7	300
# CLASSES	2	18
METRIC	ROC AUC	ACCURACY
GBK-GNN	90.85 ± 0.58	74.57 ± 0.47
FSGNN	90.08 ± 0.70	79.92 ± 0.56
SAT	93.62	82.09 ± 0.14
OURS (UNDIRECTED)	94.02	91.90 ± 0.25
OURS (DIRECTED)	**94.75**	$\mathbf{92.55 \pm 0.02}$

Table 1. Model performance on Minesweeper and Roman-Empire. Performance of GBK-GNN and FSGNN are reported from Platonov et al. (2023). The "SAT" baseline and our best SAT-enhanced φ-directed and φ-undirected models ("Ours (Undirected)", "Ours (Directed)") are described in relevant dataset sections and in the Appendix (Section B.1). Performance of the SAT and SAT-enhanced models are computed over three runs for Roman-Empire and one run for Minesweeper.

	CLUSTER
# GRAPHS	$12,000$
# AVG. NODES	117.20
# AVG. EDGES	4301.72
# CLASSES	6
METRIC	ACCURACY
SAT	75.23
SAT+JKN	76.03
SAT+DIR	75.66
SAT+JKN+DIR	76.15

Table 2. Model performance on Cluster. The "SAT" baseline configuration is described in Table 8. "JKN" and "Dir" indicate the addition of a JKN with an LSTM aggregator and a φ-directed respectively. Due to time constraints, results over a single run are reported.

rent state-of-the-art on this dataset, which achieved 91.64% of accuracy (Zhao et al., 2023b). A detailed description of the baseline SAT model and an ablation study of our best performing models can be found in the Appendix, Section B.1.2.

Figure 2 shows an evaluation of the effect of increasing the number k of hops in the structure extractor of our best enhanced models while applying gradient gates with different hyperparameters p. We generally observe that ungated models ($p = 0.0$) exhibit stability in performance as k increases, probably due to the presence of long-term dependencies in the graph. This suggests that our SAT models are not prone to over-smoothing on this dataset, yet no improvement occurs from considering more extensive subgraphs for producing node representations. For undirected models, gating improves classification accuracy for $k \leq 8$, but leads to a quick decline for a higher number of hops. In the directed case, all gated models perform worse, degenerating at a faster rate as k increases.

3.2. Homophilic Datasets

To extend the scope of our analysis, we also investigate the performance of our enhanced transformer model on the homophilic dataset Cluster (Dwivedi et al., 2022), which is one of the node-classification benchmarks that Chen et al. (2022) originally tested the SAT on. We compare performance to a baseline SAT, whose architecture is described in the Appendix (Table 8).

Results of our experiments are shown in Table 2. Our enhanced model, incorporating both directionality and jumping knowledge, exhibited a 0.92% increase in accuracy compared to the baseline SAT. This suggests that our methods might also be beneficial in homophilic settings, though further testing would be necessary to make any stronger claims in that regard.

4. Discussion & Summary

Conclusions We have shown that enhancing the graph representations in Structure-Aware Transformers can lead to significantly improved performance in heterophilic settings, outperforming SOTA results on Roman-Empire and achieving near-SOTA on Minesweeper. While directional structure-extracting GNNs as well as JKNs seem like direct improvements when compared to the base SAT model, the case is less clear for the use of gradient gating, which only seems to help in some cases, namely with specific dataset and GNN-depth configurations. In terms of the GNN depth, we've shown that our additions can improve the performance of deeper structure extractors. However, this did not result in overall performance gains, especially for large values of k.In general, the optimal model configuration seems heavily dataset-dependent, which might be related to the presence and the length of long-range dependencies in a given dataset.

Future Work While all three of our enhancements show great promise in improving the model performance in the heterophilic setting, it remains unclear how well they generalize. As the ideal number of layers in the structure-extracting GNN as well as the effects of the gating-parameter p are significantly different for our two main datasets, it is reasonable to assume that they are heavily dataset dependent. It would be interesting to investigate which factors in a dataset make it favour certain settings as well as trying to experiment with more flexible models. Two such approaches could be making the gating-parameter p learnable or using an ensemble of structure extractors with different depths. It would also be interesting to test significantly deeper architectures in terms of the number of transformer blocks using our methods, which we, due to hardware and time restrictions, could not do.

References

Cai, C. and Wang, Y. A note on over-smoothing for graph neural networks. *arXiv preprint arXiv:2006.13318*, 2020.

Chen, D., O'Bray, L., and Borgwardt, K. Structure-aware transformer for graph representation learning. In *International Conference on Machine Learning*, pp. 3469–3489. PMLR, 2022.

Du, L., Shi, X., Fu, Q., Ma, X., Liu, H., Han, S., and Zhang, D. Gbk-gnn: Gated bi-kernel graph neural networks for modeling both homophily and heterophily, 2022.

Dwivedi, V. P., Joshi, C. K., Luu, A. T., Laurent, T., Bengio, Y., and Bresson, X. Benchmarking graph neural networks, 2022.

Li, P., Wang, Y., Wang, H., and Leskovec, J. Distance encoding: Design provably more powerful neural networks for graph representation learning, 2020.

Maurya, S. K., Liu, X., and Murata, T. Simplifying approach to node classification in graph neural networks, 2021.

Mialon, G., Chen, D., Selosse, M., and Mairal, J. Graphit: Encoding graph structure in transformers. *arXiv preprint arXiv:2106.05667*, 2021.

Noci, L., Li, C., Li, M. B., He, B., Hofmann, T., Maddison, C., and Roy, M. D. The shaped transformer: Attention models in the infinite depth-and-width limit. *arXiv preprint arXiv:2306.17759*, 2023.

Oono, K. and Suzuki, T. Graph neural networks exponentially lose expressive power for node classification. *arXiv preprint arXiv:1905.10947*, 2019.

Platonov, O., Kuznedelev, D., Diskin, M., Babenko, A., and Prokhorenkova, L. A critical look at the evaluation of gnns under heterophily: are we really making progress? *arXiv preprint arXiv:2302.11640*, 2023.

Rossi, E., Charpentier, B., Di Giovanni, F., Frasca, F., Günnemann, S., and Bronstein, M. Edge directionality improves learning on heterophilic graphs. *arXiv preprint arXiv:2305.10498*, 2023.

Rusch, T. K., Chamberlain, B. P., Mahoney, M. W., Bronstein, M. M., and Mishra, S. Gradient gating for deep multi-rate learning on graphs. *arXiv preprint arXiv:2210.00513*, 2022.

Rusch, T. K., Bronstein, M. M., and Mishra, S. A survey on oversmoothing in graph neural networks. *arXiv preprint arXiv:2303.10993*, 2023.

Shaw, P., Uszkoreit, J., and Vaswani, A. Self-attention with relative position representations, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K.-i., and Jegelka, S. Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning*, pp. 5453–5462. PMLR, 2018.

Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., and Liu, T.-Y. Do transformers really perform bad for graph representation?, 2021.

Zhang, P., Yan, Y., Li, C., Wang, S., Xie, X., and Kim, S. Can transformer and gnn help each other?, 2023.

Zhao, H., Ma, S., Zhang, D., Deng, Z.-H., and Wei, F. Are more layers beneficial to graph transformers? *arXiv preprint arXiv:2303.00579*, 2023a.

Zhao, K., Kang, Q., Song, Y., She, R., Wang, S., and Tay, W. P. Graph neural convection-diffusion with heterophily, 2023b.

A. Project/Proposal Divergence Explaination

A.1. On the goal of mitigating over-smoothing

While our initial proposal focused on the idea of mitigating over-smoothing in order to improve the model performance, it does not seem as if our methods' strong performance is correlated with a reduction in over-smoothing. Our failure cases are most likely due to over-fitting and not over-smoothing issues, which is reflected by the loss curves. Models involving deeper Structure Extractors seem especially susceptible to over-fitting; gradient gates seem to mitigate this issue.

A.2. On the investigation of Shaped Attention

Early experiments with shaped attention did not seem promising. A reason for this might simply be that our testing configurations were shallow in the number of transformer blocks when compared to the 18 to 24 layers used in the preliminary experiments by Noci et al. (2023).

A.3. On the use of ZINC as baseline dataset

While we originally proposed to run our baselines on the ZINC dataset, this was not an ideal choice. The graphs contained in the dataset are fairly small (average of 23.2 nodes per graph), which does not make larger k-hops more informative. We thus decided to conduct our analysis on other datasets which could better represent our goals and setting.

A.4. On the use of DeepGraph as baseline model

We originally intended to use DeepGraph (Zhao et al., 2023a) as a baseline model, due to the fact that is another example of structure-aware graph transformer. However, since our model is fundamentally derived from the SAT architecture, we decided to focus our comparative analysis solely on the SAT model for a more direct evaluation and in-depth examination of the enhancements introduced through our modifications.

B. Training Description & Ablation Study

B.1. Heterophilic Datasets

B.1.1. MINESWEEPER

All experiments for Minesweeper were conducted on a Nvidia GeForce GTX 1070. We used the Adam optimizer with a learning rate of 0.0005 and trained the models for 2000 epochs using early stopping on the validation accuracy with a patience of 300.

Table 3 describes the configuration of the enhanced SAT model used for the k-p study on Minesweeper. Table 4 describes the configuration of the SAT baseline model used for the ablation study on Minesweeper. Table 5 presents the ablation study for our best performing SAT-based model on Minesweeper.

HYPERPARAMETER	VALUE
GNN TYPE (φ)	GRAPHSAGE
# LAYERS	4
HIDDEN DIMENSIONS	32
FFN HIDDEN DIMENSIONS	64
# ATTENTION HEADS	1
DROPOUT	0.5

Table 3. Configuration used for the k-p study on Minesweeper.

HYPERPARAMETER	VALUE
GNN TYPE (φ)	PNA
# LAYERS	7
HIDDEN DIMENSIONS	32
FFN HIDDEN DIMENSIONS	64
# ATTENTION HEADS	1
DROPOUT	0.5
SIZE OF SUBGRAPHS (k)	3 / 5

Table 4. Configuration of baseline SAT model on Minesweeper. The subgraph size is 3 for the undirected, 5 for the directed case..

MODEL	MODEL SIZE	ROC AUC (%)	$\Delta_{\text{ROC AUC}}$
BASE	146 K	93.62	
BASE + ABS PE	147 K	93.00	-0.62
BASE + NODE PROJ	153 K	89.53	-4.09
BASE + JKN	178 K	91.95	-1.67
BASE + NODE PROJ + JKN	**185 K**	**94.02**	**+0.4**
BASE + NODE PROJ + JKN + GATES	185 K	86.92	-6.7
BASE + DIR	329 K	94.67	+1.05
BASE + NODE PROJ + JKN + DIR	**367 K**	**94.75**	**+1.13**
BASE + NODE PROJ + JKN + DIR + GATES	367 K	92.57	-1.05

Table 5. Ablation study of best SAT-based models on Minesweeper. The column "Model Size" refers to the number of parameters of the model. The column "$\Delta_{\text{ROC AUC}}$" is equal to the difference in percentage ROC AUC compared to the base SAT model. The label "Abs PE" refers to the addition of a random walk positional encoding to the transformer model (Li et al., 2020; Dwivedi et al., 2022). The label "Node Proj" refers to the insertion of a 3-layer MLP before the transformer model to better project the node features in a latent space. Labels "JKN", "Gates" and "Dir" refer to the addition of jumping knowledge with LSTM aggregators, gradient gating and the use of directed GNNs respectively, in the manner described in Section 3. Gating parameter p as well as hop size k where chosen according to the best results from our k-p study (Figure 1). Due to time constraints, results over a single run are reported.

B.1.2. ROMAN-EMPIRE

All experiments for Roman Empire were conducted on a Nvidia Tesla A100 GPU with 80GB of memory. We used the Adam optimizer with a learning rate of 0.0005 and trained the models for 3000 epochs using early stopping on the validation accuracy with a patience of 300.

Table 6 describes the configuration of the SAT baseline model used on Roman Empire. Table 7 presents an ablation study for our best-performing enhanced-SAT models.

HYPERPARAMETER	VALUE
GNN TYPE (φ)	PNA
# LAYERS	5
HIDDEN DIMENSIONS	32
FFN HIDDEN DIMENSIONS	64
# ATTENTION HEADS	1
DROPOUT	0.5
SIZE OF SUBGRAPHS (k)	3

Table 6. Configuration of baseline SAT model on Roman Empire

MODEL	MODEL SIZE	ACCURACY (%)	Δ_{ACC}
BASE	115 K	82.09	
BASE + ABS PE	116 K	83.27	+0.18
BASE + NODE PROJ	131 K	85.88	+3.79
BASE + JKN	146 K	83.09	+1.00
BASE + NODE PROJ + JKN	162 K	86.32	+4.23
BASE + NODE PROJ + JKN + GATES	**162 K**	**91.90**	**+9.81**
BASE + DIR	170 K	90.96	+8.87
BASE + NODE PROJ + JKN + DIR	**218 K**	**92.55**	**+10.46**
BASE + NODE PROJ + JKN + DIR + GATES	218 K	91.93	+9.92

Table 7. Ablation study of best SAT-based models on Roman-Empire. The column "Model Size" refers to the number of parameters of the model. The column "Δ_{ACC}" is equal to the difference in percentage accuracy compared to the base SAT model. The label "Abs PE" refers to the addition of a random walk positional encoding to the transformer model (Li et al., 2020; Dwivedi et al., 2022). The label "Node Proj" refers to the insertion of a 3-layer MLP before the transformer model to better project the node features in a latent space. Labels "JKN", "Gates" and "Dir" refer to the addition of jumping knowledge with LSTM aggregators, gradient gating and the use of directed GNNs respectively, in the manner described in Section 3. For gated models, we reported the best-performing architecture from our k-p study (Figure 2). Due to time constraints, results over a single run are reported.

B.2. Homophilic Datasets

B.2.1. CLUSTER

All experiments for Cluster were conducted on a Nvidia Tesla A100 GPU with 80GB of memory. We used the Adam optimizer with a learning rate of 0.0005 and trained the models for 200 epochs using a batch size of 64.

HYPERPARAMETER	VALUE
GNN TYPE (φ)	PNA
# LAYERS	6
HIDDEN DIMENSIONS	48
FFN HIDDEN DIMENSIONS	96
# ATTENTION HEADS	8
DROPOUT	0.2
SIZE OF SUBGRAPHS (k)	3

Table 8. Configuration of baseline SAT model on Cluster

C. On the use of absolute encodings

In many of our experiments, absolute encodings such as the commonly used Random Walk Encoding did not improve the model and in many cases lead to worse accuracy. We assume that they are not well suited to truly heterophilic settings, but further experiments would have to be done to confirm this hypothesis.