

# Massive Activations in Graph Neural Networks: Decoding Attention for Domain-Dependent Interpretability

Lorenzo Bini<sup>1,\*</sup>, Marco Sorbi<sup>2,\*</sup> and Stéphane Marchand-Maillet<sup>1</sup>

<sup>1</sup>Department of Computer Science, University of Geneva, Switzerland

<sup>2</sup>Research Institute for Statistics and Information Science, Centre Universitaire d'Informatique, University of Geneva, Switzerland

## Abstract

Graph Neural Networks (GNNs) have become increasingly popular for effectively modeling graph-structured data, and attention mechanisms have been pivotal in enabling these models to capture complex patterns. In our study, we reveal a critical yet underexplored consequence of integrating attention into edge-featured GNNs: the emergence of Massive Activations (MAs) within attention layers. By developing a novel method for detecting MAs on edge features, we show that these extreme activations are not only activation anomalies but encode domain-relevant signals. Our post-hoc interpretability analysis demonstrates that, in molecular graphs, MAs aggregate predominantly on common bond types (e.g., single and double bonds) while sparing more informative ones (e.g., triple bonds). Furthermore, our ablation studies confirm that MAs can serve as natural attribution indicators, reallocating to less informative edges. Our study assesses various edge-featured attention-based GNN models using benchmark datasets, including ZINC, TOX21, and PROTEINS. Key contributions include (1) establishing the direct link between attention mechanisms and MAs generation in edge-featured GNNs, (2) developing a robust definition and detection method for MAs enabling reliable post-hoc interpretability. Overall, our study reveals the complex interplay between attention mechanisms, edge-featured GNNs model, and MAs emergence, providing crucial insights for relating GNNs internals to domain knowledge.

## Keywords

Graph Neural Networks, Edge-Featured Attention, Massive Activations, Post-Hoc Interpretability, Explainable AI, Domain-Relevant Signals, Molecular Graphs, Attention Mechanisms, Activation Anomalies

## 1. Introduction

Graph Neural Networks (GNNs) have rapidly gained traction in scientific research by effectively modeling complex graph-structured data, demonstrating remarkable success across various high-stakes applications such as bioinformatics [1], social network analysis [2], recommendation systems [3] and molecular biology [4]. In this way, understanding the internal workings of these models is crucial for ensuring their reliability and trustworthiness on such applications. Explainability in GNNs allows researchers and practitioners to identify which nodes and edges influence the model's decisions, thereby facilitating debugging, improving transparency, and building trust in the model's predictions [5]. Central to the recent advancements in GNNs is the integration of attention mechanisms, which enable the models to focus on the most relevant parts of the input graph, thereby enhancing their ability to capture intricate patterns and dependencies.

Despite the substantial progress, the phenomenon of Massive Activations (MAs) [6] within attention layers has not been thoroughly explored in the context of GNNs. MAs, characterized by exceedingly large activation values, can significantly impact the stability and interpretability of neural networks. In

EXPLIMED 2025 - Second Workshop on Explainable Artificial Intelligence for the Medical Domain - 25–30 October 2025, Bologna, Italy

\*Corresponding author.

<sup>†</sup>These authors contributed equally.

✉ lorenzo.bini@unige.ch (L. Bini); marco.sorbi@unige.ch (M. Sorbi); stephane.marchand-maillet@unige.ch (S. Marchand-Maillet)

🌐 <https://lorenzobini4.github.io/> (L. Bini); <https://www.unige.ch/gsem/en/research/institutes/isis/team/phd/marco-sorbi/> (M. Sorbi); <https://dblp.org/pid/32/3825.html> (S. Marchand-Maillet)

🆔 0009-0008-6779-2133 (L. Bini); 0009-0006-9028-3279 (M. Sorbi); 0000-0002-4875-6101 (S. Marchand-Maillet)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

particular, understanding and mitigating MAs in GNNs is crucial for ensuring robust and reliable model behavior, especially when dealing with complex and large-scale graphs.

However, a critical aspect of our approach lies in our deliberate choice to use edge-featured attention GNNs. These models are specifically designed to incorporate additional edge attributes, which are typically domain-specific as chemical bond types in molecular graphs (e.g., ZINC [7] and TOX21 [8, 9]) or spatial and interaction properties in protein graphs (e.g., PROTEINS [10]), into their message-passing frameworks. In doing so, they attend not only to nodes but also to the rich, domain-specific information carried by edges. Conventional attention-based GNNs, such as standard Graph Attention Networks (GATs) [11] and their variants that lack explicit edge-feature attention, fall outside the scope of our analysis. Our choice of models and datasets is driven by the idea that incorporating extra information at the edge-level can fundamentally alter the behavior of the attention mechanism and, consequently, the emergence of MAs.

Our central motivation is to investigate how edge-featured attention mechanisms in graph-based networks generate extreme activation values, termed MAs, which deviate from expected norms. Through empirical and statistical analyses, including the Kolmogorov–Smirnov test [12], we demonstrate that these MAs are not only anomalies but encode domain-relevant signals (details can be found in Appendices B and C). For instance, in molecular graphs, MAs predominantly localize on common bond types (e.g., single/double bonds) rather than informative triple bonds, aligning with chemical intuition and suggesting MAs act as natural attribution indicators to highlight less informative edges. To systematically detect and characterize MAs, we develop a post-hoc interpretability framework linking edge feature integration in attention mechanisms to MA generation, alongside introducing the Explicit Bias Term (EBT) to stabilize activation distributions. Our experiments comprehensively evaluate GNN architectures, GraphTransformer [13], GraphiT [14], and SAN [15], across diverse tasks (graph regression, multi-label classification) to validate the consistency of MAs. By establishing MA identification criteria and conducting ablation studies, we underscore the role of edge features in shaping these activations, thereby offering actionable insights for model interpretation and stabilization. While our current analysis provides a deep characterization of MAs, we remain committed to further exploring additional datasets and configurations in future work.

In summary, our contributions are twofold<sup>1</sup>:

- We provide the first systematic study on MAs in edge-featured attention-based GNNs, highlighting their impact on model interpretability.
- We propose a robust detection methodology for MAs, accompanied by detailed experimental protocols and ablation studies to enable reliable post-hoc interpretability of model attention outputs.

Through this work, we aim to shed light on a critical yet understudied aspect of attention-based GNNs, offering valuable insights for the development of more interpretable graph-based models.

## 2. Related Works

GNNs have emerged as powerful tools for analyzing graph-structured data, with applications in health-care [16], molecular property prediction [17], and computational biology discovery [18]. The evolution of GNNs has seen significant advancements, particularly with the integration of attention mechanisms inspired by transformers in natural language processing. GATs [11] pioneered the use of self-attention in GNNs, enabling nodes to dynamically weigh their neighbors, thereby enhancing the model’s ability to capture complex graph relationships. Subsequent innovations, such as GraphiT [14] and the Structure-Aware Network (SAN) [15], further generalized transformer architectures for graphs and incorporated structural properties, improving performance across tasks.

Recent studies on Large Language Models (LLMs) and Vision Transformers (ViTs) have identified the presence of extreme activation values (MAs) in their attention layers [6], prompting investigations into

<sup>1</sup>The source code is available on GitHub at [github.com/msorbi/gnn-ma](https://github.com/msorbi/gnn-ma)

their implications for model behavior, interpretability, and robustness. While similar phenomena have been observed in ViTs, the study of MAs in GNNs remains underexplored, representing a critical gap in understanding these models.

Broader research on neural network interpretability, such as feature visualization [19] and network dissection [20], offers potential methodologies for analyzing MAs in GNNs. Additionally, insights from attention flow [21] and attention head importance [22] in transformers suggest that not all attention heads contribute equally, raising questions about similar patterns in graph transformers and their relation to MAs. These findings highlight the need for further research into MAs in GNNs to uncover their role, impact, and potential vulnerabilities. The study of internal representations in deep learning models has been a topic of significant interest in the machine learning community. Works such as Bau et al. [23] have explored the interpretability of neural networks by analyzing activation patterns and their relationships to input features and model decisions. However, the specific phenomenon of MAs in GNNs has remained largely unexplored until now, representing a crucial gap in our understanding of these models and their relationships to the domain of the data they process.

### 3. Establishing the Reference

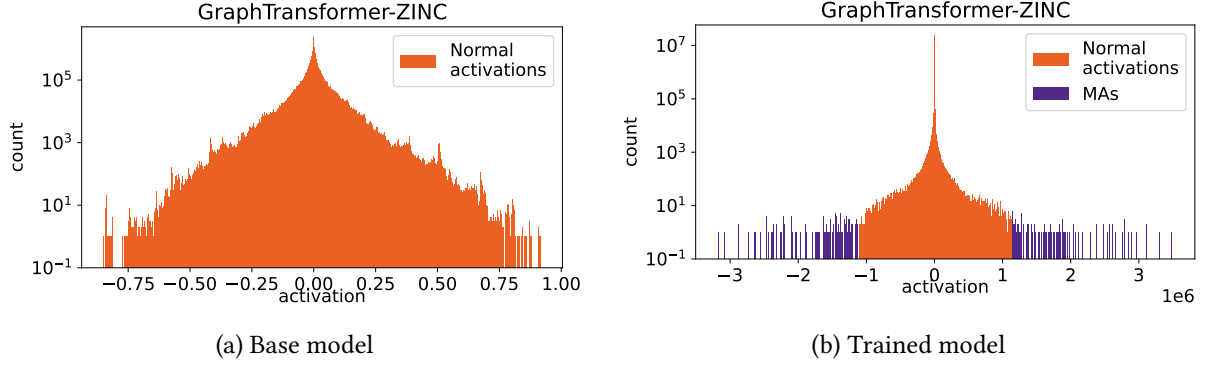
In this section, we detail our approach to analyzing activation distributions in attention-based GNNs, emphasizing a dual perspective: an untrained baseline analysis and a-posteriori observation of a distribution shift in trained models, mapped as outlier activations. We begin by stabilizing a controlled baseline to establish an interpretability reference. This baseline serves as a litmus test for detecting and quantifying deviations and outliers in trained models, as explained in Sections 4 and 5. In their initialized state, attention values follow a symmetric, near-zero distribution (Figure 1a), a consequence of standard weight initialization schemes. This initial behavior embodies our expectations for the model’s internal dynamics before any task-specific training occurs. We start by considering the untrained (base) model, where network parameters are initialized via Xavier initialization. To form a meaningful baseline, we normalize the activation values within each layer. Specifically, for each edge activation, we compute the ratio:

$$\text{ratio}(\text{activation}) = \frac{|\text{activation}|}{\text{median}(|\text{edge activations}|)}. \quad (1)$$

This normalization, dividing by the layer’s edge median, accounts for scale variations across layers and models. To facilitate a meaningful analysis, we apply a logarithmic transformation to the activations ratio (Equation (1)). This transformation exposes the intrinsic shape of the activation distribution, making subtle differences more discernible. As illustrated in Figure 1a, the resulting base distribution is highly peaked, with the majority of values clustered around zero, yet exhibits a long tail for higher values. This sharp peak serves as a robust baseline, reflecting the model’s inherent activation scale before any training-induced changes occur. In this state, the model has not yet learned task-specific features and the activations predominantly reflect the properties of the random initialization.

Our choice is to model the log-transformed base distribution as a Gamma distribution. This decision is motivated by both theoretical and empirical observations. The Gamma distribution, a flexible two-parameter family, is well-suited to capture the skewed, unimodal behavior that arises from the logarithmic transformation of Equation (1). In the untrained (base) model these transformed activation values are well-captured by the Gamma distribution. Empirically, as shown in Figure 3a, our analysis demonstrates that the negative log-transformed activation ratios from the base model align closely with the Gamma approximation. This is validated by a very low Kolmogorov-Smirnov (KS) statistic (approximately 0.020), confirming that the Gamma distribution accurately reflects the statistical properties of the base activations. Thus, both theoretical suitability and strong empirical fit justify the use of the Gamma to model the base activation distribution.

Before delving into the modeling of the distribution shift, it is important to bridge our analysis from the established baseline to the observation of training-induced changes. In the untrained (base) model, as described above, the baseline serves as our reference point for understanding the activation behavior



**Figure 1:** Comparison of activation distributions between base and trained model. The trained model presents two long tails representing MAs values. Notice the  $y$ -axis is log-scaled, making the base model activations distribution clustered around zero.

before any task-specific learning occurs. However, as the model is trained, its internal dynamics evolves significantly, as later shown in Section 4. By comparing the base and trained models in Figure 1, we observe that activation profile exhibits anomalous concentrations on the left and right tails. As depicted in Figure 3, while the Gamma distribution accurately approximates the base activations, it fails to capture the extreme values, i.e. MAs appearing after training (which correspond to left-hand values due to the application of log-transformation). This two-part framework, beginning with an initial baseline and progressing to a post-hoc investigation, ensures our analysis not only captures the behavior of the base model but also offers explainable insights into the modifications induced by training. In Section 4 we introduce the appropriate definitions and terminology for MAs. Then, throughout Section 5 we proceed with the investigation of the training-corrupted distribution and the consequences of the MAs’ emergence.

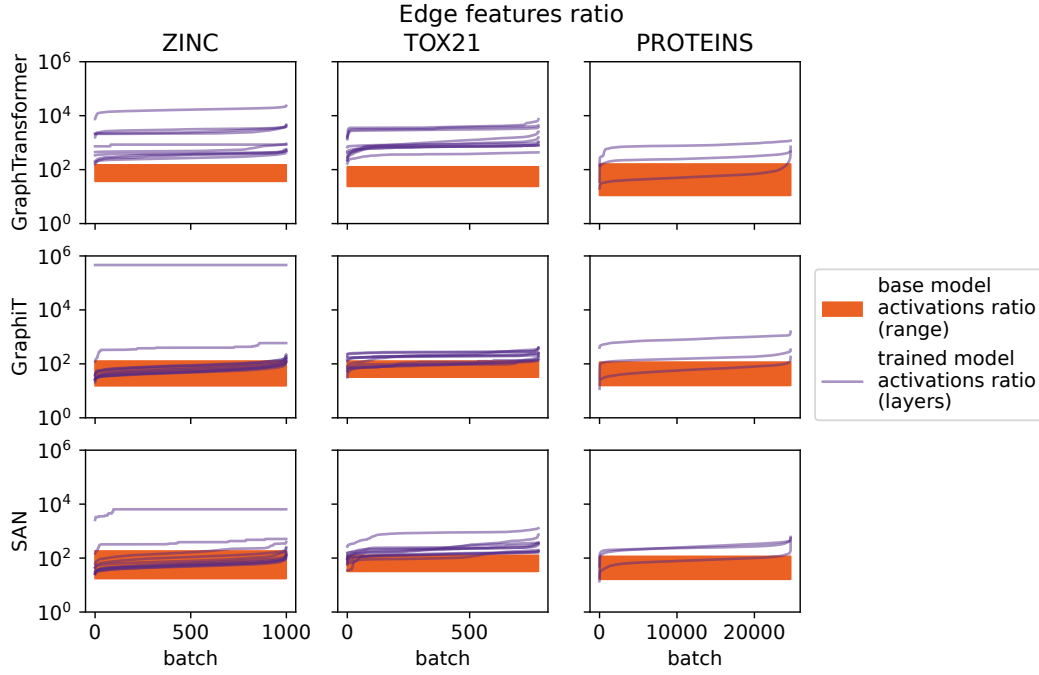
## 4. Terminology of Massive Activations in GNNs

Building upon the work on MAs in LLMs [6], we extend this investigation to edge-featured attention-based GNNs, focusing specifically on graph transformer architectures. Our study encompasses various models, including GraphTransformer (GT) [13], GraphiT [14], and Structure-Aware Network (SAN) [15], applied to diverse task datasets such as ZINC, TOX21, and OGBN-PROTEINS (see Appendices A and D for details on models configurations and datasets composition). This comprehensive approach allows us to examine the generality of MAs across different attention-based GNN architectures.

### 4.1. Characterization of Massive Activations

MAs in GNNs refer to specific activation values that exhibit unusually high magnitudes compared to the typical activations within a layer. These activations are defined by the following criterion, where an activation value is intended to be its absolute value.

**Relative Threshold:** In the paper by Sun et al. [6], MAs were defined as at least 1,000 times larger than the median activation value within the layer. This relative threshold criterion helped differentiate MAs from regular high activations that might occur due to normal variations in the data or model parameters. The formal definition was represented as  $MAs = \{a \mid a > 1000 \times \text{median}(\mathbf{A})\}$ , where  $\mathbf{A}$  represents the set of activation values in a given layer. However, in contrast to previous studies that employed a fixed relative threshold to detect LLMs MAs, our work is intended to characterize their nature within an a-posteriori explainable framework. This investigation ensures a comparative analysis of the GNNs attention activations, where the untrained model serves as a reference to identify emerging outliers.



**Figure 2:** Comparison of MAs on trained against base models. Represented ratios have been sorted increasingly for each layer independently. Orange box represents the range of normal ratios obtained in the base model, while ratios exceeding the base come from MAs.

#### 4.1.1. Detection Procedure

For both base and trained models, we detected MAs following a systematic procedure:

**Normalization:** We normalized the activation values within each layer, dividing them by the edge median on the layer, to account for variations in scale between different layers and models. This normalization step ensures a consistent basis for comparison. Since attention is computed between pairs of adjacent nodes only, in contrast to LLMs where it is computed among each pair of tokens, the model tends to spread MAs among the edges to make them “available” to the whole graph. Indeed, our prior analysis indicates that MAs are a common phenomenon across different models and datasets, that they are not confined to specific layers but are distributed throughout the model architecture, and that MAs are an inherent characteristic of the attention-based mechanism in graph transformers and related architectures, not strictly dependent on the choice of the dataset (see Appendix B for further details, in particular Figure 7).

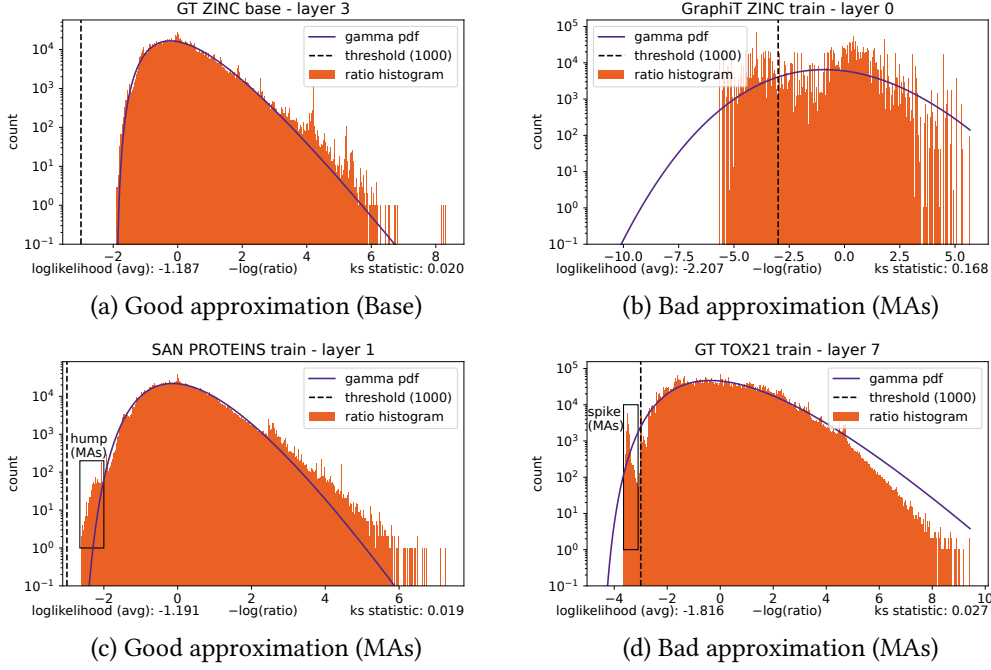
**Batch Analysis:** We analyzed the activations on a batch-by-batch basis, minimizing the batch size, to have suitable isolation between the MAs and to ensure that the detection of MAs is not influenced by outliers in specific samples. For each activation we computed its ratio as in Equation (1), and those exceeding the threshold were flagged as massive. We then considered the maximum ratio of each batch to detect those containing MAs. We performed this analysis across multiple layers to identify patterns and layers that are more prone to exhibiting MAs. This aggregation helps in understanding the hierarchical nature of MAs within the model.

Figure 2 reports the analysis results. The batch ratios significantly increase in the trained transformers, concerning base ones, often even overcoming the threshold of 1000 defined by previous works [6], showing the presence of MAs in graph transformers.

## 5. Methodology and Observation

Focusing on the attention components related to the edge features (see Appendix D.6), first, we analyzed the ratio defined in Equation (1), taking the maximum for every batch, across each model layers, and





**Figure 3:** Activation distributions for base and trained (with MAs) models. In Figure 3d we clearly distinguish a spike on the left of the distribution, corresponding to a ratio of 1000 ( $-\log(\text{ratio}) = -3$ ), which identifies the separation between the basic and massive regimes. The approximation pdf is rescaled to match the histogram scale.

visually compared the outcomes to value ranges obtained using the same model in a base state (with parameters randomly initialized, without training) to verify the appearance of MAs. The graphical comparison, reported in Figure 2, shows ratios over the base range in most of the trained models, representing MAs.

To better characterize MAs, we studied their distribution employing the Kolmogorov-Smirnov statistic [12], as discussed in Appendix C. We found that a gamma distribution well approximates the negative logarithm of the activations’ magnitudes, as well as their ratios. Figure 3a shows this approximation for a base model layer. We point out that, according to the existing definition, items on the left of the  $-3$  are MAs. We compared the distributions of the log-values between the base and trained models, as illustrated in Figure 3, which highlights a significant shift in the trained model’s distribution, confirming the emergence of MAs during training. This shift indicates that the threshold around  $-\log(\text{ratio}) = -3$  (e.g., a ratio of 1000 or higher) effectively captures these significant activations, though it sometimes appears slightly shifted to the right, as shown in Figure 3c.

When MAs appear, two phenomena are observed: either a large number of extreme activation values are added to the left-hand side of the distribution, preventing a good approximation (Figure 3b), or a few values appear as spikes, humps, or out-of-distribution values, which may or may not deteriorate the approximation (Figures 3c and 3d). For instance, Figure 3a represents the base model with untrained weights, where the gamma approximation fits the sample histogram well, evidenced by a low KS statistic of 0.020. In contrast, Figure 3b shows the trained model’s distribution with a significant shift due to a large hump on the left side, representing extreme activation ratios (MAs), resulting in a poor gamma approximation with a KS statistic of 0.168. Similarly, Figure 3d displays a clear spike at  $-\log(\text{ratio}) = -3$  (a ratio of 1000) in the trained model’s distribution, indicating the distinction between basic and massive activation regimes and a poor gamma fit with a KS statistic of 0.027. Finally, Figure 3c shows the trained model’s distribution with a noticeable hump on the left side, indicating MAs. Although the gamma approximation fits better here (KS statistic of 0.019), the presence of MAs is still evident, confirming their addition to the left-hand side of the distribution.

Inspired by recent advancements in addressing bias instability in LLMs [6], we introduced an EBT

into our graph transformer models. This bias term is discovered to counteract the emergence of MAs by stabilizing the activation magnitudes during the attention computation. The EBT is computed as follows:

$$\mathbf{b}_e = Q\mathbf{k}e' \quad (2)$$

$$\mathbf{b}_v = \text{softmax}(\mathbf{A}_e)\mathbf{v}', \quad (3)$$

where  $\mathbf{k}, \mathbf{e}, \mathbf{v} \in \mathbb{R}^d$  are the key, edge, and node bias terms (one per each attention head),  $\mathbf{A}_e$  is the edge attention output, and  $d$  the corresponding hidden dimension.  $\mathbf{b}_e$  and  $\mathbf{b}_v$  represent the edge and node bias terms and are added to the edge and node attention outputs, respectively. By incorporating EBT into the edge and node attention computations, and adding bias in the linear projections of the attention inputs, we regulated the distribution of activation values, thus mitigating the occurrence of MAs. Further details on the MA detection procedure and EBT’s impact are available in Appendix B.

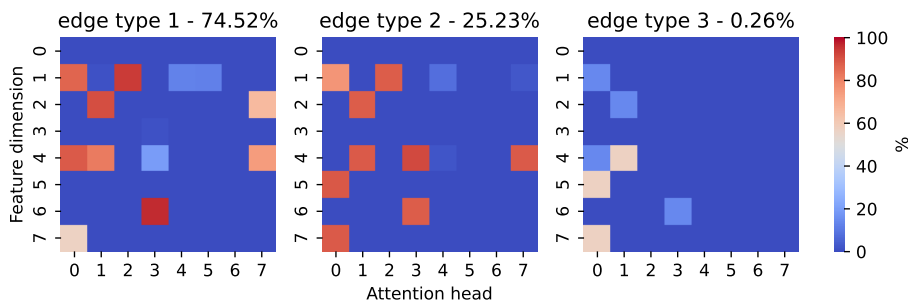
In the next section, we delve into the interpretability of edge-related MAs, demonstrating how their emergence provides insights into the model’s attention allocation. By analyzing MAs in relation to domain-specific edge features, we reveal their role as natural attribution indicators. This investigation highlights how MAs can be leveraged to understand and refine graph transformer models, improving their interpretability and facilitating their use in scientific discovery.

## 6. Interpretability of Edge-Related Massive Activation

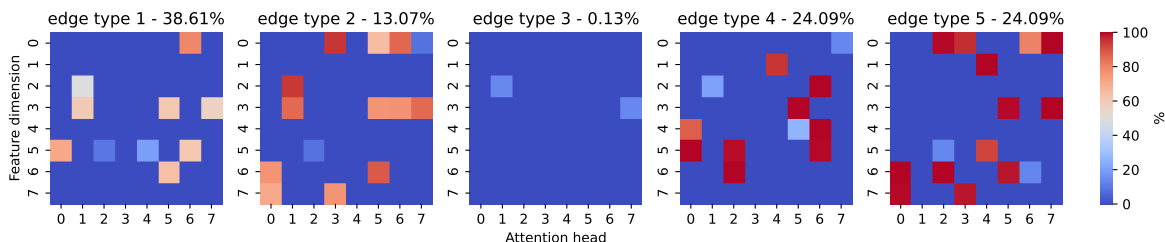
The emergence of MAs raises critical questions about *why* and *where* these outliers occur in graph structures. In the context of molecule graphs, we analyze MAs through the lens of edge types, a human-interpretable graph feature, and quantify their role in driving model behavior. We employ edge type-wise activation heatmap to localize MAs within the graph topology. In the ZINC dataset, edge types represent different types of chemical bonds between atoms in a molecule, specifically edge type 1 corresponds to a single bond (e.g., C–H), edge type 2 represents a double bond (e.g., C=O), and edge type 3 indicates a triple bond (e.g., C≡C). Triple bonds are less common but highly significant in certain chemical contexts. For each edge type, we explain the model’s attention output through a heatmap (Figures 4 and 5), where we visualize MAs per attention head and hidden feature dimension. Specifically, each cell in the heatmap represents the percentage of edges having one MA in that position. For example, Figure 5 heatmap with edge type 5, shows that at position (7, 0) 100% of edges have one MA each on that location. Although it appears to be no regularity on their locations, Figure 4 reveals a distinguished pattern: MAs are aggregated on edge types 1 and 2, and rare on type 3. This observation provides several critical insights into the model’s internal behavior:

- The aggregation of MAs on edge types 1 and 2 indicates that the model has a particular regard for most rare edges type.
- Under normal conditions, without the influence of MAs, the activation values on each edge would depend on the “token” contextual information. However, the presence of MAs introduces extreme values that overwrite these domain-dependent signals.
- In accordance with Shannon information [24], a higher frequency of occurrence is generally associated with lower per-instance information content, as the information becomes more diffusely distributed. Broadly, given an event  $x$  with probability  $P$ , the information content is defined as  $I(X) := -\log_2[\Pr(x)] = -\log_2(P)$ . In this way, type 3 edges (less frequent) are most informative ones.
- The model appears to have learned to identify less informative edges and exploit them to allocate MAs, thereby leaving unmodified original domain information on critical edges.

These insights suggest MAs can serve as edge importance indicator to retrieve domain-relevant information. For instance, in self-supervised/contrastive learning scenarios, rather than solely relying on hand-crafted augmentations (which may be suboptimal for certain tasks) one could design augmentation



**Figure 4:** Heatmaps showing MAs concentration across the three edge types in the ZINC dataset. Each heatmap visualizes the percentage of edges with MAs per attention head and hidden feature dimension. Notably, MAs predominantly aggregate on edge types 1 and 2, while being rare on type 3, indicating the model’s tendency to allocate MAs to more frequent edges. Type 3 consists of 0.26% of edges in the dataset.



**Figure 5:** Ablation studies show the reallocation of MAs on the chemically meaningless edges (types 4 and 5), designed to carry low intrinsic information. This supports the hypothesis that MAs can serve as indicators of edge importance.

strategies leveraging MAs as indicators. Leveraging these indicators can be beneficial for downstream tasks, where identifying critical edges, those that significantly influence the model’s performance, is essential for creating meaningful augmentations. Measures like link entropy [25] and graph cuts [26] can be employed to assess the importance of edges [27], guided by MAs as indicators for deploying augmentation strategies to improve learning efficiency.

It is important to clarify that the significance of an edge is not uniquely determined by its type; rather, it depends on contextual information and graph structure as well [28]. For our current analysis, however, we have focused on investigating the relationship between edge type and MAs presence.

### 6.1. Ablation Studies on the Interpretability of Edge-Related MAs

To further investigate our use of MAs as indicators of less informative edges, we conducted an ablation study designed to decouple chemical informativeness from edge frequency. In our experiment, for each molecule in the dataset we introduced a global dummy node that connects to all other atoms. This connection is established through two new types of edges: type 4 for incoming connections to the dummy node and type 5 for outgoing connections. As a result, while the most frequent edge type (i.e., single chemical bond) remains type 1, the newly introduced edges (types 4 and 5) are intentionally meaningless from a chemical standpoint and thus represent edges with very low intrinsic information content. This controlled setup allows us to clearly observe that the network, once retrained, reallocates MAs towards dummy edges (types 4 and 5) designed to be less informative, as shown in Figure 5. This reallocation confirms our hypothesis that MAs serve as markers for edges carrying lower domain-specific information content. Such findings suggest that MAs could be exploited as indicators of edge importance to guide downstream tasks.



## 7. Conclusion and Future Work

In this work, we have presented the first study of MAs in edge-featured attention-based GNNs. Our novel methodology for detecting and analyzing MAs, supported by ablation studies, has demonstrated that these extreme activations are not model artifacts but can be linked with edge importance. By establishing a robust framework for post-hoc interpretability, we have shown that MAs provide valuable insights into how attention mechanisms allocate importance across edges, revealing, for example, that common bond types in molecular graphs tend to accumulate these activations while more informative bonds remain relatively unaltered. This work thus not only deepens our understanding of the internal mechanisms of edge-featured attention GNNs but also sets the stage for their application in extracting actionable scientific insights. Furthermore, our investigation highlights the role of EBT in stabilizing activation distributions.

Looking forward, our future work will expand this interpretability framework across a broader range of architectures and datasets. We aim to further explore how MAs patterns can be systematically exploited to improve model transparency and guide the design of data-adaptive strategies for downstream tasks such as link prediction, drug design, and self-supervised learning. By investigating how measures like edge entropy relate to MAs distribution, we plan to refine augmentation and feature re-weighting techniques that enhance both model performance and interpretability.

In summary, our study provides a key step towards developing more transparent and interpretable graph-based models. By addressing the challenges posed by MAs and leveraging them as natural attribution indicators, we aim to bridge the gap between complex neural network internals and domain-specific scientific discovery.

## Acknowledgments

The Swiss National Science Foundation partially funds this work under grants number 207509 “Structural Intrinsic Dimensionality”, and 215733 “Une édition sémantique et multilingue en ligne des registres du Conseil de Genève (1545-1550)”.

## Declaration on Generative AI

During the preparation of this work, the author(s) used Grammarly in order to: Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication’s content.

## References

- [1] X.-M. Zhang, L. Liang, L. Liu, M.-J. Tang, Graph neural networks and their current applications in bioinformatics, *Frontiers in genetics* 12 (2021) 690049.
- [2] S. Min, Z. Gao, J. Peng, L. Wang, K. Qin, B. Fang, Stgsn—a spatial-temporal graph neural network framework for time-evolving social networks, *Knowledge-Based Systems* 214 (2021) 106746.
- [3] C. Gao, X. Wang, X. He, Y. Li, Graph neural networks for recommender system, in: *Proceedings of the fifteenth ACM international conference on web search and data mining*, 2022, pp. 1623–1625.
- [4] H. Cai, H. Zhang, D. Zhao, J. Wu, L. Wang, Fp-gnn: a versatile deep learning architecture for enhanced molecular property prediction, *Briefings in bioinformatics* 23 (2022) bbac408.
- [5] H. Yuan, H. Yu, S. Gui, S. Ji, Explainability in graph neural networks: A taxonomic survey, *IEEE transactions on pattern analysis and machine intelligence* 45 (2022) 5782–5799.
- [6] M. Sun, X. Chen, J. Z. Kolter, Z. Liu, Massive activations in large language models, 2024. URL: <https://arxiv.org/abs/2402.17762>. arXiv: 2402.17762.
- [7] J. J. Irwin, T. Sterling, M. M. Mysinger, E. S. Bolstad, R. G. Coleman, Zinc: a free tool to discover chemistry for biology, *Journal of chemical information and modeling* 52 (2012) 1757–1768.

- [8] A. Mayr, G. Klambauer, T. Unterthiner, S. Hochreiter, Deeptox: toxicity prediction using deep learning, *Frontiers in Environmental Science* 3 (2016) 80.
- [9] R. Huang, M. Xia, D.-T. Nguyen, T. Zhao, S. Sakamuru, J. Zhao, S. A. Shahane, A. Rossoshek, A. Simonov, Tox21challenge to build predictive models of nuclear receptor and stress response pathways as mediated by exposure to environmental chemicals and drugs, *Frontiers in Environmental Science* 3 (2016) 85.
- [10] 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, *Advances in neural information processing systems* 33 (2020) 22118–22133.
- [11] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, *arXiv preprint arXiv:1710.10903* (2017).
- [12] I. M. Chakravarti, R. G. Laha, J. Roy, *Handbook of methods of applied statistics*, Wiley Series in Probability and Mathematical Statistics (USA) eng (1967).
- [13] V. P. Dwivedi, X. Bresson, A generalization of transformer networks to graphs, 2021. URL: <https://arxiv.org/abs/2012.09699>. *arXiv:2012.09699*.
- [14] G. Mialon, D. Chen, M. Selosse, J. Mairal, Graphit: Encoding graph structure in transformers, *arXiv preprint arXiv:2106.05667* (2021).
- [15] D. Kreuzer, D. Beaini, W. Hamilton, V. Létourneau, P. Tossou, Rethinking graph transformers with spectral attention, *Advances in Neural Information Processing Systems* 34 (2021) 21618–21629.
- [16] S. G. Paul, A. Saha, M. Z. Hasan, S. R. H. Noori, A. Moustafa, A systematic review of graph neural network in healthcare-based applications: Recent advances, trends, and future directions, *IEEE Access* (2024).
- [17] O. Wieder, S. Kohlbacher, M. Kuenemann, A. Garon, P. Ducrot, T. Seidel, T. Langer, A compact review of molecular property prediction with graph neural networks, *Drug Discovery Today: Technologies* 37 (2020) 1–12.
- [18] L. Bini, F. N. Mojarrad, M. Liarou, T. Matthes, S. Marchand-Maillet, Flowcyt: A comparative study of deep learning approaches for multi-class classification in flow cytometry benchmarking, *arXiv preprint arXiv:2403.00024* (2024).
- [19] C. Olah, A. Mordvintsev, L. Schubert, Feature visualization, *Distill* 2 (2017) e7.
- [20] D. Bau, B. Zhou, A. Khosla, A. Oliva, A. Torralba, Network dissection: Quantifying interpretability of deep visual representations, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6541–6549.
- [21] S. Abnar, W. Zuidema, Quantifying attention flow in transformers, *arXiv preprint arXiv:2005.00928* (2020).
- [22] P. Michel, O. Levy, G. Neubig, Are sixteen heads really better than one?, *Advances in neural information processing systems* 32 (2019).
- [23] D. Bau, J.-Y. Zhu, H. Strobel, A. Lapedriza, B. Zhou, A. Torralba, Understanding the role of individual units in a deep neural network, *Proceedings of the National Academy of Sciences* 117 (2020) 30071–30078.
- [24] C. E. Shannon, A mathematical theory of communication, *The Bell system technical journal* 27 (1948) 379–423.
- [25] M. Dehmer, A. Mowshowitz, A history of graph entropy measures, *Information Sciences* 181 (2011) 57–78.
- [26] H. Shin, J. Park, D. Kang, A graph-cut-based approach to community detection in networks, *Applied Sciences* 12 (2022) 6218.
- [27] Y. Qian, Y. Li, M. Zhang, G. Ma, F. Lu, Quantifying edge significance on maintaining global connectivity, *Scientific reports* 7 (2017) 45380.
- [28] K. R. Žalik, M. Žalik, Density-based entropy centrality for community detection in complex networks, *Entropy* 25 (2023) 1196.

## A. Dataset Composition

This section provides additional details on the used datasets throughout the experiments.

The ZINC dataset [7] is a benchmark collection for evaluating GNNs in molecular chemistry, where molecules are represented as graphs with atoms as nodes and chemical bonds as edges. Contents include:

- **Graphs:** The dataset includes over 250,000 molecular graphs. Each molecule is represented by a graph with nodes (atoms) and edges (bonds), incorporating various bond types (e.g., single, double, triple).
- **Node Features:** Atoms are described by features that capture their chemical properties, such as atom types, hybridization states, and other atomic attributes.
- **Edge Features:** Bonds between atoms are characterized by features representing bond types and additional chemical information.
- **Task:** The primary task is **graph regression**, where the goal is to predict continuous values associated with each molecule. This often involves predicting molecular properties such as solubility or biological activity.

ZINC [7] is useful for evaluating GNNs' performance in learning molecular representations and predicting continuous chemical properties, providing insights into the model's ability to generalize across diverse chemical compounds.

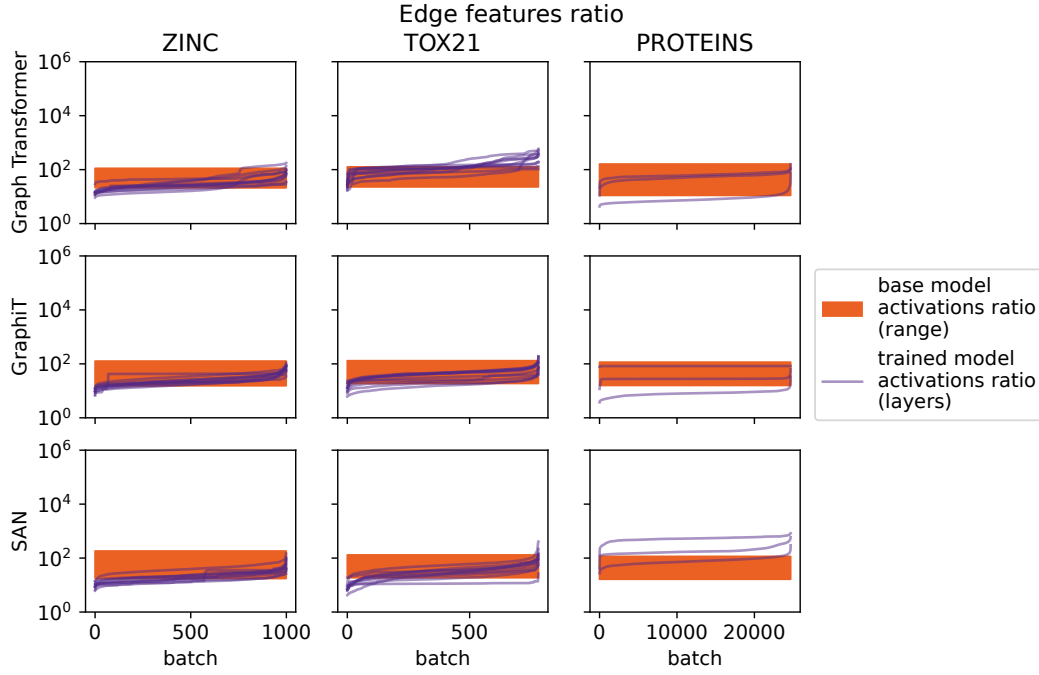
The TOX21 dataset [8, 9] is designed for toxicity prediction and focuses on classifying chemical compounds based on their potential toxicity. It is part of the Toxicology Data Challenge and features molecular graphs with associated toxicity labels. Contents include:

- **Graphs:** The dataset consists of molecular graphs where nodes represent atoms and edges represent chemical bonds. It includes thousands of molecules with toxicity annotations, and it consists of 7,831 graphs with each graph representing a molecular structure with associated toxicity labels.
- **Node Features:** Atoms are encoded with features representing their types, hybridization states, and other chemical properties.
- **Edge Features:** Bonds are detailed with features indicating bond types and additional chemical attributes.
- **Task:** The main task is **multi-label graph classification**, where each molecule is classified into multiple toxicity categories. This allows for the prediction of various toxicity endpoints simultaneously.

TOX21 [8, 9] is valuable for assessing GNN models in predicting toxicity from molecular structures, which is crucial for drug discovery and safety evaluation, providing a benchmark for multi-label classification tasks.

The OGBN-PROTEINS dataset, part of the Open Graph Benchmark (OGB) [10], focuses on protein function prediction. It contains one large graph representing protein structures, with nodes corresponding to amino acids and edges to their interactions. Contents include:

- **One Large Graph:** OGBN-PROTEINS contains 54,879 nodes and 89,724 edges. These nodes represent amino acids in protein structures, and edges represent interactions or bonds between these amino acids. It includes various protein structures used for functional prediction.
- **Node Features:** Amino acids are described by features capturing biochemical properties, such as amino acid type, secondary structure, and other relevant attributes.
- **Edge Features:** Edges denote interactions between amino acids and include features reflecting the nature of these interactions or spatial relationships.
- **Task:** The task is **multi-label node classification**, where the goal is to predict multiple functional categories for each amino acid node in the protein graph. This involves classifying nodes into various functional classes based on their role in the protein's functionality.



**Figure 6:** Comparison of MAs on trained against base models, with the use of Explicit Bias Term. Represented ratios have been sorted increasingly for each layer independently.

OGBN-PROTEINS [10] is suitable for evaluating GNNs on biological data, specifically in predicting protein functions based on structural information. It provides insights into how well models can handle multi-label node classification tasks in a complex biological context.

## B. Further Discussion on MAs Detection Procedure

The analysis presented in Section 5 highlights key insights into the emergence and distribution of MAs in edge-featured attention-based GNNs. As illustrated in Figures 2 and 7, distinct patterns emerge across datasets and model architectures, revealing the interplay between attention mechanisms, dataset characteristics, and learned biases. Below, we summarize the main findings drawn from our evaluation.

### 1. Dataset Influence:

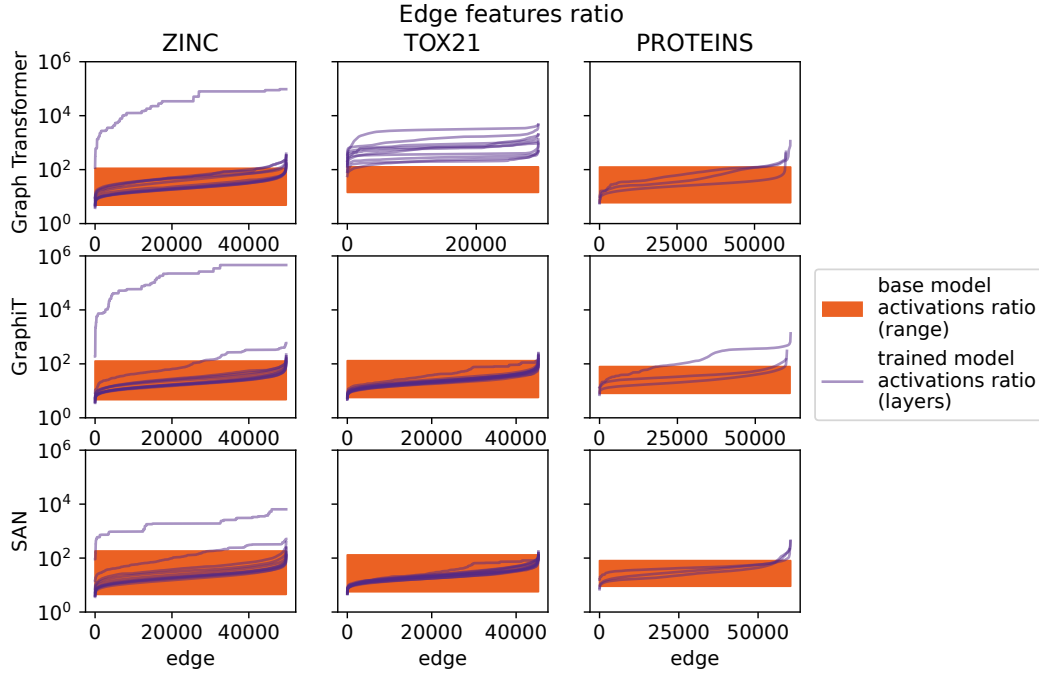
- The ZINC and OGBN-PROTEINS datasets consistently show higher activation values across all models compared to TOX21, suggesting that the nature of these datasets significantly influences the emergence of MAs. Even though many MAs are emerging from GT on TOX21.

### 2. Model Architecture:

- Different GNN models exhibit varying levels of MAs. For instance, GraphTransformer and GraphiT tend to show more pronounced MAs than SAN, indicating that model architecture plays a crucial role.

### 3. Impact of Attention Bias:

- Previous works suspect that MAs have the function of learned bias, showing that they disappear introducing bias at the attention layer. This holds for LLMs and ViTs, and for our GNNs as well, as shown in Figure 2 where the presence of MAs is affected by the introduction of the Explicit Bias Term on the attention. Figure 6 and text below suggest that MAs are intrinsic to the models' functioning, being anti-correlated with the learned bias.



**Figure 7:** Comparison of MAs for trained vs base models, along all the edges. Activation values have been normalized within each layer by the layer’s edge median. Represented ratios have been sorted increasingly for each layer independently.

The consistent observation of MAs in edge features, across various GNN models and datasets, points to a fundamental characteristic of how these models process relational information. Table 1 shows that EBT does not systematically influence the test loss equally across different models and datasets. We have considered the test loss metric to keep the approach general, making it extendable to different downstream tasks. This ensures that the proposed method can be applied broadly across various applications of graph transformers.

Although the test loss remains relatively unchanged with the introduction of EBT, its presence helps in mitigating the occurrence of MAs, as evidenced by the reduction in extreme activation values observed in earlier figures. By analyzing these results, it becomes evident that while EBT does not drastically alter the test performance, it plays a crucial role in controlling activation anomalies, thereby contributing to the robustness and reliability of graph transformer models.

As illustrated in Figure 6, the introduction of EBT leads to a substantial reduction in both the frequency and magnitude of MAs, aligning activation ratios more closely with those seen in the base models. This stabilization effect is consistently observed across all datasets, ZINC, TOX21, and OGBN-PROTEINS, demonstrating that EBT effectively regulates activation distributions, bringing them closer to the expected reference behavior of untrained models. This consistency underscores the general applicability of EBT in various contexts and downstream tasks. Moreover, Figure 6 shows that EBT mitigates MAs across different layers of the models. This is crucial as it indicates that EBT’s effect is not limited to specific parts of the network but is extended throughout the entire architecture. For example, GraphTransformer on ZINC without EBT shows MAs frequently exceed  $10^4$ , while when EBT has been applied these ratios are significantly reduced, aligning more closely with the base model’s range.

### C. Kolmogorov-Smirnov Test

This section provides additional details on the Kolmogorv-Smirnov (KS) test [12] used to analyze the distribution of activations. The KS test is a non-parametric test that compares the cumulative distribution functions of two samples. It is used to compare a sample with a reference probability

**Table 1**

Comparison of test loss with and w/o bias for the different models and datasets. In bold the worst performances.

Dataset	Model	Test loss	Test loss (EBT)
ZINC	GraphTransformer	0.26	<b>0.29</b>
	GraphiT	0.13	<b>0.31</b>
	SAN	0.18	<b>0.27</b>
TOX21	GraphTransformer	0.25	<b>0.29</b>
	GraphiT	<b>0.38</b>	0.32
	SAN	<b>0.38</b>	0.31
OGBN-PROTEINS	GraphTransformer	<b>0.13</b>	0.12
	GraphiT	0.14	<b>0.16</b>
	SAN	0.13	0.13

distribution (one-sample KS test) or to compare two samples (two-sample KS test) with each other. We primarily used the one-sample KS test to assess the goodness of fit between our observed activation distributions and a theoretical gamma distribution.

In our study, we utilized the KS statistic to compare the distribution of activation values before and after training (i.e. base against trained model), identifying MAs. Xavier initialization was chosen due to its well-established ability to maintain stable activation distributions throughout deep networks, reducing the risk of vanishing or exploding gradients. As shown in Figure 1, the distribution observed in the untrained model is the closest approximation to a Delta function among all cases, with activations concentrated around their expected mean (zero). This serves as a crucial reference for assessing how training and the emergence of MAs alter the model’s internal behavior. Once training begins, learned weights and attention mechanisms introduce deviations from this distribution.

### C.1. One-Sample Kolmogorov-Smirnov Test

The one-sample KS test can typically be formulated as follows:

#### C.1.1. Null Hypothesis

The null hypothesis for the one-sample KS test is:

$H_0$ : The sample data follows the specified distribution (in our case, a gamma distribution).

#### C.1.2. Test Statistic

The KS statistic  $D_n$  is defined as the supremum of the absolute difference between the empirical cumulative distribution function (ECDF)  $F_n(x)$  of the sample and the cumulative distribution function (CDF)  $F(x)$  of the reference distribution:

$$D_n = \sup_x |F_n(x) - F(x)| \quad (4)$$

where  $\sup_x$  denotes the supremum of the set of distances.

#### C.1.3. Empirical Cumulative Distribution Function

For a given sample  $x_1, x_2, \dots, x_n$ , the ECDF is defined as:

$$F_n(x) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{x_i \leq x} \quad (5)$$



where  $\mathbf{1}_{x_i \leq x}$  is the indicator function, equal to 1 if  $x_i \leq x$  and 0 otherwise.

#### C.1.4. Critical Values and p-value

The distribution of the KS test statistic under the null hypothesis can be calculated, which allows us to obtain critical values and p-values. The null hypothesis is rejected if the test statistic  $D_n$  is greater than the critical value at a chosen significance level  $\alpha$ , or equivalently if the p-value is less than  $\alpha$ .

### C.2. Application to MAs Detection

In our experiments, we used the KS statistic to assess whether the distribution of activation ratios in our GNNs follows a gamma distribution. The process is as follows:

1. We computed the activation ratios for each layer of our models, as defined in Equation (1) of the main paper.
2. We took the negative logarithm of these ratios to transform the distribution.
3. We fit a gamma distribution to this transformed data using maximum likelihood estimation.
4. We performed a one-sample KS test to compare our sample data to the fitted gamma distribution.

The KS test statistic provides a measure of the discrepancy between the observed distribution of activation ratios and the theoretical gamma distribution. A lower KS statistic indicates a better fit, suggesting that the activation ratios more closely follow the expected distribution.

### C.3. Interpretation in the Context of MAs

Following the described procedure in Section C.2, we employed the KS statistic as quantitative/statistical measure to detect the presence of MAs:

- For untrained (base) models, we typically observed low KS statistics, indicating that the activation ratios closely follow a gamma distribution.
- For trained models exhibiting MAs, we often saw higher KS statistics. This indicates a departure from the gamma distribution, which we interpret as evidence of MAs.
- The magnitude of the KS statistic provided a quantitative measure of how significantly the presence of MAs distorts the expected distribution of activation ratios.

Moreover, we complemented our KS statistic results with visual inspections of the distributions and other analyses as described in the main paper.

## D. Model Architecture

This section provides additional details on the models' architecture used throughout all the experiments, namely GT [13], GraphiT [14] and SAN [15]. These graph-transformer architectures integrate the principles of both GNNs and transformers, leveraging the strengths of attention mechanisms to capture intricate relationships within graph-structured data. Graph transformers extend the transformer structure, typically used for sequence data, to graphs, operating by embedding nodes and edges into higher-dimensional spaces and then applying multi-head self-attention mechanisms to capture dependencies between nodes.

Mathematically, let  $\mathcal{G} = (V, E)$  be a graph where  $V = \{v_1, \dots, v_n\}$  is the set of nodes and  $E \subseteq V \times V$  is the set of edges. Each node  $v_i$  is associated with a feature vector  $\mathbf{x}_i \in \mathbb{R}^d$ , and each edge  $(v_i, v_j)$  may have an edge feature  $\mathbf{e}_{ij} \in \mathbb{R}^k$ . Therefore, graph transformer models are designed as follows.

### D.1. Input Embedding

The initial node features  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$  are typically projected to a higher-dimensional space:

$$\mathbf{H}^{(0)} = \mathbf{X}\mathbf{W}_{in} + \mathbf{b}_{in} \quad (6)$$

where  $\mathbf{W}_{in} \in \mathbb{R}^{d \times d'}$  is a learnable weight matrix and  $\mathbf{b}_{in} \in \mathbb{R}^{d'}$  is a bias vector.

### D.2. Positional Encoding

To capture structural information, positional encodings  $\mathbf{P} \in \mathbb{R}^{n \times d'}$  are often added:

$$\mathbf{H}^{(0)} = \mathbf{H}^{(0)} + \mathbf{P} \quad (7)$$

### D.3. Multi-Head Attention Layer

The core of a graph transformer is the multi-head attention mechanism. For each attention head  $i$  (out of  $h$  heads) there are also:

1. Query, Key, and Value Projections:

$$\mathbf{Q}_i = \mathbf{H}^{(l)}\mathbf{W}_i^Q \quad (8)$$

$$\mathbf{K}_i = \mathbf{H}^{(l)}\mathbf{W}_i^K \quad (9)$$

$$\mathbf{V}_i = \mathbf{H}^{(l)}\mathbf{W}_i^V \quad (10)$$

where  $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V \in \mathbb{R}^{d' \times d_k}$  are learnable weight matrices, and  $d_k = d' / h$ .

2. Attentions Scores (node features only):

$$\mathbf{A}_i = \text{softmax}\left(\frac{\mathbf{Q}_i\mathbf{K}_i^T}{\sqrt{d_k}} + \mathbf{M}\right), \quad (11)$$

where  $\mathbf{M} \in \mathbb{R}^{n \times n}$  is a mask matrix to enforce the graph structure:

$$M_{i,j} = \begin{cases} 0 & \text{if } (v_i, v_j) \in E \text{ or } i = j \\ -\infty & \text{otherwise.} \end{cases} \quad (12)$$

3. Output of each head:

$$\mathbf{head}_i = \mathbf{A}_i\mathbf{V}_i. \quad (13)$$

4. Concatenation and Projection:

$$\mathbf{H}' = \text{Concat}(\mathbf{head}_1, \dots, \mathbf{head}_h)\mathbf{W}^O, \quad (14)$$

where  $\mathbf{W}^O \in \mathbb{R}^{d' \times d'}$  is a learnable weight matrix.

### D.4. Feed-Forward Network (FFN)

Each attention layer is typically followed by a position-wise feed-forward network:

$$\text{FFN}(\mathbf{x}) = \max(0, \mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \quad (15)$$

where  $\mathbf{W}_1 \in \mathbb{R}^{d' \times d_{ff}}$ ,  $\mathbf{W}_2 \in \mathbb{R}^{d_{ff} \times d'}$ ,  $\mathbf{b}_1 \in \mathbb{R}^{d_{ff}}$ , and  $\mathbf{b}_2 \in \mathbb{R}^{d'}$  are learnable parameters.

## D.5. Layer Normalization and Residual Connections

Each sub-layer (attention and FFN) employs a residual connection followed by layer normalization:

$$\mathbf{H}^{(l+1)} = \text{LayerNorm}(\mathbf{H}^{(l)} + \text{Sublayer}(\mathbf{H}^{(l)})) \quad (16)$$

where Sublayer is either the multi-head attention or the FFN.

## D.6. Edge Feature Integration

GraphTransformer, GraphiT and SAN incorporate edge features:

1. In attention computation:

$$A_{i,j} = \text{softmax}\left(\frac{\mathbf{q}_i^T \mathbf{k}_j + f(\mathbf{e}_{ij})}{\sqrt{d_k}}\right) \quad (17)$$

where  $f$  is a learnable function (e.g., a small neural network) that projects edge features.

2. In value computation:

$$\mathbf{v}_{ij} = \mathbf{V}_i + g(\mathbf{e}_{ij}) \quad (18)$$

where  $g$  is another learnable function.

## D.7. Global Node

Some architectures introduce a global node  $v_g$  connected to all other nodes to capture graph-level information:

$$\mathbf{h}_g^{(l+1)} = \text{Attention}(\mathbf{h}_g^{(l)}, \mathbf{H}^{(l)}) \quad (19)$$

## D.8. Output Layer

The final layer depends on the task:

- For node classification:  $\mathbf{y}_{node} = \text{softmax}(\mathbf{H}_{node}^{(L)} \mathbf{W}_{out} + \mathbf{b}_{out})$
- For graph classification:  $\mathbf{Y}_{graph} = \text{MLP}(\text{Pool}(\mathbf{H}^{(L)}))$

where Pool is a pooling operation (e.g., mean, sum, or attention-based pooling) to switch from single node to graph embedding level.

## D.9. Training

The model is typically trained end-to-end using backpropagation to minimize a task-specific loss function, such as cross-entropy for classification or mean squared error for regression.

## E. Online Resources

The source code is available on GitHub at [github.com/msorbi/gnn-ma](https://github.com/msorbi/gnn-ma)