# NORMA: Subgraph Discovery via Representation Norm Maximization

**Xiangyu Fu**[1] **Wei Liu**[1,*] **Jun Wang**[2] **Yang Qiu**[3] **Yuhua Li**[1] **Ruixuan Li**[1]
[1]School of Computer Science and Technology, Huazhong University of Science and Technology
[2]iWudao Tech
[3]School of Automation Science and Engineering, South China University of Technology
{xy_fu,idc_lw,idcliyuhua,rxli}@hust.edu.cn
jwang@iwudao.tech, anders@scut.edu.cn
[*]Corresponding author: idc_lw@hust.edu.cn

## Abstract

Identifying explanatory subgraphs is a central task in interpreting graph neural networks (GNNs). Many post-hoc explainers optimize a cross-entropy surrogate of mutual information, which can provide weak guidance when predictive evidence depends on combinations of nodes or edges. In particular, partially recovering an informative motif may leave the prediction probability nearly unchanged. We propose NORMA, a parameterized explainer that guides subgraph selection by maximizing the $\ell_2$ norm of the subgraph representation. Our approach is motivated by feature–weight alignment: features utilized by a trained GNN tend to align with directions amplified by its weights, yielding larger representation norms. We provide a theoretical characterization of this norm-growth mechanism and examine it empirically. Experiments on six benchmark datasets, together with controlled comparisons of objectives and mask parameterizations, show that NORMA effectively recovers ground-truth explanatory structures and provides more reliable optimization than its cross-entropy counterpart.[1]

## 1 Introduction

Owing to their superior performance, Graph Neural Networks (GNNs) (Kipf and Welling, 2017; Xu et al., 2019) have become indispensable tools for analyzing graph-structured data across various domains, including social networks (Xie et al., 2022), molecular chemistry (Gilmer et al., 2017), and natural language processing (Wu et al., 2023b; Schlichtkrull et al., 2022). However, GNNs are often regarded as black-box models, making their predictions difficult to interpret and trust, especially in high-stakes applications (Yuan et al., 2022; Chen et al., 2023). This lack of explainability poses serious challenges in scenarios demanding transparency and accountability, motivating extensive research into dedicated methods for explaining GNNs.

GNN explanation commonly assumes that a prediction is determined by a small, decisive subgraph, while the remaining graph has negligible influence (Ying et al., 2019; Luo et al., 2020). Given a trained GNN $f$ and an input graph $G$, the goal is therefore to identify a compact subgraph $G_s \subseteq G$, defined over nodes or edges, that explains the prediction $\hat{Y} = f(G)$. A widely used formulation searches for such a size-constrained subgraph by maximizing its mutual information with the prediction, i.e., $I(G_s, \hat{Y})$, known as Maximizing Mutual Information (MMI) (Schlichtkrull et al., 2022; Miao et al., 2022; Wang et al., 2021).

[1]Code available at https://github.com/just-monica/NORMA.git.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

However, the standard MMI formulation can be difficult to optimize when graph evidence is non-additive. In general, $I(G_s, \hat{Y})$ cannot be decomposed into independent, per-component contributions, i.e., it is not equal to $\sum_{u \in G_s} I(u, \hat{Y})$ (we present more details in Appendix B, with a concrete example in Section 3.2). In GNN explanations, such non-additivity often appears as a synergistic effect, where predictive evidence emerges only when multiple components are selected jointly as a motif. Since exact MI is generally intractable, MMI-based explainers typically optimize a cross-entropy surrogate such as $-\log P(\hat{Y} \mid G_s)$. This surrogate can inherit the threshold-like behavior induced by non-additive evidence: before enough components are assembled into an informative pattern, the model posterior may change only marginally. Consequently, the surrogate loss may remain flat or weakly informative even as key nodes or edges are gradually recovered, providing limited guidance for subgraph exploration and causing optimization to stall at poor local optima.

We need a criterion that tracks how much key evidence has been recovered during optimization. Our motivation comes from feature–weight alignment: neural networks operate through matrix multiplications, where features aligned with directions emphasized by learned weights are preferentially amplified, while weakly aligned features are largely suppressed (Imani et al., 2021; Qiu et al., 2025a). Thus, model-utilized features tend to induce larger-magnitude representations in the forward propagation and have stronger influence on prediction, whereas irrelevant features are less likely to produce large norms. Low effective rank or spectral concentration in trained weights (Feng et al., 2022; Huh et al., 2021) further makes this amplification more pronounced, though it is not the primary source of feature importance. Therefore, subgraphs containing more model-utilized features are expected to produce representations with larger norms.

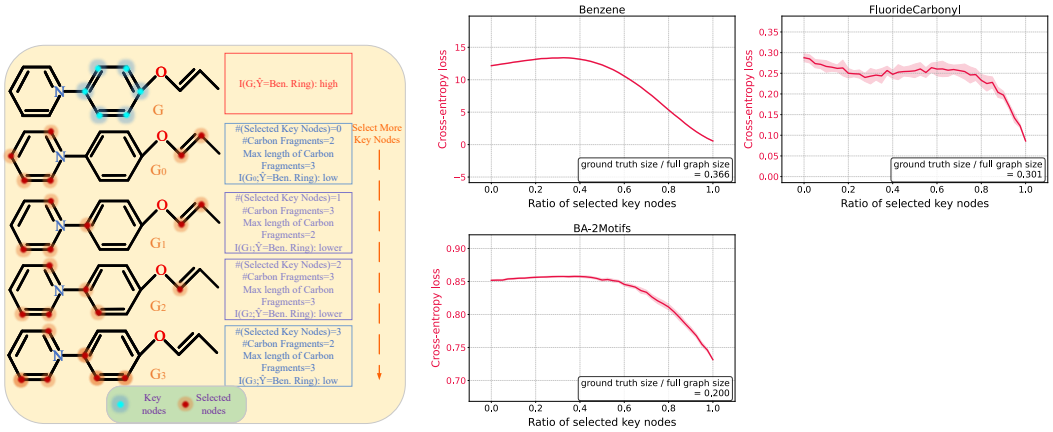
We argue that, in the context of explanation, features that can be effectively utilized by the model should be regarded as important features. Our analysis and experiments support representation norm as an informative search objective; the final subgraph is evaluated against ground-truth explanations. Building on the above analysis and observations, we introduce a new solution for GNN explanation: we identify explanatory subgraphs (and thus important features) by maximizing the norm of the subgraph's representation. Following this principle, we develop NORMA (**N**orm-**O**riented **R**epresentation **M**aximization for subgraph **A**nalysis), a parameterized explainer that uses the negative $\ell_2$ norm of the subgraph's representation as its optimization objective. Experiments across multiple datasets show that NORMA consistently outperforms MMI-based criteria.

Our main contributions are summarized as follows: 1. We empirically identify an optimization difficulty in MMI-based GNN explanation: the practical cross-entropy surrogate can remain flat or weakly informative during partial recovery of key structures. 2. By analyzing how input features interact with learned model parameters, we introduce a subgraph-exploration criterion that favors features producing larger output magnitudes, and provide theoretical and empirical support for its informative optimization signal. 3. Based on this criterion, we propose NORMA, a parameterized explainer that identifies explanatory subgraphs by maximizing the $\ell_2$ norm of the subgraphs's representation. Experiments across multiple datasets demonstrate consistent improvements over MMI-based methods.

## 2 Related Works

Many post-hoc GNN explanation methods use prediction probabilities to identify compact subgraphs that retain strong support for the original prediction, as exemplified by EiG-Search (Lu et al., 2024). A widely adopted formulation is mutual information maximization (MMI), with GNNExplainer (Ying et al., 2019) and PGExplainer (Luo et al., 2020) as representative methods that seek subgraphs preserving information about the model prediction under size constraints. These approaches rely on the model's predictive responses to evaluate and select explanatory structures. Despite the widespread use of prediction probabilities, ProxyExplainer (Chen et al., 2024b) and MixupExplainer (Zhang et al., 2023) highlight that predictions on extracted subgraphs **do not always reliably indicate whether the extracted subgraphs are truly important**. We further analyze the limitations of probability-based search signals in Section 3.

Studies of feature–weight alignment suggest that task-relevant representations can align with dominant directions of learned weight matrices (Rangamani et al., 2023; Qiu et al., 2025a). Features with stronger projections onto high-gain directions are preferentially amplified during forward propagation, motivating larger representation magnitudes for model-utilized features. This perspective provides a




(a) When the number of selected key nodes increases from 0 to 3, the selected carbon fragments remain short and insufficient to form a benzene-ring pattern. Thus, the partial subgraphs provide limited evidence for the prediction and may induce only weak model responses.

(b) The trend of the cross-entropy loss with respect to the ratio of selected key nodes (fixing the number of selected nodes) for Benzene, FluorideCarbonyl and BA-2Motifs. Clearly, the cross-entropy shows a flat, uninformative region within a certain range.

Figure 1: Benzene-motif illustration for subgraph selection (left) and cross-entropy trends versus key-node ratio on three datasets (right). Replacing non-key nodes with key nodes keeps the selected subgraph size fixed. The flat regions illustrate that recovering additional ground-truth components need not immediately change the model's prediction probability.

basis for investigating representation norm as a subgraph-selection signal. A more detailed discussion of both research threads is provided in Appendix A.

## 3 Uninformative Surrogate Landscapes in MMI-based Explainers

### 3.1 MMI Formulation and Its Cross-Entropy Surrogate

For a **well-trained** graph classification GNN $f_{\text{exp}}()$, $f_{\text{exp}}()$ takes a graph $G$ as input and predicts $G$'s label: $\hat{Y} = f_{\text{exp}}(G)$. The goal of post-hoc explanation of $f_{\text{exp}}()$ is to extract a subgraph $G^* \in G$ that maximizes the mutual information between $G^*$ and $\hat{Y}$:

$$G^* = \arg\max_{G'} I(\hat{Y}, G') = \arg\max_{G'} (H(\hat{Y}) - H(\hat{Y}|G')) = \arg\min_{G'} H(\hat{Y}|G') \tag{1}$$

As is proved in (Miao et al., 2022), the conditional entropy term $H(\hat{Y}|G')$ has an upper bound $-\mathbb{E}_{G',\hat{Y}}\left[\log \mathbb{P}(\hat{Y}|G')\right]$, where $\mathbb{P}()$ is an arbitrary distribution and often implemented via $f_{\text{exp}}(G')$. In order not to obtain a trivial solution (i.e. $G^* = G$), it is necessary to constrain the size between $G^*$ and $G$ as the regularization term $\mathcal{L}_{const}$. After adding the size constraint, the objective of explanation can be formulated as:

$$G^* = \arg\min_{G'} \mathbb{E}_{G',\hat{Y}}\left[-\log \mathbb{P}(\hat{Y}|G')\right] + \lambda \mathcal{L}_{const}(G', G) \tag{2}$$

By using masks to obtain the subgraph, the optimization problem of subgraphs is transformed into an optimization problem of masks or mask generators (Ying et al., 2019; Luo et al., 2020).

### 3.2 Flat or Weakly Informative Cross-Entropy Landscapes

However, discriminative evidence in graph data often appears as motifs, where the relevant signal is inherently non-additive. In general, the mutual information between a subgraph and the prediction, $I(G_s, \hat{Y})$, cannot be decomposed into independent per-component terms. As a result, partially recovered components may contribute little in isolation: only when key elements are jointly included

does the subgraph become informative. Since practical MMI-based explainers optimize a cross-entropy surrogate determined by the model posterior $P(\hat{Y} \mid G_s)$, this non-additivity can manifest at the optimization level as a weak response of the surrogate loss to partial motif recovery.

We illustrate this issue using the Benzene dataset, where the prediction depends on whether a molecule contains a benzene ring. As shown in Figure 1a, the six ring carbon atoms are the key components for the prediction $\hat{Y} = \mathrm{Ben.\ Ring}$, while the remaining nodes are irrelevant.

During subgraph selection, suppose the explainer always selects six nodes and gradually replaces non-ring nodes with ring nodes. As shown in Figure 1a, when only 1–3 ring atoms are selected, the resulting subgraphs $G_1$, $G_2$, and $G_3$ still resemble common short carbon fragments, such as isolated carbon atoms or short C–C paths. These fragments frequently appear in both positive and negative molecules, and therefore provide limited evidence for identifying a benzene ring. Thus, the model response may change only marginally, offering little optimization signal before a more complete ring-like structure is recovered.

Only when the selected subgraph contains longer contiguous carbon fragments, e.g., 4–5 consecutive ring atoms, does it provide stronger ring-like evidence and begin to rule out negative samples. Accordingly, the model posterior shifts more clearly toward $\hat{Y} = \mathrm{Ben.\ Ring}$, making the cross-entropy surrogate more responsive. This matches Fig. 1b, where the Benzene curve becomes informative only after sufficient key nodes are included.

The above discussion explains the source of the threshold-like behavior from the perspective of MI non-additivity. Since exact MI is intractable in practice, we next examine how this behavior appears in the cross-entropy surrogate optimized by practical MMI-based explainers.

We examine this surrogate-level effect through a controlled experiment. On Benzene, FluorideCarbonyl, and BA-2Motifs, we fix the selected subgraph size and gradually replace non-key nodes with key nodes, producing subgraphs $G'$ with increasing key-node ratios. We then evaluate each $G'$ using the cross-entropy surrogate $\mathcal{L}_{\mathrm{CE}}(f_{\mathrm{exp}}(G'), \hat{Y})$. As shown in Figure 1b, the surrogate can remain nearly flat or weakly informative when only a small fraction of key nodes is selected, despite the progressive inclusion of key components.

# 4 Growth of Graph Representation Norm with the Number of Important Nodes

In this paper, our assumptions and analysis are primarily tailored to GCNs. We briefly discuss in the Appendix M how our conclusions for GCN extend to GIN and GAT. Before presenting our analysis, we clarify the distinction from Liu et al. (2025b). While Liu et al. (2025b) adopt a norm-based objective in the self-explainable rationalization setting, our work studies post-hoc GNN explanation with a fixed target model. Their motivation does not directly yield a formal guarantee for GNN subgraph explanation, where norm behavior is shaped by message passing, nonlinear transformations, normalization, multi-layer convolution, and graph-level pooling. We therefore provide a dedicated theoretical analysis that gives a more rigorous and quantitative characterization of norm growth for post-hoc subgraph discovery.

## 4.1 Key Feature–Matrix Alignment and Magnitude Amplification

We propose a new criterion for GNN explanation: select the subgraph that **maximizes** the graph-level representation **norm**. The motivation comes from how trained networks utilize input features. Empirically, only a small subset of features is effectively exploited and impacts prediction ("important" features). Prior work suggests that utilized features tend to align with the principal directions of learned weight matrices (Qiu et al., 2025a; Papyan et al., 2020), which capture the knowledge acquired during training (Imani et al., 2021; Rangamani et al., 2023), whereas non-utilized features behave closer to random directions (Liu et al., 2025b).

Low effective rank provides an auxiliary perspective on this mechanism. When trained weight matrices have a few dominant directions (Huh et al., 2021; Feng et al., 2022; Baratin et al., 2021), alignment-induced amplification becomes more pronounced: aligned features are amplified through matrix multiplications (Kang et al., 2024), while weakly aligned features are suppressed.

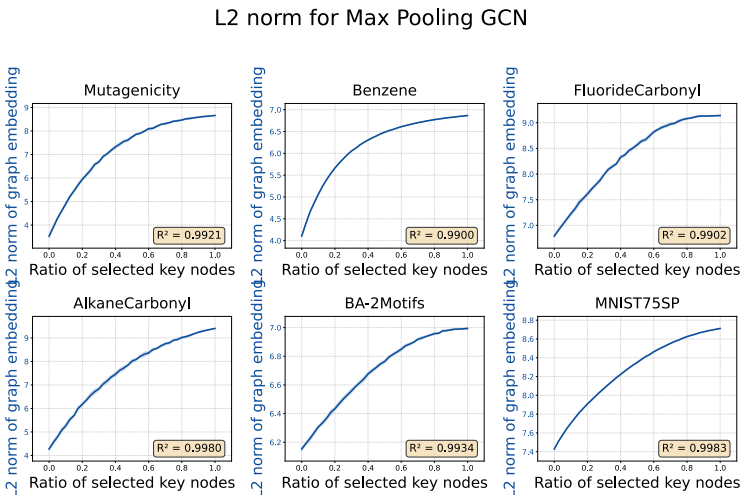



Figure 2: Growth of the graph-embedding $\ell_2$ norm for max-pooling GCNs as the key-node ratio varies under fixed subgraph size. Each plot reports the $R^2$ of a logarithmic fit; higher values indicate closer agreement.

Together, these observations suggest that model-utilized features induce larger representation magnitudes. Section 4.2 formalizes this intuition through a signal–noise separation along the directions emphasized by the trained model:

$$W \approx UV^\top,\ \mathbb{E}[U^\top x_{\text{key}}] \succ 0,\ \mathbb{E}[U^\top x_{\text{non-key}}] = 0. \tag{3}$$

Sec. 4.2 instantiates this abstraction in a one-layer message-passing model and introduces mild technical assumptions for tractable analysis.

**Preliminary experiment.** To probe this intuition, we use normalized node activation norms as node scores and multiply endpoint scores to obtain edge scores. Although this differs from optimizing the subgraph embedding norm, large node activations can contribute to large graph-level norms. Table 1 compares this training-free heuristic with the edge-mask CE method PGExplainer. Activation norm is competitive, exceeding PGExplainer's mean Edge AUROC on three of six datasets and providing preliminary support for norm-guided subgraph discovery.

Table 1: Preliminary comparison on GCN (Edge AUROC; higher is better). Values are reproduced from Table 5; PGExplainer reports mean ± standard deviation over five runs, and activation norm reports a single score.

| Method | Mutag. | Benzene | Fluoride-C | Alkane-C | BA-2M | MNIST75SP |
|---|---|---|---|---|---|---|
| PGExplainer | 0.954±0.005 | 0.735±0.059 | **0.684±0.072** | 0.792±0.033 | **0.890±0.069** | **0.683±0.016** |
| Activation norm | **0.958** | **0.920** | 0.675 | **0.947** | 0.757 | 0.608 |

### 4.2 Theoretical Analysis for the Growth of Graph Representation Norm

We adopt the feature–weight alignment abstraction introduced in Sec. 4.1. Let $\mathcal{K} \subseteq \{1, \ldots, S\}$ denote the key-node set and $K := |\mathcal{K}|$. The assumptions below are motivated by the empirical observations reviewed above: trained weights often exhibit a small number of dominant directions (Huh et al., 2021; Feng et al., 2022), task-relevant features increasingly align with those directions (Qiu et al., 2025a; Papyan et al., 2020), and non-utilized features show weaker and less stable alignment (Imani et al., 2021; Kang et al., 2024). We compactly formalize these observations as

$$\begin{aligned} &W \approx UV^\top, \qquad U, V \in \mathbb{R}^{d\times r}, \qquad V^\top V = I_r, \\ &\mu := \mathbb{E}[U^\top x_i] \succ 0, \qquad \mathbb{E}[x_i x_i^\top] = \sigma^2 I, \qquad i \in \mathcal{K}, \\ &\mathbb{E}[U^\top x_j] = 0, \qquad \mathbb{E}[x_j x_j^\top] = \sigma^2 I, \qquad j \notin \mathcal{K}. \end{aligned} \tag{4}$$

Here, $U$ spans the directions emphasized by the trained model, with scaling absorbed into $U$. The assumptions capture stable key-feature alignment and expected non-key cancellation; they are tractable abstractions rather than exact distributional descriptions (see Appendix F).

**Uniform neighbor mixing (simplified setting).** We consider a graph with $S$ nodes in which each node has $n$ neighbors sampled uniformly from all nodes. Hence,

$$\mathbb{E}_{m\in\mathcal{N}(k)}[x_m] = \frac{K}{S}\mathbb{E}[x_i] + \frac{S-K}{S}\mathbb{E}[x_j], \qquad i \in \mathcal{K},\ j \notin \mathcal{K}. \tag{5}$$

This isolates the expected contributions of key and non-key features.

**Linear aggregation (simplified message passing).** For clarity, we use

$$h_k = W^\top \left( x_k + \frac{1}{n} \sum_{m\in\mathcal{N}(k)} x_m \right). \tag{6}$$

This retains the interaction between aggregation and the learned transformation while abstracting away nonlinearities that are not needed for the scaling argument.

**Simplified covariance structure.** We further assume

$$\sum_{u,v} \mathrm{Tr}(\mathrm{Cov}(x_u, x_v)) = O(K). \tag{7}$$

This keeps second-order fluctuations lower order than the aligned mean signal.

For sum pooling, $z_{\mathrm{sum}} = \sum_{k=1}^{S} h_k$. Under the above conditions, the dominant terms satisfy

$$\begin{aligned} \mathbb{E}[z_{\mathrm{sum}}] = 2KV\mu, \quad \|\mathbb{E}[z_{\mathrm{sum}}]\|_2^2 = 4K^2\|V\mu\|_2^2 = \Theta(K^2), \\ \mathrm{Tr}(\mathrm{Cov}(z_{\mathrm{sum}})) = O(K), \qquad \mathbb{E}\|z_{\mathrm{sum}}\|_2^2 = \Theta(K^2). \end{aligned} \tag{8}$$

Finally, $\|\mathbb{E}[z_{\mathrm{sum}}]\|_2 \le \mathbb{E}\|z_{\mathrm{sum}}\|_2 \le \sqrt{\mathbb{E}\|z_{\mathrm{sum}}\|_2^2}$, so

$$\boxed{\mathbb{E}\|z_{\mathrm{sum}}\|_2 = \Theta(K)}. \tag{9}$$

The complete derivation and the extensions to normalization, activation, multi-layer propagation, and other pooling operators are provided in Appendix F. Sec. 6.2 and Appendix E.6 provide empirical evidence beyond these simplified settings.

### 4.3 Main Conclusion

We use the following proposition to formalize the norm-growth mechanism observed in our experiments under a standard GCN-style message-passing architecture.

**Proposition 1** (Norm growth under pooling)**.** *Consider a GCN with normalized adjacency $\hat{A} = D^{-1/2}(A+I)D^{-1/2}$ and layerwise update*

$$H^{(\ell+1)} = \mathrm{ReLU}\Big(\mathrm{LN}\Big(\hat{A}H^{(\ell)}W^{(\ell)}\Big)\Big). \tag{10}$$

*Let $z$ be the graph embedding obtained by sum/mean/max pooling over a node set $S$. Under the low-rank alignment and small-leakage conditions (Sec. 4.2, App. F), and the max-pooling EVT (Gnedenko, 1993) conditions in Appendix J.4.2, let $K := |\mathcal{K}|$ (denoted $|I|$ in the appendix) be the number of informative nodes. Then*

1. ***Sum/mean pooling:*** $\mathbb{E}\|z\|_2 = \Theta(K)$ *(mean differs by a constant $1/|S|$ factor).*
2. ***Max pooling:*** $\mathbb{E}\|z\|_2 = \Theta(\log K)$.

*Proof.* A proof sketch is provided below; the complete proof is in Appendix F.

Proposition 1 characterizes sufficient conditions under which the observed norm-growth mechanism holds, relying on feature–weight alignment, non-degenerate propagation, and informative pooling. Under these conditions, the representation norm serves as a monotonic proxy for the number of model-utilized nodes captured by a subgraph, whose linear or logarithmic growth provides an informative objective for subgraph discovery; this behavior is empirically validated in Sec. 6.2, with GIN/GAT evidence and operator-view discussion in Appendices E.6, E.7, and M.

## 5 Method Design

For graph classification GNN $f_{\text{exp}}$, it routinely consists of three components: node encoder $f_{\text{enc}}$, pooling layer $f_{\text{pool}}$ and classification head $f_{\text{clf}}$.

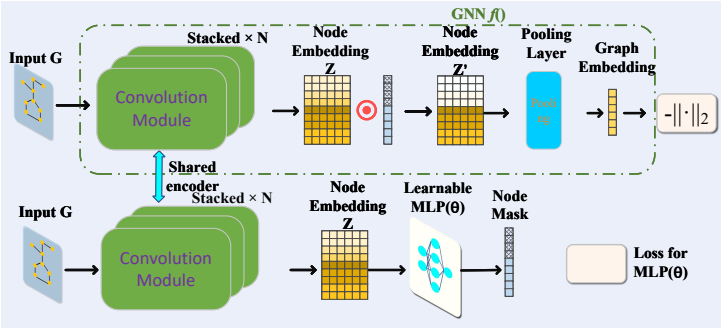



Figure 3: Overview of NORMA.

We follow the method in (Miao et al., 2022) to sample subgraph $G_s$ via the corresponding node mask $M$ generated by the explainer $\Phi_\theta$:

$$\begin{gathered} G_s = (V, E, \mathbf{X} \odot M),\ M = [m_1, \ldots, m_{|V|}] \\ m_i \sim \text{Bernoulli}(r_i),\ r_i = \phi_\theta(z_i) \in [0,1],\ Z = f_{\text{enc}}(G) = [z_1, \ldots, z_{|V|}] \end{gathered} \tag{11}$$

As is shown in Figure 3, $z'_{\text{s}}$ is defined as the graph-level representation of the sampled subgraph $G_s$, and the **norm loss** is defined as the negative $l_2$ norm of the graph embedding vector $z'_{\text{s}}$:

$$\mathcal{L}_{\text{norm}} = -\|z'_{\text{s}}\|_2,\ z'_{\text{s}} = f_{\text{pool}}(f_{\text{enc}}(G_s)) \tag{12}$$

By minimizing $\mathcal{L}_{\text{norm}}$, the explainer $\Phi_\theta$ is trained to capture the part of input graph that $f_{\text{exp}}$ utilizes.

Following the previous works (Liu et al., 2025b; Ying et al., 2019), the size constraint $\mathcal{L}_{const}$ is implementation via the size loss with pre-defined percentage $s$ and entropy loss:

$$\mathcal{L}_{\text{const}} = \lambda_1 \left| \frac{||M||_1}{|V|} - s \right| - \frac{\lambda_2}{|V|} \sum_{i=1}^{|V|} [r_i \log(r_i) + (1 - r_i) \log(1 - r_i)] \tag{13}$$

Finally, the training object for explainer $\Phi_\theta$ can be formalized as:

$$\mathcal{L}_\Phi = \mathcal{L}_{\text{norm}} + \mathcal{L}_{\text{const}}\ s.t.\ M = \Phi_\theta(\mathbf{Z}), \mathbf{Z} = f_{\text{enc}}(G = (V, E, \mathbf{X})) \tag{14}$$

## 6 Experiments

In the experiments, we focus on the following research questions (RQs): **RQ1:** Does the norm of the subgraph's representation conform to the characterization in Proposition 1? **RQ2:** Can the norm objective serve as an effective loss function for standard graph explanation tasks? **RQ3:** Does the norm objective provide a more stable optimization signal than the cross-entropy surrogate used by MMI-based explainers? Additional objective/mask comparisons and norm–output correlations are in Appendices E.1–E.2; case studies and sensitivity analyses are in Appendices E.9 and E.4.

### 6.1 Experimental Settings

**Datasets** We evaluate NORMA on six datasets with annotated ground-truth rationales: four real-world molecular datasets, Benzene, FluorideCarbonyl, AlkaneCarbonyl (Sanchez-Lengeling et al., 2020), and Mutagenicity (Kazius et al., 2005); one synthetic dataset, BA-2Motifs (Luo et al., 2020); and one superpixel dataset, MNIST75SP (Monti et al., 2017). For each dataset, the annotated nodes or edges define the explanatory subgraph and are used as the reference for evaluating explanation quality.

**GNNs** We train GCN classifiers as the explained models, using a 5-layer GCN for MNIST75SP and a 3-layer GCN for the other datasets. Each layer applies GCNConv (Kipf and Welling, 2017),

Table 2: Explanation Performance (AUROC) on edges on GCN. The SOTA of previous works is emphasised with underline.

| Method | Mutagenicity | Benzene | Fluoride-Carbonyl | Alkane-Carbonyl | BA-2Motifs | MNIST75SP |
|---|---|---|---|---|---|---|
| GradCAM | 0.716±0.000 | 0.602±0.000 | 0.699±0.000 | 0.568±0.000 | 0.560±0.000 | 0.511±0.000 |
| GNNExplainer | 0.544±0.002 | 0.523±0.001 | 0.680±0.001 | 0.827±0.017 | 0.549±0.006 | 0.498±0.001 |
| PGExplainer | 0.954±0.005 | 0.735±0.059 | 0.684±0.072 | 0.792±0.033 | 0.890±0.069 | <u>0.683±0.016</u> |
| Eig-Search | 0.873±0.005 | 0.630±0.002 | <u>0.740±0.002</u> | <u>0.898±0.003</u> | 0.901±0.001 | 0.490±0.004 |
| MatchExplainer | 0.856±0.001 | 0.810±0.002 | 0.736±0.001 | 0.841±0.007 | <u>0.953±0.003</u> | 0.466±0.009 |
| ProxyExplainer | <u>0.975±0.004</u> | <u>0.847±0.022</u> | 0.739±0.014 | 0.864±0.019 | 0.927±0.033 | 0.644±0.011 |
| NORMA (Ours) | **0.998±0.002** | **0.933±0.004** | **0.766±0.005** | **0.923±0.001** | **0.972±0.001** | **0.762±0.002** |

LayerNorm, and ReLU, followed by max-pooling readout. All models are trained on 80% of each dataset with Adam, learning rate $1e^{-3}$, and weight decay $1e^{-4}$; details are in Appendix D.

**Metrics** We compare predicted masks with annotated ground-truth rationales, treating rationale edges/nodes as positives and the remaining components as negatives. Edge AUROC is used as the primary metric, and $\mathrm{Fidelity}{-}^{\mathrm{prob}}$ (Yuan et al., 2022) under different sparsity levels is reported as a complementary faithfulness metric.

**Baselines** We compare NORMA with representative graph explainers from different categories: GradCAM (Pope et al., 2019), GNNExplainer (Ying et al., 2019), PGExplainer (Luo et al., 2020), ProxyExplainer (Chen et al., 2024b), Eig-Search (Lu et al., 2024), and MatchExplainer (Wu et al., 2023a) [2].

### 6.2 RQ1

Proposition 1 predicts logarithmic norm growth for max-pooling GNNs and linear growth for mean- and sum-pooling GNNs.

To verify this, we fix the selected subgraph size to the ground-truth rationale size and vary the proportion of selected key nodes from 0.0 to 1.0. For each ratio, we compute the $\ell_2$ norm of the resulting graph-level representation and average it over the test graphs. We then compare the empirical curve with the predicted growth form using the coefficient of determination $R^2$. Figure 2 shows that the empirical max-pooling curves are consistent with logarithmic growth over the evaluated range, with high $R^2$ values. The same protocol also confirms the predicted linear growth for mean- and sum-pooling GCNs in the appendix, as shown in Figures 5 and 6. We further observe similar log-shaped trends for max-pooling GIN and GAT in Figures 12 and 13. These results support the norm-growth mechanism without uniquely identifying its asymptotic rate.

### 6.3 RQ2

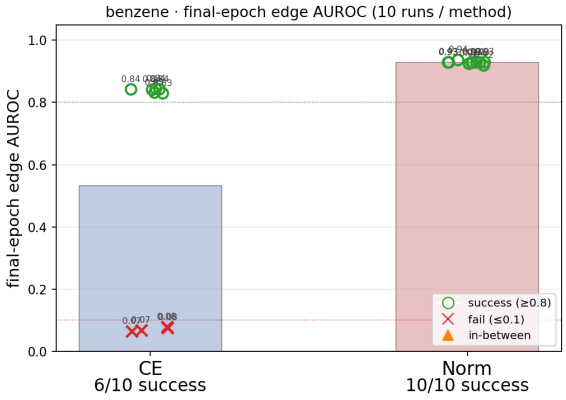
We evaluate explanation accuracy by comparing predicted edge importance scores with dataset-provided rationales using Edge AUROC.

As shown in Table 2, NORMA achieves consistently strong performance across all datasets, indicating that optimizing the representation norm effectively guides the explainer toward important graph structures. Appendix E.1 provides a more strictly controlled comparison of the norm and CE optimization objectives across node- and edge-mask parameterizations. We additionally report $\mathrm{Fidelity}{-}^{\mathrm{prob}}$ under different sparsity levels as a complementary faithfulness check in Appendix E.5.

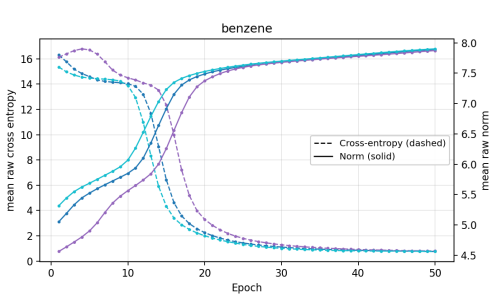
### 6.4 RQ3

We replace NORMA's norm loss with CE while retaining the node-mask parameterization and training protocol. Edge scores are derived from node scores; PGExplainer instead learns edge masks directly.

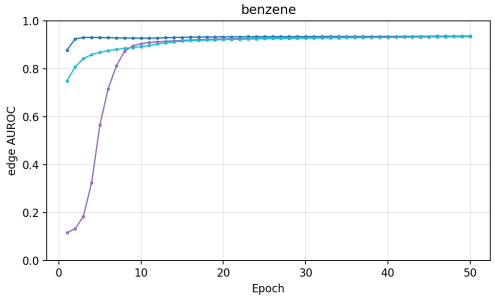
[2]The edge mask of MatchExplainer is obtained by multiplying the node masks of the two endpoints, following the same approach used in NORMA. See Appendix D for more details.




(a) Final-epoch AUROC across 10 seeds.




(b) Norm and cross-entropy trajectories.




(c) Edge-AUROC trajectories.

Figure 4: Optimization stability on Benzene: (a) final Edge AUROC across seeds; (b) norm (solid) and CE (dashed) during norm optimization; (c) corresponding Edge AUROC.

On Benzene, we run both explainers with 10 random seeds. As shown in Figure 4a, NORMA succeeds in all 10 runs, while the CE-based explainer succeeds in only 6 runs, indicating higher sensitivity to initialization. Figures 4b and 4c further show that, under norm optimization, the representation norm increases steadily, the subgraph cross-entropy decreases rapidly despite not being explicitly optimized, and Edge AUROC improves consistently. These results suggest that the norm objective provides a more reliable signal for recovering explanatory subgraphs. Additional results on BA-2Motifs are reported in Appendix E.8.

## 7 Conclusion and Limitations

We identify an optimization challenge in MMI-based post-hoc graph explanation: non-additive graph evidence can make the practical cross-entropy surrogate weakly informative under partial recovery of key structures. We propose **NORMA**, which optimizes the $\ell_2$ norm of the subgraph representation. Theory and experiments support norm-guided recovery of explanatory structures, with strong benchmark performance relative to cross-entropy-based objectives.

High norm does not imply causal correctness: NORMA may recover spurious shortcuts used by the target GNN. This challenge also affects probability-based post-hoc explainers, since shortcuts can increase predictive confidence. Distinguishing or removing such shortcuts belongs to causal learning and debiasing, beyond our scope. Strong constraints on representation magnitude, over-smoothing, or coherently activated background features may also weaken the norm signal; these settings remain to be systematically evaluated.

## Acknowledgment

This work was supported by the National Key Research and Development Program of China under Grant 2024YFC3307900; the National Natural Science Foundation of China under Grants 62376103, 62302184, 62436003, 62472332, and 62672205; the Major Science and Technology Project of Hubei Province under Grants 2025BAB011, 2024BAA008, and 2024BAB039; the Hubei Science and Technology Talent Service Project under Grant 2024DJC078; the Science and Technology Innovation Talent Program under grant 2023CB005; and Wuhan Dameng Database Co., Ltd. The authors declare no competing interests.

## References

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. *Advances in neural information processing systems*, 32, 2019.

Federico Baldassarre and Hossein Azizpour. Explainability techniques for graph convolutional networks, 2019. URL `https://arxiv.org/abs/1905.13686`.

Aristide Baratin, Thomas George, César Laurent, R Devon Hjelm, Guillaume Lajoie, Pascal Vincent, and Simon Lacoste-Julien. Implicit regularization via neural feature alignment. In *International Conference on Artificial Intelligence and Statistics*, pages 2269–2277. PMLR, 2021.

Jasmijn Bastings, Wilker Aziz, and Ivan Titov. Interpretable neural predictions with differentiable binary variables. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2963–2977, 2019.

Daniel Beaglehole, Ioannis Mitliagkas, and Atish Agarwala. Gradient descent induces alignment between weights and the empirical ntk for deep non-linear networks. *arXiv preprint arXiv:2402.05271*, 2024.

Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. A game theoretic approach to class-wise selective rationalization. *Advances in neural information processing systems*, 32, 2019.

Jialin Chen, Kenza Amara, Junchi Yu, and Rex Ying. Generative explanations for graph neural network: Methods and evaluations, 2023. URL `https://arxiv.org/abs/2311.05764`.

Yongqiang Chen, Yatao Bian, Bo Han, and James Cheng. How interpretable are interpretable graph neural networks?, 2024a. URL `https://arxiv.org/abs/2406.07955`.

Zhuomin Chen, Jiaxing Zhang, Jingchao Ni, Xiaoting Li, Yuchen Bian, Md Mezbahul Islam, Ananda Mohan Mondal, Hua Wei, and Dongsheng Luo. Generating in-distribution proxy graphs for explaining graph neural networks, 2024b. URL `https://arxiv.org/abs/2402.02036`.

Ruili Feng, Kecheng Zheng, Yukun Huang, Deli Zhao, Michael Jordan, and Zheng-Jun Zha. Rank diminishing in deep neural networks. *Advances in Neural Information Processing Systems*, 35: 33054–33065, 2022.

Tomer Galanti and Tomaso Poggio. Sgd noise and implicit low-rank bias in deep neural networks. 2022.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

BV Gnedenko. On the limiting distribution of the maximum term in a random series. *Breakthroughs in Statistics: Foundations and Basic Theory*, page 195, 1993.

Shurui Gui, Hao Yuan, Jie Wang, Qicheng Lao, Kang Li, and Shuiwang Ji. Flowx: Towards explainable graph neural networks via message flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(7):4567–4578, 2023.

Suriya Gunasekar, Blake E Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Implicit regularization in matrix factorization. *Advances in neural information processing systems*, 30, 2017.

Qiang Huang, Makoto Yamada, Yuan Tian, Dinesh Singh, and Yi Chang. Graphlime: Local interpretable model explanations for graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 35(7):6968–6972, 2022.

Minyoung Huh, Hossein Mobahi, Richard Zhang, Brian Cheung, Pulkit Agrawal, and Phillip Isola. The low-rank simplicity bias in deep networks. *arXiv preprint arXiv:2103.10427*, 2021.

Ehsan Imani, Wei Hu, and Martha White. Representation alignment in neural networks. *arXiv preprint arXiv:2112.07806*, 2021.

Neil Jethani, Mukund Sudarshan, Yindalon Aphinyanaphongs, and Rajesh Ranganath. Have we learned to explain?: How interpretability methods can learn to encode predictions in their interpretations. In *International Conference on Artificial Intelligence and Statistics*, pages 1459–1467. PMLR, 2021.

Ziwei Ji and Matus Telgarsky. Directional convergence and alignment in deep learning. *Advances in Neural Information Processing Systems*, 33:17176–17186, 2020.

Katie Kang, Amrith Setlur, Claire Tomlin, and Sergey Levine. Deep neural networks tend to extrapolate predictably, 2024. URL `https://arxiv.org/abs/2310.00873`.

Jeroen Kazius, Ross McGuire, and Roberta Bursi. Derivation and validation of toxicophores for mutagenicity prediction. *Journal of medicinal chemistry*, 48(1):312–320, 2005.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Dmitry Kopitkov and Vadim Indelman. Neural spectrum alignment: Empirical study. In *International Conference on Artificial Neural Networks*, pages 168–179. Springer, 2020.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 107–117, 2016.

Wei Liu, Haozhao Wang, Jun Wang, Ruixuan Li, Chao Yue, and YuanKai Zhang. Fr: Folded rationalization with a unified encoder. *Advances in Neural Information Processing Systems*, 35: 6954–6966, 2022.

Wei Liu, Haozhao Wang, Jun Wang, Ruixuan Li, Xinyang Li, Yuankai Zhang, and Yang Qiu. Mgr: Multi-generator based rationalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12771–12787, 2023a.

Wei Liu, Jun Wang, Haozhao Wang, Ruixuan Li, Zhiying Deng, Yuankai Zhang, and Yang Qiu. D-separation for causal self-explanation. *Advances in Neural Information Processing Systems*, 36: 43620–43633, 2023b.

Wei Liu, Jun Wang, Haozhao Wang, Ruixuan Li, Yang Qiu, Yuankai Zhang, Jie Han, and Yixiong Zou. Decoupled rationalization with asymmetric learning rates: A flexible lipschitz restraint. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1535–1547, 2023c.

Wei Liu, Zhiying Deng, Zhongyu Niu, Jun Wang, Haozhao Wang, YuanKai Zhang, and Ruixuan Li. Is the mmi criterion necessary for interpretability? degenerating non-causal features to plain noise for self-rationalization. *Advances in Neural Information Processing Systems*, 37:117636–117656, 2024a.

Wei Liu, Haozhao Wang, Jun Wang, Zhiying Deng, Yuankai Zhang, Cheng Wang, and Ruixuan Li. Enhancing the rationale-input alignment for self-explaining rationalization. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 2218–2230. IEEE, 2024b.

Wei Liu, Zhiying Deng, Zhongyu Niu, Jun Wang, Haozhao Wang, and Ruixuan Li. Exploring practical gaps in using cross entropy to implement maximum mutual information criterion for rationalization. *Transactions of the Association for Computational Linguistics*, 13:577–594, 2025a.

Wei Liu, Zhiying Deng, Zhongyu Niu, Jun Wang, Haozhao Wang, Zhigang Zeng, and Ruixuan Li. Breaking free from mmi: A new frontier in rationalization by probing input utilization. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Wei Liu, Zhongyu Niu, Lang Gao, Zhiying Deng, Jun Wang, Haozhao Wang, and Ruixuan Li. Adversarial cooperative rationalization: The risk of spurious correlations in even clean datasets. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 39126–39146. PMLR, 2025c. URL https://proceedings.mlr.press/v267/liu25av.html.

Shengyao Lu, Bang Liu, Keith G Mills, Jiao He, and Di Niu. Eig-search: Generating edge-induced subgraphs for gnn explanation in linear time. In *International Conference on Machine Learning*, pages 33069–33088. PMLR, 2024.

Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. *Advances in neural information processing systems*, 33:19620–19631, 2020.

Siqi Miao, Mia Liu, and Pan Li. Interpretable and generalizable graph learning via stochastic attention mechanism. In *International Conference on Machine Learning*, pages 15524–15543. PMLR, 2022.

Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5115–5124, 2017.

Maximilian Muschalik, Fabian Fumagalli, Paolo Frazzetto, Janine Strotherm, Luca Hermes, Alessandro Sperduti, Eyke Hüllermeier, and Barbara Hammer. Exact computation of any-order shapley interactions for graph neural networks, 2025. URL https://arxiv.org/abs/2501.16944.

Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020.

Phillip E Pope, Soheil Kolouri, Mohammad Rostami, Charles E Martin, and Heiko Hoffmann. Explainability methods for graph convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10772–10781, 2019.

Haiquan Qiu, You Wu, Yingjie Tan, Yaqing Wang, and Quanming Yao. Spectral alignment as predictor of loss explosion in neural network training. *arXiv preprint arXiv:2510.04202*, 2025a.

Yang Qiu, Yixiong Zou, Jun Wang, Wei Liu, Xiangyu Fu, and Ruixuan Li. Quantifying distributional invariance in causal subgraph for irm-free graph generalization. *Advances in Neural Information Processing Systems*, 38:73106–73136, 2025b.

Akshay Rangamani, Marius Lindegaard, Tomer Galanti, and Tomaso A Poggio. Feature learning in deep classifiers through intermediate neural collapse. In *International conference on machine learning*, pages 28729–28745. PMLR, 2023.

Benjamin Sanchez-Lengeling, Jennifer Wei, Brian Lee, Emily Reif, Peter Wang, Wesley Qian, Kevin McCloskey, Lucy Colwell, and Alexander Wiltschko. Evaluating attribution for graph neural networks. *Advances in neural information processing systems*, 33:5898–5910, 2020.

Michael Sejr Schlichtkrull, Nicola De Cao, and Ivan Titov. Interpreting graph neural networks for nlp with differentiable edge masking, 2022. URL https://arxiv.org/abs/2010.00577.

Minh Vu and My T Thai. Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. *Advances in neural information processing systems*, 33:12225–12235, 2020.

Xiang Wang, Yingxin Wu, An Zhang, Xiangnan He, and Tat-Seng Chua. Towards multi-grained explainability for graph neural networks. *Advances in neural information processing systems*, 34: 18446–18458, 2021.

Fang Wu, Siyuan Li, Xurui Jin, Yinghui Jiang, Dragomir Radev, Zhangming Niu, and Stan Z Li. Rethinking explaining graph neural networks via non-parametric subgraph matching. In *International conference on machine learning*, pages 37511–37523. PMLR, 2023a.

Lingfei Wu, Yu Chen, Kai Shen, Xiaojie Guo, Hanning Gao, Shucheng Li, Jian Pei, Bo Long, et al. Graph neural networks for natural language processing: A survey. *Foundations and Trends® in Machine Learning*, 16(2):119–328, 2023b.

Robert Wu and Vardan Papyan. Linguistic collapse: Neural collapse in (large) language models. *Advances in Neural Information Processing Systems*, 37:137432–137473, 2024.

Yingxin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. Discovering Invariant Rationales for Graph Neural Networks. In *International Conference on Learning Representations*, 2022. URL `https://mlanthology.org/iclr/2022/wu2022iclr-discovering/`.

Yaochen Xie, Zhao Xu, and Shuiwang Ji. Self-supervised representation learning via latent graph prediction. In *International conference on machine learning*, pages 24460–24477. PMLR, 2022.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks?, 2019. URL `https://arxiv.org/abs/1810.00826`.

Can Yaras, Peng Wang, Zhihui Zhu, Laura Balzano, and Qing Qu. Neural collapse with normalized features: A geometric analysis over the riemannian manifold. *Advances in neural information processing systems*, 35:11547–11560, 2022.

Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32, 2019.

Junchi Yu, Tingyang Xu, Yu Rong, Yatao Bian, Junzhou Huang, and Ran He. Graph information bottleneck for subgraph recognition, 2020. URL `https://arxiv.org/abs/2010.05563`.

Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. On explainability of graph neural networks via subgraph explorations. In *International conference on machine learning*, pages 12241–12252. PMLR, 2021.

Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. Explainability in graph neural networks: A taxonomic survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(5): 5782–5799, 2022.

Jiaxing Zhang, Dongsheng Luo, and Hua Wei. Mixupexplainer: Generalizing explanations for graph neural networks with data augmentation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3286–3296, 2023.

Liu Ziyin, Isaac Chuang, Tomer Galanti, and Tomaso Poggio. Formation of representations in neural networks. *arXiv preprint arXiv:2410.03006*, 2024.

Table 3: Notations and description

| Notation | Description |
|---|---|
| $G = (V, E, \mathbf{X})$ | A graph instance, with nodes $V$, edges $E$, and node feature matrix $\mathbf{X}$. |
| $G_s \in G$ | A subgraph of $G$. |
| $G^*$ | The optimal explanatory subgraph of $G$. |
| $f_{\text{exp}} = f_{\text{clf}}(f_{\text{pool}}(f_{\text{enc}}()))$ | The GNN to be explained. |
| $f_{\text{enc}}$ | The encoding part of GNN $f$ that embeds the nodes. |
| $f_{\text{pool}}$ | The pooling layer of GNN $f$ (only for graph-classification GNN). |
| $f_{\text{clf}}$ | The classification part of GNN $f$ that predicts the labels. |
| $\hat{Y} = f(G)$ | The label of $G$ predicted by $f$. |
| $M = [m_1, \ldots, m_{\lvert V \rvert}]$ | Masks for nodes $V$. |
| $m_u$ | Mask for node $u$. |
| $\mathbf{Z}$ | The node's embedding matrix. |
| $z_u$ | Node $u$'s embedding. |
| $z_{\text{s}}$ | Graph-level representation (used in graph-level tasks). |
| $\Phi_\theta$ | Explainer parameterized by $\theta$. |
| $\phi_\theta$ | The MLP network in explainer $\Phi_\theta$. |
| $\sigma()$ | The Sigmoid function. |
| $I(A, B)$ | Mutual information between random variable $A$ and $B$. |
| $H(A)$ | Entropy of random variable $A$. |
| $H(A\|B)$ | The conditional entropy of $A$ condition on $B$. |
| $\lVert \cdot \rVert_2$ | $l_2$ norm operator. |
| $\mathcal{L}_{\text{norm}}$ | $l_2$ norm loss. |
| $\mathcal{L}_{\text{const}}$ | Loss function for information constraint. |
| $\mathcal{L}_{\text{CE}}$ | Loss function for Cross Entropy. |
| $\lambda_1$ | The hyperparameter of size loss. |
| $\lambda_2$ | The hyperparameter of entropy loss. |
| $s$ | Pre-defined level hyperparameter for size loss function. |

# A Related Works

All the notations and their meaning can be found in Table 3.

## A.1 Explainability for GNN

There exist two main lines of research of GNN explainability: self-explainable models (Miao et al., 2022; Yu et al., 2020; Chen et al., 2024a) and post-hoc methods (Ying et al., 2019; Luo et al., 2020; Gui et al., 2023; Schlichtkrull et al., 2022; Chen et al., 2024b). Self-explainable models aim to provide predictions alongside explainable rationales. On the other hand, post-hoc methods generate explanations for well-trained GNN. In this paper, we focus on post-hoc methods.

According to the taxonomic survey (Yuan et al., 2022), instance-level post-hoc methods can be classified into four distinct categories: (1) Gradients/Features-Based Methods; (2) Perturbation-Based Methods; (3) Surrogate Methods; (4) Decomposition Methods. Gradients/Features-based methods (Baldassarre and Azizpour, 2019; Pope et al., 2019) computes the gradient of prediction probabilities with respect to input features directly, thus obtaining importance scores for the input features. Perturbation-based methods (Ying et al., 2019; Luo et al., 2020; Chen et al., 2024b) determine feature importance by systematically analyzing prediction variations induced by controlled input perturbation. Surrogate methods (Huang et al., 2022; Vu and Thai, 2020) approximate model behavior by sampling datasets from the neighborhood of the input graph and subsequently fitting a simple, explainable model to the sampled data. Decomposition-based methods (Yuan et al., 2021; Muschalik et al., 2025; Lu et al., 2024) derive feature importance scores by dissecting model predictions to quantify individual feature contributions to the final prediction. This effect is more evident when the key subgraph is relatively large. During early optimization, the explainer may gradually recover additional relevant nodes; however, the cross-entropy loss often fails to provide informative optimization signals for such incremental improvements. As a result, the loss surface becomes flat over a wide region,

gradients diminish, and optimization can stall before a meaningful explanation emerges, limiting the explainer's ability to refine explanations in a stable and progressive manner.

### A.2 Spectral Alignment, Feature Amplification, and Low-Rank Bias

Multiple lines of work suggest that trained networks organize task-relevant information along structured directions in representation and weight spaces. Representation alignment studies show that, after training, leading singular directions of learned representations tend to align with task targets, with stronger alignment often appearing in deeper layers and high-performing pretrained models (Imani et al., 2021). From a kernel perspective, neural spectrum alignment further shows that, during optimization, the top eigenfunctions of the neural tangent kernel can align toward the target function, suggesting that training progressively reshapes dominant model directions toward task-relevant signals (Kopitkov and Indelman, 2020). Related spectral-alignment analyses directly examine the relationship between layer inputs and the principal singular directions of weight matrices, providing a mechanistic view that feature effectiveness depends not only on feature magnitude, but also on whether the feature lies in directions emphasized by the learned transformation (Qiu et al., 2025a). Recent studies on representation formation similarly suggest that latent representations, weights, and gradients can become mutually aligned during training, further supporting the view that learned networks organize useful information along coherent directions (Ziyin et al., 2024; Beaglehole et al., 2024).

Neural collapse offers another concrete form of feature–weight alignment. In the terminal phase of training, last-layer features collapse to their class means, class means form a structured simplex ETF geometry, and classifier weights align with the corresponding class means up to rescaling (Papyan et al., 2020). This feature–classifier alignment has also been theoretically studied under normalized-feature settings (Yaras et al., 2022) and observed beyond standard vision classifiers, including in large language models (Wu and Papyan, 2024). Extending the phenomenon beyond the final layer, intermediate neural collapse shows that, in deeper layers, within-class covariance decreases relative to between-class covariance, while the subspace spanned by class means aligns with the dominant singular subspace of the corresponding layerwise linear map (Rangamani et al., 2023). Together, these studies support the view that model-utilized features are not arbitrary, but tend to align with directions emphasized by learned representations, kernels, and transformations.

These observations motivate a direct mechanism for feature amplification. For a linear map $z = Wx$ with SVD $W = U\Sigma V^\top$, the output norm decomposes as

$$\|z\|_2^2 = \sum_i \sigma_i^2 \|v_i^\top x\|_2^2. \tag{15}$$

Thus, features with stronger projections onto high-gain directions of $W$ are preferentially amplified during forward propagation. If prediction-relevant features align more strongly with such directions than irrelevant features, they naturally induce larger activation magnitudes and have a larger effect on the model prediction. Consistent with this view, analyses of extrapolation and out-of-distribution behavior suggest that weaker representation–weight alignment is associated with diminished effective feature magnitudes and more predictable degradation in generalization regimes (Kang et al., 2024).

Low-rank bias provides an important supporting perspective on this amplification mechanism. A growing body of work suggests that trained neural networks often exhibit spectrally concentrated, effectively low-rank operators or representations, where a relatively small number of dominant directions account for much of the transformation energy. Empirically, deep models often favor low effective-rank solutions, with embedding or representation spectra concentrating on a few leading components, a phenomenon described as the "low-rank simplicity bias" (Huh et al., 2021). Related perspectives identify depth-wise rank reduction in deep architectures, including rank-diminishing trends across layers under common compositions (Feng et al., 2022). Complementary analyses attribute such structure to optimization-induced implicit regularization: in matrix factorization and deep matrix factorization, gradient-based training is biased toward low-complexity solutions and exhibits depth-dependent spectral dynamics (Gunasekar et al., 2017; Arora et al., 2019). More directly in deep networks, SGD-induced stochasticity has been argued to yield an implicit low-rank bias, with theoretical characterizations linking mini-batch SGD, often together with weight decay, to proximity to low-rank matrices (Galanti and Poggio, 2022).

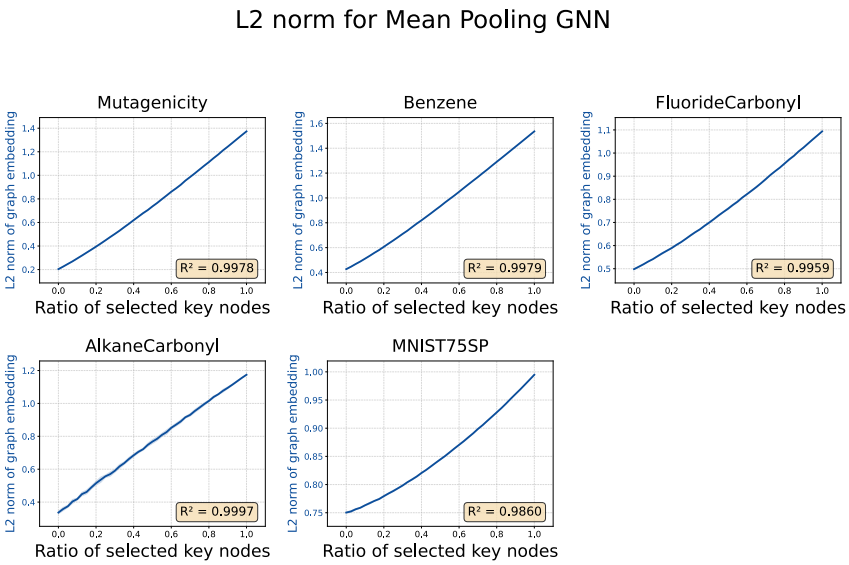



Figure 5: Growth rate of the $\ell_2$ norm of graph embedding for mean pooling GCN. The $R^2$ between the actual and ideal linear curves is shown at the bottom right of each plot, values closer to 1.0 indicate the actual curve is closer to the ideal linear curve.

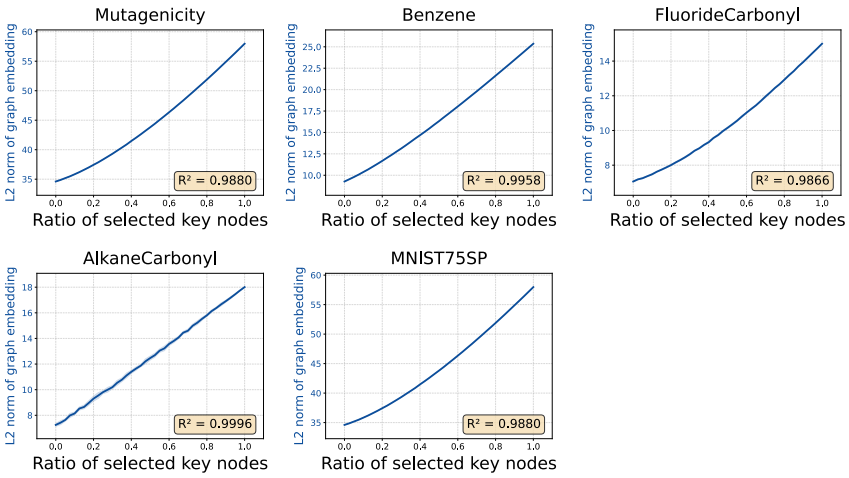



Figure 6: Growth rate of the $\ell_2$ norm of graph embedding for sum pooling GCN.

Therefore, low effective rank is not the source of feature importance by itself. Rather, it makes alignment-induced amplification more concentrated and analytically transparent: when only a limited number of directions dominate the transformation, alignment with these directions becomes especially consequential. This perspective supports our use of representation norm as a proxy for how strongly a selected subgraph aligns with and is utilized by the trained model.

### A.3 Connections to Self-Rationalization

Extractive self-rationalization learns to select a compact subset of the input that supports prediction. Lei et al. (2016) jointly train a generator and a predictive encoder to extract short, coherent text rationales, while Bastings et al. (2019) introduce differentiable binary variables for gradient-based selection with explicit control over the selection rate. Beyond cooperative training, Chang et al.

(2019) formulate class-wise rationalization as a competitive game to identify evidence supporting factual and counterfactual outcomes.

Learning meaningful selections nevertheless presents challenges that extend beyond the choice of selector. In the broader setting of amortized feature-selection explanations, Jethani et al. (2021) show that jointly trained selectors and predictors can encode predictions in their selections, rather than provide reliable explanations. Folded Rationalization shares an encoder to mitigate degeneration (Liu et al., 2022), while MGR uses multiple generators to address degeneration and spurious correlations (Liu et al., 2023a). Decoupled Rationalization connects degeneration to the predictor's Lipschitz continuity and employs asymmetric learning rates (Liu et al., 2023c). Rationale-input alignment further addresses cases where selected rationales deviate semantically from the full input despite supporting accurate predictions (Liu et al., 2024b). Together, these studies illustrate why prediction accuracy alone does not ensure that a learned selection provides a meaningful explanation.

A complementary line of work examines the criterion used to select rationales. Alternatives to MMI have been developed to reduce the influence of non-causal features (Liu et al., 2024a) and to probe input utilization through representation norms (Liu et al., 2025b). Further analysis distinguishes rationale informativeness from predictor fit, identifying practical gaps between cross-entropy optimization and the intended MMI objective (Liu et al., 2025a). These observations are related to NORMA's motivation to seek an informative selection signal beyond prediction probabilities. However, NORMA studies post-hoc subgraph selection for a fixed, trained GNN, rather than jointly learning the predictor and its rationales. Its analysis concerns the response of that fixed model as informative graph components are recovered; degeneration mechanisms arising from joint training do not directly establish this behavior.

Causal self-explanation introduces a conditional-dependence criterion grounded in d-separation (Liu et al., 2023b), while adversarial cooperative rationalization examines spurious correlations induced by the selection process itself (Liu et al., 2025c). These studies also clarify the distinction between recovering evidence used by a predictor and identifying causal evidence for the task. NORMA targets the former: a model-utilized feature can receive a large representation norm even if it reflects a spurious shortcut. Consequently, norm-guided post-hoc explanation alone does not establish causal correctness.

### A.4 Connections to Graph Generalization

Graph self-explanation also connects rationale selection with generalization under distribution shifts. GSAT derives stochastic attention from the information bottleneck principle to select task-relevant subgraphs while suppressing irrelevant information (Miao et al., 2022). DIR constructs interventional distributions and learns rationales that remain invariant across them, aiming to separate causal patterns from unstable shortcuts (Wu et al., 2022). These approaches illustrate how explanation objectives can be incorporated into predictive model training to improve both interpretability and generalization. Qiu et al. (2025b) study the relationship between distributional variation and representation norm, developing a norm-guided invariant distribution objective for causal subgraph discovery and out-of-distribution graph generalization without IRM. This work is closely related to NORMA in using representation norm to guide subgraph selection. The objectives and interpretations nevertheless differ: their criterion is motivated by distributional invariance and graph generalization, whereas NORMA uses feature–weight alignment and norm growth to guide post-hoc explanation of a fixed GNN. The shared use of representation norm therefore provides a connection between the two settings, but does not by itself transfer causal or out-of-distribution generalization guarantees to NORMA.

## B Why Mutual Information is Non-Additive on Graph Explanations

This section provides background on why mutual-information-based objectives used in graph explanation are inherently non-additive, which can induce threshold-like behavior when the evidence is distributed over multiple graph components.

### B.1 Setup

Let $G$ be an input graph, $\hat{Y}$ be the prediction of a pretrained GNN $f$ on $G$, and $G_s$ be an explanatory subgraph sampled/selected from $G$. Many explanation methods can be interpreted as maximizing the

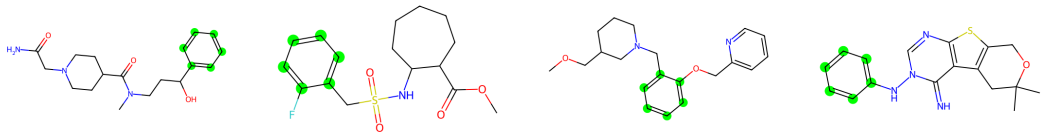


Figure 7: Positive samples (containing a benzene ring) of Benzene dataset. The benzene ring is highlighted in green.

mutual information between the explanation and the prediction:

$$\max_{G_s \subseteq G} I(\hat{Y}; G_s) \quad \text{or} \quad \max_{M} I(\hat{Y}; G_s(M)), \tag{16}$$

where $M$ denotes a mask (node/edge) parameterizing the subgraph construction.

### B.2 Mutual information is not additive across subgraph parts

Consider decomposing the explanatory subgraph into two parts, e.g., $G_s = A \cup B$ (two edge sets, or two local regions/motifs). If information were additive, one might expect

$$I(\hat{Y}; A \cup B) \approx I(\hat{Y}; A) + I(\hat{Y}; B). \tag{17}$$

However, mutual information generally does not satisfy such additivity. In fact, by the chain rule of mutual information,

$$I(\hat{Y}; A, B) = I(\hat{Y}; A) + I(\hat{Y}; B \mid A). \tag{18}$$

The second term is *conditional* and depends on $A$. Unless $B$ is conditionally independent of $A$ given $\hat{Y}$ (which rarely holds on graphs), $I(\hat{Y}; B \mid A)$ is not equal to $I(\hat{Y}; B)$. Equivalently, one can write

$$I(\hat{Y}; A \cup B) = I(\hat{Y}; A) + I(\hat{Y}; B) - I(A; B) + I(A; B \mid \hat{Y}), \tag{19}$$

showing explicit interaction terms between $A$ and $B$. Therefore, the information contributed by one part of a subgraph typically depends on whether the other part is already present.

In summary, mutual information over graph substructures is inherently non-additive due to interaction terms between subgraph parts (Eq. 18). When predictive evidence is distributed across multiple edges/nodes (e.g., motifs), MI-based objectives can exhibit synergy and threshold effects, which motivates using more informative proxy objectives for searching important structures.

### B.3 Analysis for Examples Based on Benzene

In Figure 7 and 8, we present several examples of positive (containing a benzene ring) and negative (not containing a benzene ring) samples from the Benzene dataset. We observe that positive samples often include short carbon fragments outside the benzene ring, and similar fragments also appear frequently in negative samples. If an explainer captures only a subset of the ring carbons, the selected subgraph may still resemble these common short fragments. During subgraph search, such partial ring fragments provide little discriminative evidence for separating positives from negatives. Therefore, even as more ring carbons are included, the mutual information between the explanatory subgraph $G_s$ and the label $\hat{Y} = \{\text{Ben.Ring}\}$ may fail to increase substantially. It may even decrease when the selected carbon-containing structure becomes more dispersed and deviates further from a benzene ring.

## C Growth Rate of the $\ell_2$ Norm of Graph Embedding for Mean or Sum Pooling

We follow the setup in RQ1 and plot the growth curves of GCNs using Mean-Pooling and Sum-Pooling across different datasets to verify whether the actual results align with theoretical expectations

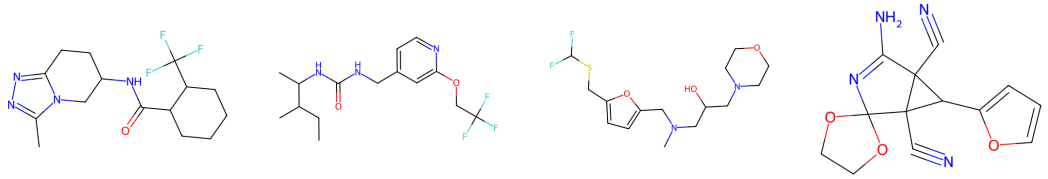


Figure 8: Negative samples (not containing a benzene ring) of Benzene dataset.

Table 4: Statistics of the datasets and GCN, GIN and GAT models

| Datasets | Mutagenicity | Benzene | Fluoride-Carbonyl | Alkane-Carbonyl | BA-2Motifs | MNIST75SP |
|---|---|---|---|---|---|---|
| Graphs# | 4337 | 12000 | 8671 | 4326 | 1000 | 70000 |
| Classes# | 2 | 2 | 2 | 2 | 2 | 10 |
| Positive Instances# | 2401 | 6001 | 1527 | 375 | 500 | - |
| Negative Instances# | 1936 | 5999 | 7144 | 3951 | 500 | - |
| Avg. Nodes # | 27.55 | 21.92 | 21.89 | 22.19 | 25.00 | 70.67 |
| Avg. rationale nodes | 4.21 | 8.03 | 6.61 | 4.28 | 5.00 | 25.25 |
| GCN/GIN layers# | 3 | 3 | 3 | 3 | 3 | 5 |
| GCN Testing Acc/Recall | 0.9255 | 0.9066 | 0.8630 | 0.9625 | 1.0 | 0.7829 |
| GIN Testing Acc/Recall | 0.9509 | 0.9217 | 0.9017 | 0.9254 | 1.0 | 0.8921 |
| GAT Testing Acc/Recall | 0.9316 | 0.8859 | 0.8435 | 0.8906 | 1.0 | 0.7921 |

in 1. Since GCNs with these two pooling strategies do not work on the BA-2Motifs dataset, we use only four datasets. Figures 5 and 6 show the growth curves of GCNs with Mean-Pooling and Sum-Pooling across the four datasets, respectively. In the bottom-right corner of each plot, we report the $R^2$ values between the actual curves and the ideal linear fit. All $R^2$ values exceed 0.98, indicating that the observed curves closely approximate linear growth, thus supporting our theoretical findings. It is important to note that our conclusion is

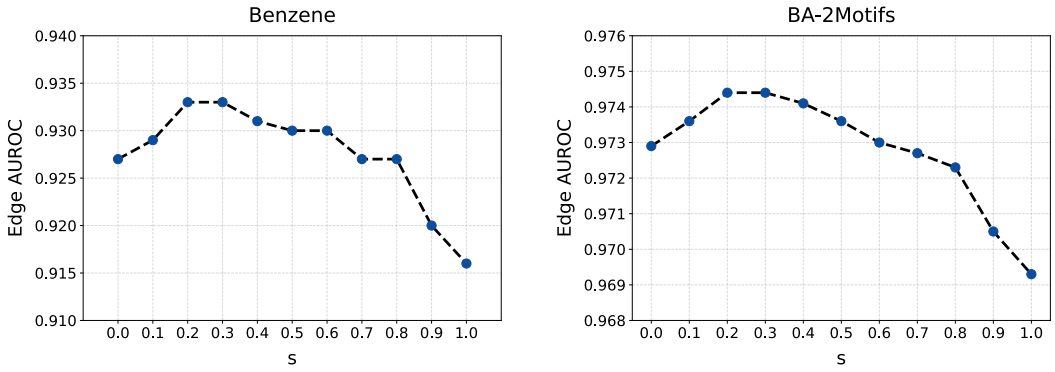
$$\mathbb{E}[\|z\|_2] = \Theta(K) \tag{20}$$

This indicates that the growth of the $\ell_2$ norm of graph embeddings is primarily driven by linear growth, but there may also be lower-order terms, which can explain why some curves are not perfectly linear.

# D Experiment Details

## D.1 Statistics of Datasets and GNNs

The GCN sequentially applies GCNConv (Kipf and Welling, 2017), LayerNorm, ReLU in each convolutional layer, followed by max pooling layer. For MNIST75SP, we use a 5-layer GCN. For other datasets, we use a 3-layer GCN. All datasets are partitioned into 80% training and 20% test sets without explicit validation set separation. All GCN models used a consistent *hidden size* of 20 across the six datasets. We use the Adam optimizer with learning rate fixed as 1e-3. The training *batch size* is set to 128 for all datasets except for the Benzene and MNIST75SP dataset, where it is configured as 512. For Benzene (Sanchez-Lengeling et al., 2020), Mutagenicity (Kazius et al., 2005) MNIST75SP (Monti et al., 2017) and BA-2Motifs (Luo et al., 2020), the training metrics is accuracy, and for FluorideCarbonyl (Sanchez-Lengeling et al., 2020), AlkaneCarbonyl (Sanchez-Lengeling et al., 2020) the training metrics is recall since these two datasets are extremely imbalanced. The dataset and training details can be found in Table 4.




(a) Hyperparameter analysis for $s$ on Benzene

(b) Hyperparameter analysis for $s$ on BA-2Motifs

Figure 9: Hyperparameter analysis for $s$

### D.2 Hyperparameter Setting

In our method the main hyperparameters are size loss coefficient $\lambda_1$, entropy loss $\lambda_2$, level $s$, learning rate, weights decay and batch size for the explainer. $\lambda_1$ is set to $1e-1$ for all datasets. All datasets use $\lambda_2 = 3e-2$. $s$ is set from 0.2 to 0.4. The learning rate is uniformly set to its default value of $3e-4$ across all datasets. The weights decay is set to $1e-4$ across all datasets. A batch size of 128 is adopted for Mutagenicity, FluorideCarbonyl and BA-2Motifs, for Benzene a larger batch size of 256 is employed, and for AlkaneCarbonyl a smaller batch size of 64 is used.

### D.3 Mask Generation and Inverse-Fidelity Protocol

**Mask generation** Most baseline explainers directly output *continuous edge masks* $m_{ij} \in [0,1]$ for each edge $(i,j) \in E$, which are used as edge-importance scores.

**NORMA:** NORMA outputs *continuous node masks* $m_i \in [0,1]$ for each node $i \in V$. To obtain a continuous edge mask for edge-level evaluation, we derive edge scores by multiplying the endpoint node scores:

$$m_{ij} = m_i \cdot m_j, \quad \forall (i,j) \in E. \tag{21}$$

**MatchExplainer:** MatchExplainer only generates *binary node masks*. To obtain continuous node importance scores, we compute each node's importance as the minimum distance to the reference nodes selected by MatchExplainer, and normalize the scores to $[0,1]$ to form continuous node masks. We then convert them to continuous edge masks via the same endpoint-product rule $m_{ij} = m_i m_j$.

**Inverse-Fidelity protocol** We adopt a unified *edge-based* subgraph construction protocol for fair comparison across methods. Given edge scores $\{m_{ij}\}$ and a sparsity level $s$ (discard ratio), we retain the top-$(1-s)$ fraction of edges and discard the remaining edges to form the explanatory subgraph $G_s$.

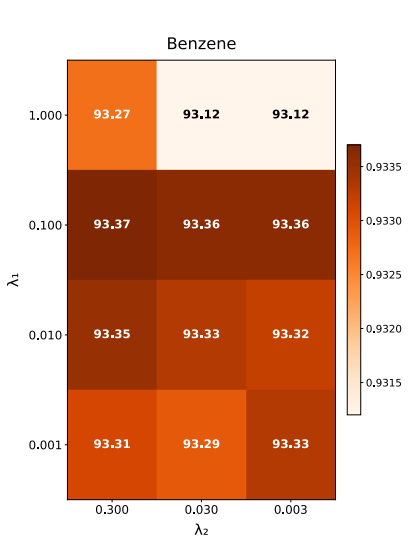
Let $\hat{Y}$ denote the predicted class on the original graph $G$ by the target GNN, and let $p(\hat{Y} \mid G)$ and $p(\hat{Y} \mid G_s)$ be the corresponding prediction probabilities. Inverse-Fidelity is computed as

$$\text{InvFid} = \mathbb{E}\left[p(\hat{Y} \mid G) - p(\hat{Y} \mid G_s)\right], \tag{22}$$

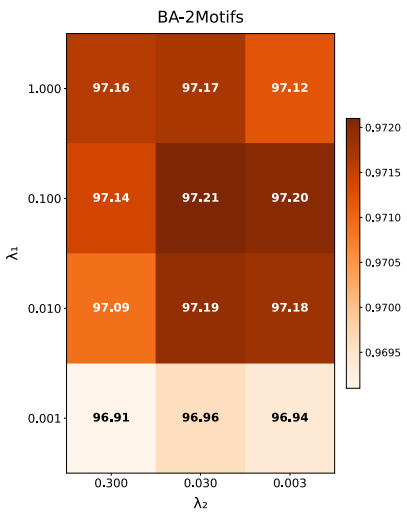
where the expectation is taken over graphs in the evaluation set (and over sampling if applicable). Lower InvFid indicates that the extracted subgraph better preserves the original model behavior.

### D.4 Experiment Platform

We conduct all experiments on a Linux machine equipped with 8 NVIDIA RTX 4090 GPUs and an Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz processor. The experimental setup uses Python 3.11.5, PyTorch 2.1.2, PyTorch Geometric 2.5.2, and CUDA 12.1.




(a) Hyperparameter analysis for $\lambda_1$ and $\lambda_2$ on Benzene




(b) Hyperparameter analysis for $\lambda_1$ and $\lambda_2$ on BA-2Motifs

Figure 10: Hyperparameter analysis for $\lambda_1$ and $\lambda_2$

# E More Experiment Results

## E.1 Objective, Mask Parameterization, and Simple Baseline Comparisons

We extend the explanation-accuracy comparison to separate the choice of objective from the mask parameterization and to assess whether direct node scoring is sufficient. All results in Table 5 use the GCN target models and Edge AUROC against the annotated ground-truth explanation edges.

**Objectives and masks.** For the node-mask variants, we retain NORMA's mask generator and replace only the objective with cross-entropy or the negative target-class logit. The target class $\hat{Y}$ is fixed to the prediction on the full graph, giving losses $-\log p(\hat{Y} \mid G_s)$ and $-a_{\hat{Y}}(G_s)$, respectively, where $a_{\hat{Y}}$ denotes the target-class logit. The node-to-edge mapping $m_{ij} = m_i m_j$, size and entropy regularization, sparsity setting, training budget, and seeds are shared across these objective variants; we use $\lambda_1 = 0.1$, $\lambda_2 = 0.03$, and $s = 0.3$. We additionally evaluate a directly parameterized edge mask with the norm and logit objective. PGExplainer supplies the direct-edge-mask CE comparison; its results are repeated from Table 2, rather than duplicated as a separate CE method. The learned variants report mean and standard deviation over five runs. The node-norm and PGExplainer entries retain the summaries from Table 2. The ten-run stability analysis in Sec. 6.4 is a separate experiment.

**Direct node scores.** The training-free alternatives score nodes by degree, hidden representation norm (activation norm), or input-feature norm, and use the same endpoint-product mapping to obtain edge scores. Activation norm uses the trained target model's node representations without fitting an explainer; NORMA instead optimizes the graph-level representation norm of the masked subgraph. The molecular inputs are one-hot atom encodings and BA-2Motifs uses constant all-ones features, so input-feature norms cannot rank nodes on these five datasets. We mark these constant-score cases with a dash; the MNIST75SP inputs provide nonconstant scores. These heuristics are deterministic for a fixed target model and input graph, and are reported as single scores.

**Effect of the objective and mask parameterization.** With the node-mask parameterization fixed, norm optimization has the highest mean AUROC on all six datasets among norm, CE, and logit objectives. CE is particularly variable on Benzene ($0.561 \pm 0.344$) and BA-2Motifs ($0.695 \pm 0.307$), consistent with the initialization sensitivity in the stability experiments. Logit optimization is competitive on several datasets, but is weaker on FluorideCarbonyl ($0.717$ versus $0.766$) and MNIST75SP ($0.615$ versus $0.762$). These are comparisons under the shared settings, rather than claims about all possible objective-specific tuning. The edge-mask norm variant exceeds the direct-

Table 5: Additional explanation comparisons (Edge AUROC, higher is better). Node-mask scores are mapped to edges by endpoint multiplication. PGExplainer is the direct-edge-mask CE variant. Learned methods report mean ± standard deviation over five runs; simple scoring rules report single values. A dash denotes constant input-feature norms with no ranking information. Bold marks the largest mean in each column.

| Method | Mutag. | Benzene | Fluoride-C | Alkane-C | BA-2M | MNIST75SP |
|---|---|---|---|---|---|---|
| *Learned objectives and mask parameterizations* | | | | | | |
| Norm (node) | 0.998±0.002 | **0.933±0.004** | **0.766±0.005** | 0.923±0.001 | **0.972±0.001** | **0.762±0.002** |
| CE (node) | 0.962±0.008 | 0.561±0.344 | 0.687±0.038 | 0.828±0.039 | 0.695±0.307 | 0.602±0.032 |
| Logit (node) | 0.991±0.005 | 0.921±0.006 | 0.717±0.029 | 0.921±0.036 | 0.966±0.002 | 0.615±0.009 |
| Norm (edge) | **0.999±0.000** | 0.888±0.038 | 0.737±0.013 | **0.974±0.009** | 0.936±0.005 | 0.682±0.024 |
| PGExplainer (CE, edge) | 0.954±0.005 | 0.735±0.059 | 0.684±0.072 | 0.792±0.033 | 0.890±0.069 | 0.683±0.016 |
| Logit (edge) | 0.993±0.002 | 0.911±0.013 | 0.729±0.042 | 0.903±0.014 | 0.943±0.001 | 0.706±0.003 |
| *Direct node scoring (no explainer training)* | | | | | | |
| Degree | 0.179 | 0.561 | 0.239 | 0.099 | 0.694 | 0.731 |
| Activation norm | 0.958 | 0.920 | 0.675 | 0.947 | 0.757 | 0.608 |
| Input-feature norm | — | — | — | — | — | 0.740 |

edge CE baseline on five datasets; MNIST75SP is nearly tied ($0.682$ versus $0.683$). Neither mask parameterization is uniformly best: the node-mask norm variant has higher mean AUROC on four datasets, while the edge-mask version is higher on Mutagenicity and AlkaneCarbonyl. The edge-mask results extend the empirical comparison; the theoretical analysis is formulated in terms of informative nodes.

**What the activation-norm baseline establishes.** The strong activation-norm result on AlkaneCarbonyl ($0.947$, compared with $0.923$ for node-mask NORMA) is consistent with the norm-maximization idea: high-norm node representations can contribute to a high-norm subgraph representation. It shows that direct node ranking can work well on this dataset, but does not establish that node norms alone suffice across datasets. Activation norm is lower than node-mask NORMA on the other five datasets, including BA-2Motifs ($0.757$ versus $0.972$), FluorideCarbonyl ($0.675$ versus $0.766$), and MNIST75SP ($0.608$ versus $0.762$). Thus, the results support both the usefulness of norm information and the additional value of learning to select a subgraph instead of relying only on individual node scores. Degree and input-feature norm are also competitive on MNIST75SP ($0.731$ and $0.740$), but neither exceeds NORMA ($0.762$).

### E.2 Correlation between Subgraph Norm and Model Outputs

This analysis examines how subgraph representation norms relate to the target model's outputs. It complements the ground-truth recovery experiments, which directly assess whether norm provides an informative signal for recovering explanatory structures. For each graph $G$, we fix $\hat{Y} = f_{\text{exp}}(G)$ and evaluate a collection of subgraphs $G_s$. For each subgraph, we record $\|z_s\|_2$, the target-class probability $p(\hat{Y} \mid G_s)$, the target-class logit $a_{\hat{Y}}(G_s)$, and cross-entropy $-\log p(\hat{Y} \mid G_s)$. We compute each correlation within an individual graph, then average the coefficients across graphs. This avoids conflating within-graph association with differences in scale between graphs. Table 6 reports both Spearman and Pearson correlations, their standard deviations across graphs, and the number of valid graphs for each dataset.

**Interpretation.** At fixed subgraph size, Figure 1b shows that cross-entropy can remain flat or even increase as more ground-truth key nodes are recovered. Figure 2, in contrast, shows smooth, monotonic growth of the average representation norm with increasing key-node coverage, with further evidence across pooling operators and GNN backbones in Figures 5, 6, 12, and 13. These experiments directly examine how the respective signals reflect progress toward recovering ground-truth structures. Moreover, the controlled node-mask comparisons in Table 5 show that replacing CE with the norm objective improves mean Edge AUROC across all six datasets. Together, these results provide the primary empirical support for norm as a proxy for recovering ground-truth explanatory structures.

Table 6: Supplementary analysis of within-graph correlations between subgraph representation norm and target-class outputs. Values report the mean and standard deviation across graphs; the standard deviation describes graph-to-graph variability, not variability across training runs. $N$ denotes the number of valid graphs for each pair. CE is computed for the same fixed target class.

| Dataset | $N$ | Norm–probability | Norm–logit | Norm–CE |
|---|---|---|---|---|
| *Spearman correlation* | | | | |
| Mutagenicity | 1015 | $0.5635 \pm 0.2409$ | $0.5744 \pm 0.2385$ | $-0.5635 \pm 0.2409$ |
| Benzene | 6001 | $0.5375 \pm 0.3200$ | $0.5416 \pm 0.3200$ | $-0.5375 \pm 0.3200$ |
| FluorideCarbonyl | 1527 | $0.5275 \pm 0.4891$ | $0.5471 \pm 0.4824$ | $-0.5275 \pm 0.4891$ |
| AlkaneCarbonyl | 375 | $0.6571 \pm 0.2206$ | $0.6769 \pm 0.2284$ | $-0.6571 \pm 0.2206$ |
| BA-2Motifs | 1000 | $0.2036 \pm 0.3275$ | $0.2879 \pm 0.3236$ | $-0.2036 \pm 0.3275$ |
| MNIST75SP | 7000 | $0.2752 \pm 0.3403$ | $0.2243 \pm 0.3746$ | $-0.2752 \pm 0.3403$ |
| *Pearson correlation* | | | | |
| Mutagenicity | 1015 | $0.7365 \pm 0.3147$ | $0.7401 \pm 0.2997$ | $-0.7011 \pm 0.2818$ |
| Benzene | 6001 | $0.7173 \pm 0.3161$ | $0.7553 \pm 0.3684$ | $-0.7245 \pm 0.2931$ |
| FluorideCarbonyl | 1527 | $0.5265 \pm 0.4767$ | $0.5695 \pm 0.5067$ | $-0.5481 \pm 0.4582$ |
| AlkaneCarbonyl | 375 | $0.6923 \pm 0.2134$ | $0.6738 \pm 0.2231$ | $-0.5931 \pm 0.2054$ |
| BA-2Motifs | 1000 | $0.1056 \pm 0.3003$ | $0.1905 \pm 0.3015$ | $-0.1094 \pm 0.2995$ |
| MNIST75SP | 7000 | $0.2068 \pm 0.2171$ | $0.2295 \pm 0.3524$ | $-0.2515 \pm 0.3187$ |

Table 6 addresses the complementary question of how norm relates to model outputs. The mean correlations indicate associations with predictive responses, while their magnitudes and across-graph variability show that these relationships differ across graphs and datasets. NORMA does not require norm to reproduce prediction probabilities: such probabilities may respond weakly or non-monotonically during partial recovery of ground-truth structures. Its effectiveness is therefore assessed through the recovery-progress experiments and controlled explanation comparisons above, rather than through a requirement of uniformly high norm–probability correlation.

### E.3 Node Classification Experiments

Although our main experiments evaluate NORMA on graph classification tasks, we further conduct experiments on node classification tasks for completeness. Since such evaluation requires node classification datasets with ground-truth explanations, and currently only a few synthetic node-level datasets provide such annotations, we evaluate NORMA on BA-Shapes, Tree-Cycle, and Tree-Grid (Luo et al., 2020).

We compare NORMA only with explainers that support node classification, since MatchExplainer, Eig-Search, and ProxyExplainer are not applicable to node classification tasks. In addition, we include MixupExplainer (Zhang et al., 2023), which supports node-level explanations, as an additional baseline.

We first construct the GCN backbone using the same GConv-LayerNorm-ReLU architecture as in Sec. 6.1. Since the task is node classification, the pooling layer is removed, while the remaining settings are kept unchanged. Specifically, we use a 3-layer GCN for BA-Shapes and Tree-Cycle, and a 4-layer GCN for Tree-Grid. The details of the GNN backbones are reported in Table 7.

For NORMA, node classification requires explaining the prediction of a target node $z_{\text{target}}$. Therefore, we replace the norm loss defined on the graph-level representation $z_{\text{s}}$ with the $\ell_2$ norm of the target-node representation:

$$\mathcal{L}_{\text{norm}} = -\|z'_{\text{target}}\|_2, \quad z'_{\text{target}} = f_{\text{enc}}(G_{\text{s}})\big|_{i=i_{\text{target}}}, \tag{23}$$

where $i_{\text{target}}$ denotes the target node. The regularization term $\mathcal{L}_{\text{const}}$ remains unchanged.

The results are shown in Table 8. As can be seen, NORMA continues to achieve strong performance on node classification tasks, demonstrating that the proposed norm-based objective is also effective beyond graph-level explanation.

Table 7: Details of GNN for node classification experiments

| Datasets | BA-Shapes | Tree-Cycle | Tree-Grid |
|---|---|---|---|
| GCN layers# | 3 | 3 | 4 |
| GCN Testing Acc | 0.8857 | 0.9029 | 0.9433 |

Table 8: Explanation Performance (AUROC) on edges for node classification tasks. The SOTA of previous works is emphasised with underline.

| Method | BA-Shapes | Tree-Cycle | Tree-Grid |
|---|---|---|---|
| GradCAM | 0.312±0.000 | 0.273±0.000 | 0.297±0.000 |
| GNNExplainer | 0.629±0.001 | 0.443±0.003 | 0.558±0.003 |
| PGExplainer | **0.999±0.000** | 0.886±0.066 | 0.665±0.031 |
| MixupExplainer | 0.811±0.102 | 0.374±0.078 | 0.714±0.137 |
| NORMA (Ours) | **0.999±0.000** | **0.972±0.003** | **0.792±0.015** |

### E.4 Hyperparameter Sensitivity

We conduct hyperparameter sensitivity studies on the real-world Benzene dataset and the synthetic BA-2Motifs dataset. NORMA involves three main hyperparameters: the size loss coefficient $\lambda_1$, the entropy loss coefficient $\lambda_2$, and the sparsity level $s$. We first evaluate the sensitivity to the sparsity level $s$, and then perform a joint sensitivity analysis of $\lambda_1$ and $\lambda_2$.

On Benzene, we fix $\lambda_1 = 1e-1$ and $\lambda_2 = 3e-2$, and vary $s \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ to examine its effect on Edge AUROC. On BA-2Motifs, we use the same settings to evaluate its impact on Edge AUROC. The corresponding results are reported in Figure 9.

We then conduct a joint sensitivity analysis for the hyperparameters $\lambda_1$ and $\lambda_2$. Specifically, we evaluate $\lambda_1 \in \{1.0, 0.1, 0.01, 0.001\}$ and $\lambda_2 \in \{0.3, 0.03, 0.003\}$, while fixing $s = 0.3$. The corresponding results are reported in Figure 10.

### E.5 Fidelity Experiment

**Behavior preservation:** We further report Inverse-Fidelity under different sparsity levels as a complementary sanity check.

$$\mathrm{InvFid} = \mathbb{E}\Big[p(f_{\exp}(G) = \hat{Y}) - p(f_{\exp}(G_s) = \hat{Y})\Big] \tag{24}$$

Here, sparsity $s$ denotes the discard ratio, i.e., we retain the top $(1-s)$ fraction of mask scores to form the explanatory subgraph (e.g., $s = 0.7$ keeps $30\%$). For fair cross-method comparison, we use a unified edge-based subgraph construction protocol for all explainers (see Appendix D.3 for details). As shown in Figure 11, NORMA consistently achieves strong results across sparsity levels, suggesting that the extracted subgraphs largely preserve the original model behavior. Taken together, the Edge AUROC results provide the primary evidence that the embedding norm is an effective objective for identifying important structures, while the Inverse-Fidelity results offer additional support that the extracted subgraphs do not substantially distort the model behavior under a fair, unified evaluation protocol. These findings positively answer RQ2 and show that the embedding norm can serve as a simple yet effective optimization objective for graph explanation tasks.

### E.6 Experiments on GIN

To verify the effectiveness of NORMA across different GNN backbones, we adopt a GIN (Xu et al., 2019) with Max Pooling as the target model to be explained. Following a setup similar to GCN, the network has three layers, each consisting of a GINConv–Norm–ReLU block, and the transformation function inside each GIN convolution is implemented as a two-layer MLP. The training protocol is identical to that used for GCN, and the resulting classification performance of the trained GIN is also reported in Table 4.

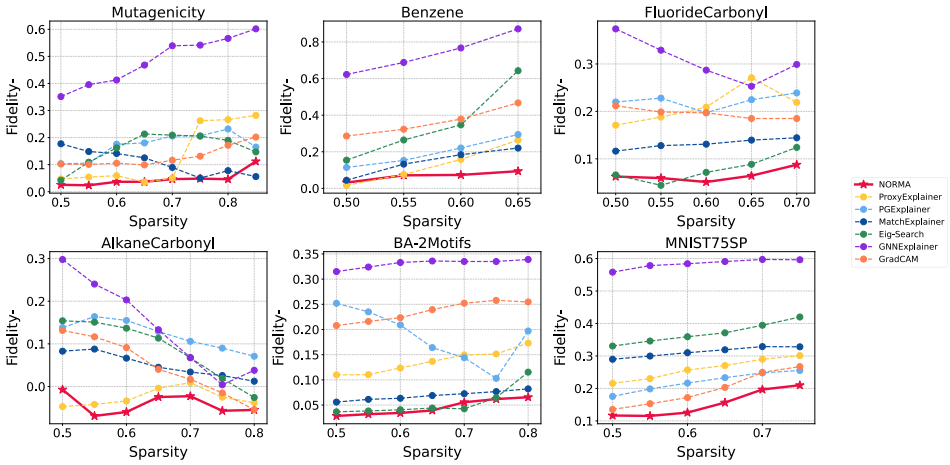



Figure 11: Inverse-Fidelity under different sparsity levels ($\mathrm{sparsity}(s)$: discard ratio, keep top $(1-s)$ edges under a unified edge-based protocol).

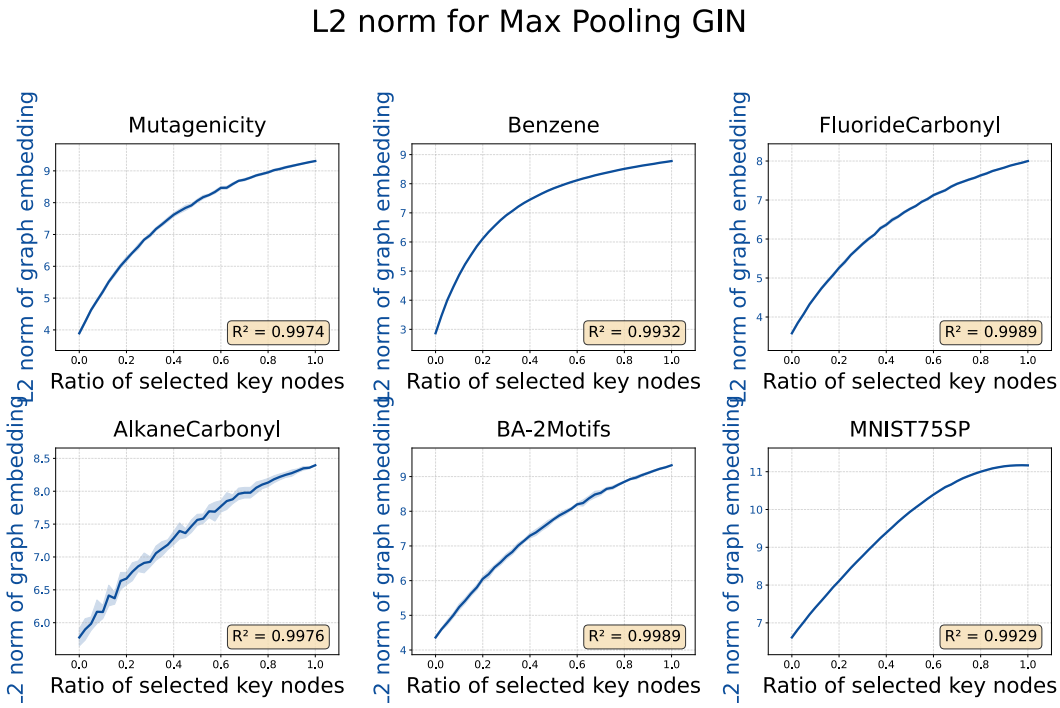



Figure 12: Growth rate of the $\ell_2$ norm of graph embedding for max pooling GIN.

We first examine, on six datasets, how the $\ell_2$ norm of the graph representation varies with the number of important nodes among the selected nodes. As shown in Figure 12, similar to GCN, the norm exhibits logarithmic growth. For each plot, we report in the bottom-right corner the $R^2$ value between the empirical curve and the fitted logarithmic curve; all $R^2$ values exceed 0.99, indicating that the observed trend closely follows a log-shaped curve. We then compare explanation accuracy (edge AUROC) on the same six datasets using GIN with the baselines in GCN experiments. As reported in Table 9, NORMA achieves state-of-the-art results on all datasets. These results suggest that NORMA is not limited to explaining GCNs, but can also be extended to other types of GNNs.

Table 9: Explanation Performance (AUROC) on edges for GIN. The SOTA of previous works is emphasised with underline.

| Method | Mutagenicity | Benzene | Fluoride-Carbonyl | Alkane-Carbonyl | BA-2Motifs | MNIST75SP |
|---|---|---|---|---|---|---|
| GradCAM | 0.606±0.000 | 0.623±0.000 | 0.522±0.000 | 0.512±0.000 | 0.645±0.000 | 0.520±0.000 |
| GNNExplainer | 0.816±0.001 | 0.459±0.003 | 0.487±0.006 | 0.729±0.002 | 0.457±0.008 | 0.500±0.000 |
| PGExplainer | 0.971±0.002 | 0.829±0.034 | 0.630±0.042 | 0.867±0.009 | 0.806±0.017 | 0.640±0.020 |
| Eig-Search | 0.924±0.000 | 0.905±0.001 | 0.817±0.002 | 0.947±0.000 | 0.836±0.001 | 0.562±0.007 |
| MatchExplainer | 0.806±0.001 | 0.848±0.000 | 0.780±0.001 | 0.740±0.004 | 0.618±0.002 | 0.458±0.002 |
| ProxyExplainer | 0.994±0.004 | 0.902±0.045 | 0.864±0.021 | 0.924±0.007 | 0.876±0.006 | 0.557±0.031 |
| NORMA (Ours) | **0.999±0.000** | **0.972±0.003** | **0.905±0.003** | **0.963±0.004** | **0.926±0.002** | **0.701±0.003** |

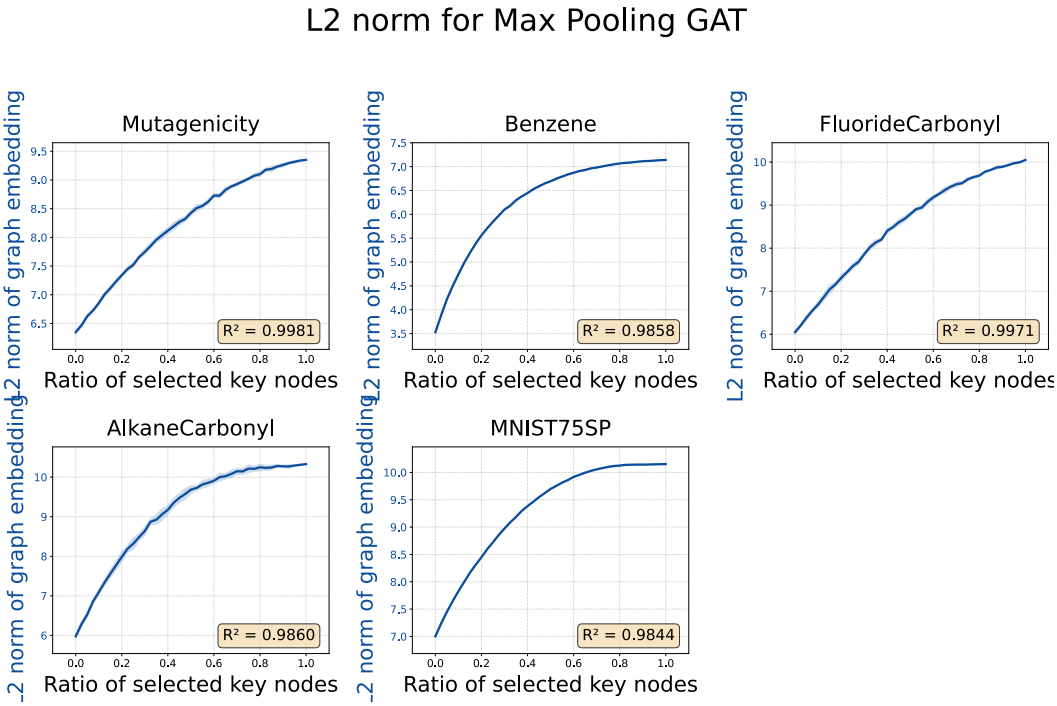



Figure 13: Growth rate of the $\ell_2$ norm of graph embedding for max pooling GAT.

### E.7 Experiments on GAT

For GAT, we conduct experiments analogous to those performed on GIN. Since the GAT model cannot be successfully trained on BA-2Motifs, this part of the evaluation is conducted on the remaining five datasets. As shown in Figure 13, similar to the observations on GCN and GIN, the representation norm under GAT also exhibits a logarithmic growth pattern. We report the $R^2$ value between the empirical curve and the fitted logarithmic curve in the lower-right corner of each plot. All $R^2$ values are greater than 0.98, which further supports the validity of our theoretical analysis.

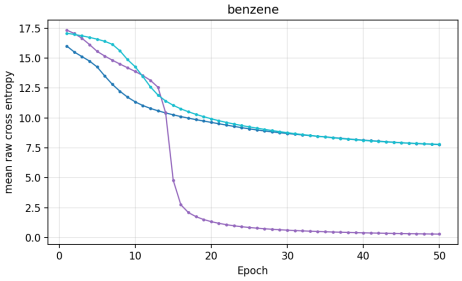
We then evaluate explanation accuracy on the five datasets using GAT as the backbone, and compare NORMA with the same set of applicable baselines in terms of edge AUROC. As reported in Table 10, NORMA achieves state-of-the-art performance across all datasets. Overall, the experiments on GAT further demonstrate the general applicability of NORMA across different GNN architectures.

### E.8 More Results of RQ3

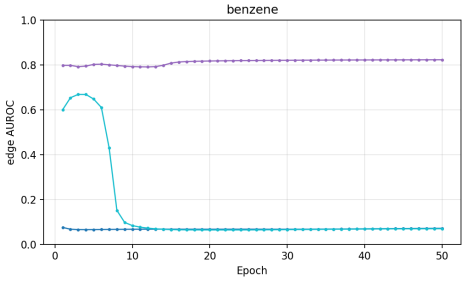
We provide additional results on Benzene and BA-2Motifs to further illustrate the optimization stability of NORMA.

Table 10: Explanation Performance (AUROC) on edges for GAT. The SOTA of previous works is emphasised with underline.

| Method | Mutagenicity | Benzene | Fluoride-Carbonyl | Alkane-Carbonyl | MNIST75SP |
|---|---|---|---|---|---|
| GradCAM | 0.595±0.000 | 0.627±0.000 | 0.620±0.000 | 0.854±0.000 | 0.517±0.000 |
| GNNExplainer | 0.838±0.000 | 0.481±0.002 | 0.692±0.002 | 0.710±0.000 | 0.499±0.000 |
| PGExplainer | 0.948±0.013 | 0.857±0.019 | 0.710±0.023 | 0.904±0.006 | 0.539±0.016 |
| Eig-Search | 0.807±0.002 | 0.725±0.000 | 0.805±0.001 | 0.982±0.000 | 0.552±0.002 |
| MatchExplainer | 0.915±0.002 | 0.784±0.002 | 0.558±0.000 | 0.856±0.003 | 0.549±0.005 |
| ProxyExplainer | 0.975±0.008 | 0.912±0.005 | 0.830±0.001 | 0.986±0.001 | 0.633±0.058 |
| NORMA (Ours) | **0.996±0.000** | **0.918±0.000** | **0.877±0.005** | **0.989±0.001** | **0.683±0.005** |




(a) Cross-entropy trajectories under cross-entropy guidance on Benzene.




(b) AUROC trajectories under cross-entropy guidance on Benzene.

Figure 14: Training dynamics under cross-entropy guidance on Benzene. Only one run converges toa strong solution, while the other two converge to poor solutions, indicating substantially less stable optimization.

First, we present the trends of norm loss and corresponding edge AUROC during optimization using the norm as the loss function on the Benzene dataset in the main text. Here, we supplement the results for the Benzene dataset when driven by cross-entropy loss. As shown in Figures 14a and 14b, when using cross-entropy to train the explainer, the model may either converge to a strong solution or get stuck in a poor one, depending on the initialization, demonstrating instability due to initialization sensitivity.

We conduct similar experiments on the BA-2Motifs dataset. As shown in Figure 15, with 10 random seeds, the norm-driven explainer successfully converges in all 10 runs, while the CE-based explainer successfully converges in only 7 runs, indicating that NORMA's optimization is more stable.

Furthermore, we present the trends of subgraph norm, corresponding cross-entropy (which is also not optimized, only computed), and the corresponding edge AUROC during optimization under the norm loss. In Figures 16a and 16b, we observe similar results as those on the Benzene dataset. Under the norm-driven loss, the subgraph norm increases rapidly, the corresponding cross-entropy decreases rapidly (also only computed, not involved in the optimization process), and the edge AUROC rises quickly. These results further demonstrate that the norm-driven optimization process is more stable and better captures the structural information of the subgraph.

On the BA-2Motifs dataset, the explainer optimized with cross-entropy, as shown in Figures 17a and 17b, also demonstrates instability in the optimization process, which is influenced by different initializations.

### E.9 Case Study Examples

We provide qualitative visualization results to examine whether NORMA identifies semantically meaningful explanatory subgraphs. Figures 18, 19, 20, and 21 show representative examples from Mutagenicity, Benzene, AlkaneCarbonyl, and BA-2Motifs, respectively. For each instance, we compare the explanations produced by NORMA with those from representative baseline explainers.

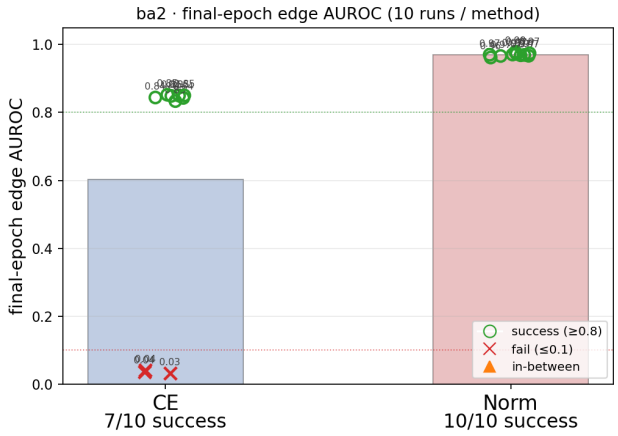
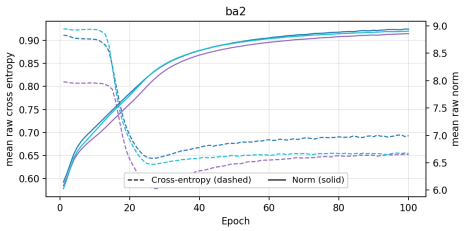
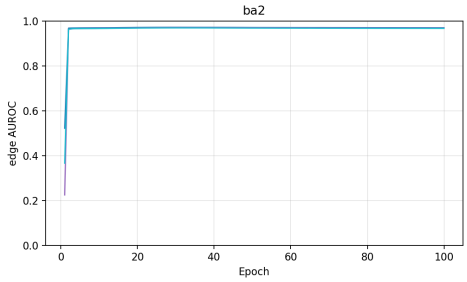



Figure 15: Results over 10 random seeds on BA-2Motifs.

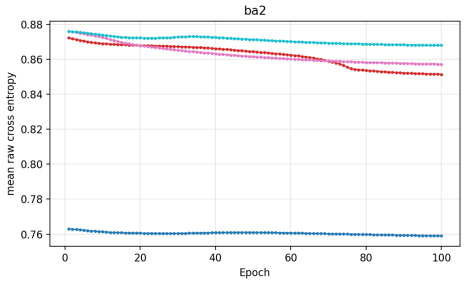



(a) $\ell_2$-norm and cross-entropy trajectories under norm guidance on BA-2Motifs.

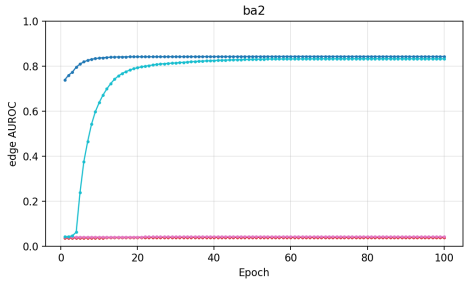



(b) AUROC trajectories under norm guidance on BA-2Motifs.

Figure 16: raining dynamics under norm guidance on BA-2Motifs. The overall behavior is similar to that on Benzene: the norm-guided explainer converges stably across different initializations, and the AUROC improves and stabilizes very quickly




(a) Cross-entropy trajectories under cross-entropy guidance on BA-2Motifs.




(b) AUROC trajectories under cross-entropy guidance on BA-2Motifs.

Figure 17: Training dynamics under cross-entropy guidance on BA-2Motifs. Under poor initialization, cross-entropy may either recover to a strong solution or fail completely, further illustrating its instability and sensitivity to initialization.

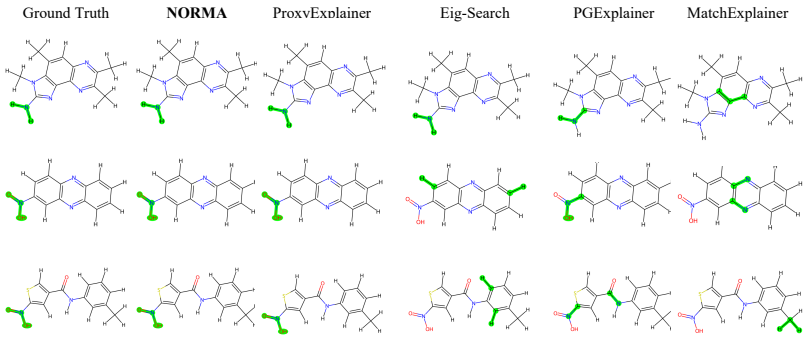



Figure 18: Case Study on Mutagenicity Dataset

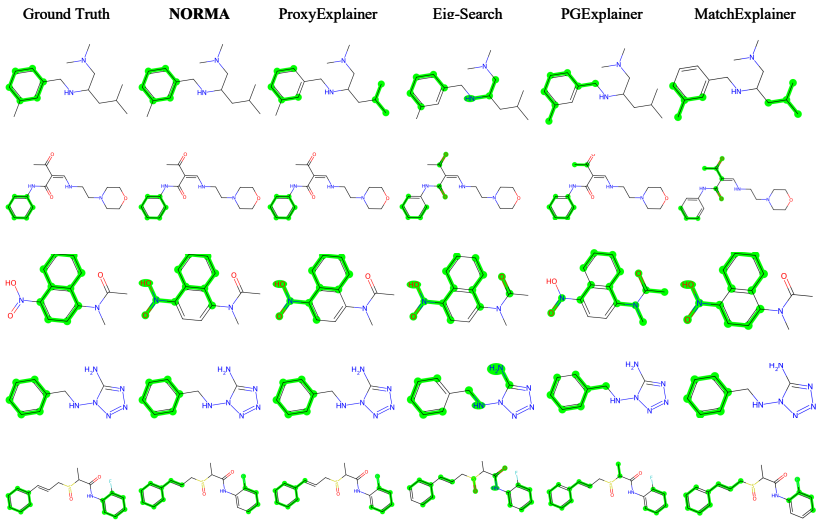



Figure 19: Case Study on Benzene Dataset

The highlighted edges indicate the substructures selected as important by each method, while the ground-truth rationale is provided for reference.

Across these examples, NORMA more consistently recovers the annotated explanatory subgraphs. On molecular datasets, the selected edges often correspond to chemically meaningful functional groups or motifs, such as benzene rings or nitrogen-containing groups. On BA-2Motifs, NORMA also better captures the motif structures responsible for the graph label. These visualizations complement the quantitative AUROC results in Section 6.3, showing that the norm objective can guide the explainer toward compact and semantically relevant explanations.

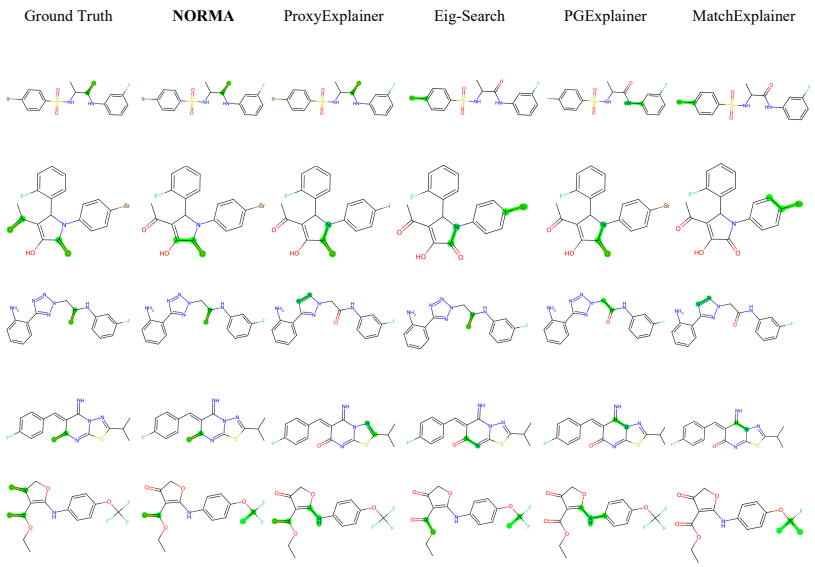



Figure 20: Case Study on Alkane-Carbonyl Dataset

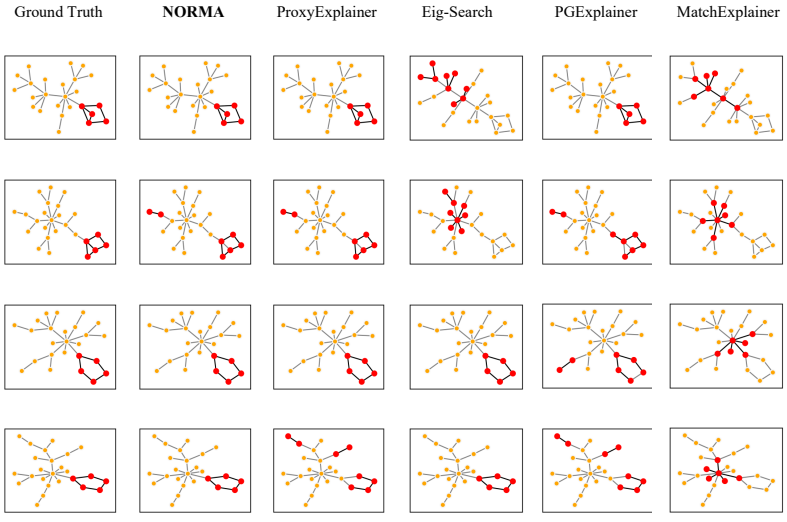



Figure 21: Case Study on BA-2Motifs Dataset

# F Theoretical Support for Growth Rate of Graph Representation Norm

We organize our analysis as follows. In Appendix G, we specify the GCN model used in our theoretical analysis, the low-rank assumption on the weight matrices, and the formal characterization of important and non-important nodes with respect to the principal directions of the weight matrices. In Appendix H, we analyze the "alignment gain" within a single graph convolution layer. In Appendix I, we study the cumulative effect of alignment gain across multiple convolution layers. In Appendix J, we provide a proof of the growth rate of the pooled graph-embedding norm. In Appendix K, we examine how LayerNorm affects the growth rate, and in Appendix L, we analyze the effect of ReLU. Finally, in Appendix M, we briefly discuss the growth behavior for GAT and GIN models.

# G Setup and Low-rank Alignment Assumption

The assumptions introduced below strengthen those in Section 4. They are mainly imposed to facilitate the complete non-asymptotic analysis, particularly for controlling higher-order terms in the derivation. These assumptions are technical in nature and do not introduce additional structural constraints on the problem formulation.

**Model and layers.** Given a graph $G = (V, E)$ with node features $X \in \mathbb{R}^{n\times d}$, denote the GNN to be explained by $f_{\exp}$. For an input graph $G = (V, E)$, the model produces a graph-level representation $z$. In our theoretical analysis, we focus on a GCN architecture. The $\ell$-th layer of $f_{\exp}$ is given by

$$H^{(\ell)} = \sigma\Big(\mathrm{LN}\Big(\hat{A}H^{(\ell-1)}W^{(\ell)}\Big)\Big), \qquad \hat{A} = D^{-\frac{1}{2}}(A+I)D^{-\frac{1}{2}}, \qquad \sigma = \mathrm{ReLU}, \tag{25}$$

with $H^{(0)} = X$. The LayerNorm operation is denoted by

$$\mathrm{LN}(h) = \Gamma^{(\ell)}\frac{h-\mu(h)}{\sigma(h)} + \beta^{(\ell)}, \tag{26}$$

where $\Gamma^{(\ell)}$ and $\beta^{(\ell)}$ are learnable channel-wise scale and bias parameters.

## G.1 Rank-$r$ Low-rank Assumption on the Weight Matrix

Assume that for the $\ell$-th layer weight matrix $W_\ell \in \mathbb{R}^{d_\ell\times d_{\ell-1}}$, there exist two column-orthonormal bases $U_\ell \in \mathbb{R}^{d_\ell\times r}$ and $U_{\ell-1} \in \mathbb{R}^{d_{\ell-1}\times r}$, satisfying $U_\ell^\top U_\ell = U_{\ell-1}^\top U_{\ell-1} = I_r$, and an invertible "gain" matrix $S_\ell \in \mathbb{R}^{r\times r}$ such that

$$\begin{aligned}
&\underbrace{\left\|U_\ell^\top W_\ell U_{\ell-1} - S_\ell\right\|}_{\text{main-kernel alignment error}} \le \tau_\ell,\\
&\underbrace{\left\|U_\ell^\top W_\ell \Pi_{U_{\ell-1}}^{\perp}\right\|}_{\text{right leakage}} \le \delta_\ell,\\
&\underbrace{\left\|\Pi_{U_\ell}^{\perp} W_\ell U_{\ell-1}\right\|}_{\text{left leakage}} \le \delta'_\ell,
\end{aligned} \tag{27}$$

where

$$\Pi_{U_{\ell-1}} := U_{\ell-1}U_{\ell-1}^\top, \qquad \Pi_{U_\ell} := U_\ell U_\ell^\top,$$

and $\tau_\ell$, $\delta_\ell$, and $\delta'_\ell$ are small constants.

After training, the model-utilized features in the input or intermediate representations are aligned with these principal directions. In contrast, components that are not aligned with the principal directions are progressively suppressed during inter-layer propagation, as their energy leaks into the null space or the non-aligned subspace, resulting in smaller representation norms.

**Equivalent factorization.** Equivalently, there exist $U_\ell$, $U_{\ell-1}$, and $S_\ell$ such that

$$W_\ell = U_\ell S_\ell U_{\ell-1}^\top + E_\ell, \tag{28}$$

where the residual satisfies $\|E_\ell\| \le \varepsilon_\ell$ under a chosen matrix norm. Common choices include:

1. the operator, or spectral, norm:
$$\|E_\ell\|_2 \le \varepsilon_\ell,$$
which controls worst-case directional amplification;
2. the Frobenius norm:
$$\|E_\ell\|_F \le \varepsilon_\ell,$$
which controls the total residual energy.

This factorization corresponds to the alignment formulation above through

$$\begin{aligned} \left\|U_\ell^\top E_\ell U_{\ell-1}\right\| &\le \tau_\ell, \\ \left\|U_\ell^\top E_\ell \Pi_{U_{\ell-1}}^\perp\right\| &\le \delta_\ell, \\ \left\|\Pi_{U_\ell}^\perp E_\ell U_{\ell-1}\right\| &\le \delta'_\ell. \end{aligned} \tag{29}$$

**Exact rank-$r$ case.** When $\varepsilon_\ell = 0$, the weight matrix is exactly rank-$r$ in the sense that

$$W_\ell = U_\ell S_\ell U_{\ell-1}^\top,$$

and hence $\mathrm{rank}(W_\ell) \le r$.

**Justification.** This assumption is motivated by extensive empirical evidence that trained neural networks exhibit a pronounced low-rank or rank-diminishing bias in their learned weight matrices. Across architectures, the spectrum of weight operators becomes highly skewed after training, with only a small number of dominant singular directions capturing most of the input–output transformation capacity (Huh et al., 2021; Feng et al., 2022). Such behavior has been observed both in fully connected networks and message-passing architectures, and persists even when exact rank deficiency is absent, suggesting an effective low-rank structure rather than a strict constraint. Modeling the weight matrix as a low-rank operator therefore captures the dominant geometry induced by training dynamics while abstracting away weak, noisy directions that contribute marginally to representation propagation (Galanti and Poggio, 2022; Gunasekar et al., 2017).

### G.2 Rank-$r$ Modeling of Important and Non-important Features

For the representation of any node $j$, let $h_j^{(\ell)} \in \mathbb{R}^{d_\ell}$. We define its coefficients on the principal subspace as

$$\alpha_j^{(\ell-1)} := U_{\ell-1}^\top h_j^{(\ell-1)} \in \mathbb{R}^r, \qquad h_j^{(\ell-1)} = U_{\ell-1}\alpha_j^{(\ell-1)} + r_j^{(\ell-1)}, \qquad r_j^{(\ell-1)} \perp \mathrm{span}(U_{\ell-1}). \tag{30}$$

Given an important node set $I$ of size $k$ and its complement $I^{\mathrm{c}}$ of size $s-k$, we characterize them statistically as follows:

1. **Important nodes.** The coefficients $\alpha_j^{(\ell-1)}$ exhibit a nonzero mean direction. Specifically, there exists $\mu_I \in \mathbb{R}^r$ such that
$$\mathbb{E}\left[\alpha_j^{(\ell-1)}\right] = \mu_I,$$
and their covariance is bounded as
$$\mathrm{Cov}\left(\alpha_j^{(\ell-1)}\right) \preceq \Sigma_I.$$
2. **Non-important nodes.** The coefficients $\alpha_j^{(\ell-1)}$ have approximately zero mean or are nearly orthogonal to $\mu_I$; for example,
$$\left\|\mathbb{E}\left[\alpha_j^{(\ell-1)}\right]\right\| \le \epsilon_\mu.$$
Their covariance satisfies
$$\mathrm{Cov}\left(\alpha_j^{(\ell-1)}\right) \preceq \Sigma_N.$$

3. **Residual components.** The residual norm has bounded second moment, i.e., the energy of $r_j^{(\ell-1)}$ is controlled. Different nodes are assumed to be only weakly dependent or mixing, which will later be used for variance bounds.

### G.2.1 Detailed Modeling of Important and Non-important Features

**Decomposition and notation.** At layer $\ell$, for any node $j$, the representation $h_j^{(\ell)} \in \mathbb{R}^{d_\ell}$ is decomposed with respect to the principal subspace as

$$\alpha_j^{(\ell-1)} := U_{\ell-1}^\top h_j^{(\ell-1)} \in \mathbb{R}^r, \qquad h_j^{(\ell-1)} = U_{\ell-1}\alpha_j^{(\ell-1)} + r_j^{(\ell-1)}, \qquad r_j^{(\ell-1)} \perp \operatorname{span}(U_{\ell-1}). \tag{31}$$

Here $U_{\ell-1} \in \mathbb{R}^{d_{\ell-1}\times r}$ is a column-orthonormal basis, and $r_j^{(\ell-1)}$ denotes the orthogonal residual. We denote the important node set by $I$, with $|I| = k$, and its complement by $I^{\mathrm{c}}$, with $|I^{\mathrm{c}}| = s - k$.

### G.2.2 Informative, or Important, Nodes

For informative nodes, we impose the following conditions:

1. **Mean direction.** There exists $\mu_I \in \mathbb{R}^r$ such that
$$\mathbb{E}\Big[\alpha_j^{(\ell-1)} \mid j \in I\Big] = \mu_I, \qquad \|\mu_I\| \geq \mu_{\min} > 0. \tag{32}$$
For compatibility with ReLU and nonnegative activations, one may, without loss of generality, fix the signs or ordering of the columns of $U_{\ell-1}$ so that the dominant coordinates of $\mu_I$ are nonnegative. This is only for notational convenience; the lower bounds do not depend on this choice.
2. **Covariance and tail bounds.** The covariance satisfies
$$\operatorname{Cov}\Big(\alpha_j^{(\ell-1)} \mid j \in I\Big) \preceq \Sigma_I, \qquad \|\Sigma_I\|_2 \leq \sigma_I^2. \tag{33}$$
When tail bounds are needed, we assume light-tailed behavior of $\alpha_j^{(\ell-1)}$, such as sub-Gaussian tails, or at least bounded second moments.
3. **Residual energy.** The orthogonal residuals satisfy
$$\mathbb{E}\Big\|r_j^{(\ell-1)}\Big\|^2 \leq \rho_\perp, \qquad \frac{\mathbb{E}\Big\|r_j^{(\ell-1)}\Big\|}{\mathbb{E}\Big\|U_{\ell-1}\alpha_j^{(\ell-1)}\Big\|} \leq \lambda_\perp \ll 1, \tag{34}$$
where $\rho_\perp$ and $\lambda_\perp$ control the leakage strength and will appear together with $\delta_\ell$ in subsequent error bounds.

**Justification.** This assumption reflects the observation that features consistently utilized by a trained network tend to align with the dominant directions of the learned weight matrices. Prior work shows that, during training, representations associated with task-relevant or model-utilized signals progressively concentrate along a small set of principal directions, leading to stable directional alignment between features and weights (Ji and Telgarsky, 2020; Qiu et al., 2025a). From a representation learning perspective, these dominant directions encode the knowledge acquired by the model and act as attractors for repeatedly exploited features (Papyan et al., 2020; Rangamani et al., 2023). As a result, modeling key features as having a nonzero mean projection onto the principal subspace provides a concise abstraction of their coherent and reinforcing contribution to downstream representations.

### G.2.3 Non-informative, or Non-important, Nodes

For non-informative nodes, we impose the following conditions:

1. **Near-zero or near-orthogonal mean.** The mean projection of non-important nodes onto the principal subspace is either small or nearly orthogonal to the informative direction:
$$\Big\|\mathbb{E}\Big[\alpha_j^{(\ell-1)} \mid j \in I^{\mathrm{c}}\Big]\Big\| \leq \epsilon_\mu \qquad \text{or} \qquad \frac{\Big|\Big\langle \mathbb{E}\Big[\alpha_j^{(\ell-1)} \mid j \in I^{\mathrm{c}}\Big], \mu_I \Big\rangle\Big|}{\|\mu_I\|} \leq \epsilon_{\mathrm{ortho}}, \tag{35}$$
where $\epsilon_\mu$ and $\epsilon_{\mathrm{ortho}}$ are small constants.

2. **Covariance and tail bounds.** The covariance satisfies

$$\mathrm{Cov}\Big(\alpha_j^{(\ell-1)} \mid j \in I^{\mathrm{c}}\Big) \preceq \Sigma_N, \qquad \|\Sigma_N\|_2 \le \sigma_N^2. \tag{36}$$

3. **Residual energy.** As in the informative-node case, the orthogonal residuals satisfy

$$\mathbb{E}\Big\|r_j^{(\ell-1)}\Big\|^2 \le \rho_\perp. \tag{37}$$

**Justification.** In contrast, features that are not systematically exploited by the network do not develop stable alignment with the dominant weight directions. Empirical studies suggest that such non-key features exhibit fluctuating orientations across samples and layers, lacking consistent structure in the principal subspace identified by the trained weights (Imani et al., 2021). Consequently, their projections onto this subspace tend to cancel out in expectation rather than accumulate. Approximating non-key features as zero-mean and isotropic with respect to the principal subspace thus provides a conservative abstraction of their aggregate behavior, emphasizing their limited contribution to representation growth (Kang et al., 2024).

### G.2.4 Quantities and Parameters Used in Subsequent Derivations

We summarize the main quantities used in the following derivations:

1. **Signal strength.** The quantity
$$\|S_\ell \mu_I\|$$
measures the magnitude of the principal component after one-layer propagation.
2. **Noise scale.** The quantities
$$\sigma_I^2, \qquad \sigma_N^2,$$
together with the mixing strength, determine how the variance terms grow with the number of nodes.
3. **Leakage and alignment.** The quantities
$$\rho_\perp, \qquad \lambda_\perp,$$
together with the alignment and leakage constants
$$\tau_\ell, \qquad \delta_\ell,$$
enter the subsequent error bounds.

## H Single-layer Propagation: Alignment Gain

**Lemma 1** (Single-layer principal-direction gain with LayerNorm and ReLU)**.** *Suppose the principal-direction decomposition at layer $\ell$ is defined as above. Consider the linear diffusion*

$$H^{(\ell)} = \hat{A} H^{(\ell-1)} W_\ell. \tag{38}$$

*Let the intermediate pre-normalization and pre-activation variable after aggregation be*

$$h_i^{(\ell)} := \sum_j \hat{A}_{ij} W_\ell h_j^{(\ell-1)}, \tag{39}$$

*where $\hat{A}$ is a nonnegative row-stochastic aggregation kernel, such as normalized adjacency or attention weights. Then, in the principal-direction coordinates with basis $U_\ell$, we have*

$$U_\ell^\top h_i^{(\ell)} \approx S_\ell \sum_j \hat{A}_{ij} \alpha_j^{(\ell-1)} + \varepsilon_i^{(\ell-1)}, \tag{40}$$

*where the error term satisfies*

$$\Big\|\varepsilon_i^{(\ell-1)}\Big\| \le \tau_\ell \sum_j \hat{A}_{ij} \Big\|\Pi_{U_{\ell-1}} h_j^{(\ell-1)}\Big\| + \delta_\ell \sum_j \hat{A}_{ij} \Big\|\Pi_{U_{\ell-1}}^{\perp} h_j^{(\ell-1)}\Big\| + \eta_\ell. \tag{41}$$

*Here $\eta_\ell$ aggregates approximation errors due to biases, nonlinear truncation, LayerNorm fluctuations, and related module-level perturbations.*

Equivalently, writing the relation coordinate-wise, let

$$U_\ell = [u_{\ell,1}, \dots, u_{\ell,r}], \qquad U_{\ell-1} = [u_{\ell-1,1}, \dots, u_{\ell-1,r}].$$

For any $p = 1, \dots, r$,

$$u_{\ell,p}^\top h_i^{(\ell)} \approx \sum_{q=1}^{r} (S_\ell)_{pq} \sum_j \hat{A}_{ij} \left( u_{\ell-1,q}^\top h_j^{(\ell-1)} \right) + \varepsilon_{i,p}^{(\ell-1)}. \tag{42}$$

The vector $\varepsilon_i^{(\ell-1)} \in \mathbb{R}^r$ collects the principal-subspace alignment residual, orthogonal leakage, and module-level approximation errors. These correspond respectively to

$$U_\ell^\top W_\ell \Pi_{U_{\ell-1}} - S_\ell U_{\ell-1}^\top, \qquad U_\ell^\top W_\ell \Pi_{U_{\ell-1}}^\perp,$$

and small perturbations from LayerNorm and activation.

After the affine transformation in LayerNorm and ReLU gating, aligned directions are more likely to remain positive and stronger. Therefore, there exists a positive channel vector $g_\ell \in \mathbb{R}_{>0}^r$, absorbing $\Gamma^{(\ell)} := \mathrm{diag}(g_\ell)$ and typical effects of $\sigma(\cdot)$, such that for any $p = 1, \dots, r$,

$$u_{\ell,p}^\top h_i^{(\ell)} \approx g_{\ell,p} \cdot \max \left\{ 0, \sum_{q=1}^{r} (S_\ell)_{pq} \sum_j \hat{A}_{ij} \left( u_{\ell-1,q}^\top h_j^{(\ell-1)} \right) \right\}. \tag{43}$$

### H.1 Detailed Derivation

We now provide the detailed derivation of Lemma 1.

1. **Projection onto the output principal subspace.** By projecting the intermediate representation onto the output principal subspace spanned by $U_\ell$, we obtain

$$\begin{aligned} U_\ell^\top h_i^{(\ell)} &= U_\ell^\top \sum_j \hat{A}_{ij} W_\ell h_j^{(\ell-1)} \\ &= \sum_j \hat{A}_{ij} U_\ell^\top W_\ell h_j^{(\ell-1)}. \end{aligned} \tag{44}$$

2. **Decomposition into principal and orthogonal components.** We decompose the input representation into its principal-subspace component and orthogonal residual:

$$\begin{aligned} h_j^{(\ell-1)} &= \Pi_{U_{\ell-1}} h_j^{(\ell-1)} + \Pi_{U_{\ell-1}}^\perp h_j^{(\ell-1)} \\ &= U_{\ell-1} \alpha_j^{(\ell-1)} + r_j^{(\ell-1)}. \end{aligned} \tag{45}$$

3. **Application of the rank-$r$ alignment assumption.** Using the above decomposition,

$$h_j^{(\ell-1)} = \Pi_{U_{\ell-1}} h_j^{(\ell-1)} + \Pi_{U_{\ell-1}}^\perp h_j^{(\ell-1)},$$

we have

$$U_\ell^\top W_\ell h_j^{(\ell-1)} = U_\ell^\top W_\ell \left( \Pi_{U_{\ell-1}} + \Pi_{U_{\ell-1}}^\perp \right) h_j^{(\ell-1)}. \tag{46}$$

For the first term, we perform an add-and-subtract matching:

$$\begin{aligned} U_\ell^\top W_\ell \Pi_{U_{\ell-1}} h_j^{(\ell-1)} = &\underbrace{\left( U_\ell^\top W_\ell \Pi_{U_{\ell-1}} - S_\ell U_{\ell-1}^\top \right)}_{\text{alignment error}} \Pi_{U_{\ell-1}} h_j^{(\ell-1)} \\ &+ S_\ell U_{\ell-1}^\top \Pi_{U_{\ell-1}} h_j^{(\ell-1)}. \end{aligned} \tag{47}$$

Therefore,

$$\begin{aligned} U_\ell^\top W_\ell h_j^{(\ell-1)} = &\underbrace{\left( U_\ell^\top W_\ell \Pi_{U_{\ell-1}} - S_\ell U_{\ell-1}^\top \right) \Pi_{U_{\ell-1}} h_j^{(\ell-1)}}_{\text{principal-subspace alignment error}} \\ &+ \underbrace{S_\ell U_{\ell-1}^\top \Pi_{U_{\ell-1}} h_j^{(\ell-1)}}_{\text{principal term}} + \underbrace{U_\ell^\top W_\ell \Pi_{U_{\ell-1}}^\perp h_j^{(\ell-1)}}_{\text{leakage term}}. \end{aligned} \tag{48}$$

4. **Combining alignment error and leakage.** Let

$$\alpha_j^{(\ell-1)} := U_{\ell-1}^\top h_j^{(\ell-1)}.$$

We define the accumulated error term as

$$\begin{aligned}\varepsilon_i^{(\ell-1)} := \sum_j \hat{A}_{ij} \Big[ \left(U_\ell^\top W_\ell \Pi_{U_{\ell-1}} - S_\ell U_{\ell-1}^\top\right) \Pi_{U_{\ell-1}} h_j^{(\ell-1)} \\ + U_\ell^\top W_\ell \Pi_{U_{\ell-1}}^\perp h_j^{(\ell-1)} \Big] + \eta_\ell.\end{aligned} \tag{49}$$

Taking the operator norm and using the rank-$r$ alignment bounds

$$\left\|U_\ell^\top W_\ell \Pi_{U_{\ell-1}} - S_\ell U_{\ell-1}^\top\right\| \le \tau_\ell, \qquad \left\|U_\ell^\top W_\ell \Pi_{U_{\ell-1}}^\perp\right\| \le \delta_\ell,$$

yields

$$\begin{aligned}\left\|\varepsilon_i^{(\ell-1)}\right\| \le \tau_\ell \sum_j \hat{A}_{ij} \left\|\Pi_{U_{\ell-1}} h_j^{(\ell-1)}\right\| \\ + \delta_\ell \sum_j \hat{A}_{ij} \left\|\Pi_{U_{\ell-1}}^\perp h_j^{(\ell-1)}\right\| + \eta_\ell.\end{aligned} \tag{50}$$

5. **Resulting single-layer expression.** Substituting the definition of $\varepsilon_i^{(\ell-1)}$ back into the projected representation gives

$$U_\ell^\top h_i^{(\ell)} = S_\ell \sum_j \hat{A}_{ij} \alpha_j^{(\ell-1)} + \varepsilon_i^{(\ell-1)}. \tag{51}$$

Since $\hat{A}$ is row-stochastic, $\sum_j \hat{A}_{ij} = 1$. The term $\eta_\ell$ absorbs small perturbations from biases, activation, and LayerNorm.

# I Layered Expression: Effective Nonnegative Propagation Operator $P$ and Total Gain $G$

**Lemma 2** (Layerwise factorization)**.** *After $L$ layers of propagation, the representations on the rank-$r$ principal subspace satisfy*

$$\alpha^{(L)} \approx G\left(P\alpha^{(0)}\right) + \varepsilon^{(0\to L)}, \tag{52}$$

*where*

$$G = \prod_{\ell=1}^{L} \Gamma_\ell S_\ell$$

*is the total channel-wise gain, and*

$$P = \prod_{\ell=1}^{L} \hat{A}^{(\ell)}$$

*is the effective node-wise propagation operator. The operator $P$ is nonnegative and row-stochastic.*

Here $\varepsilon^{(0\to L)}$ denotes the accumulated approximation error over the $0 \to L$ propagation. The matrix $G$ acts along the channel, or principal-subspace, dimension, while $P$ acts along the node dimension by propagating coefficients across the graph.

Equivalently, if the node coefficients are stacked row-wise as $\alpha^{(\ell)} \in \mathbb{R}^{n\times r}$, the same relation can be written as

$$\alpha^{(L)} \approx \left(P\alpha^{(0)}\right) G^\top + \varepsilon^{(0\to L)}. \tag{53}$$

### I.1 Derivation of Lemma 2

#### I.1.1 Layer-wise Form

We first rewrite Lemma 1 in matrix form. For each node $j$, define

$$\alpha_j^{(\ell-1)} := U_{\ell-1}^\top h_j^{(\ell-1)} \in \mathbb{R}^r, \qquad M^{(\ell)} := \Gamma^{(\ell)} S^{(\ell)} \in \mathbb{R}^{r\times r}. \tag{54}$$

Here $S^{(\ell)}$ denotes the rank-$r$ principal-subspace gain introduced in the low-rank alignment assumption, and

$$\Gamma^{(\ell)} := \mathrm{diag}(g_\ell)$$

is the nonnegative channel-wise scaling induced by activation and normalization.

Let $\hat{A}^{(\ell)}$ denote the nonnegative row-stochastic aggregation kernel at layer $\ell$, such as normalized adjacency or attention weights. Then the layer-wise propagation in the principal-subspace coordinates can be written as

$$\alpha_i^{(\ell)} \approx M^{(\ell)} \sum_j \hat{A}_{ij}^{(\ell)} \alpha_j^{(\ell-1)} + \varepsilon_i^{(\ell-1)}. \tag{55}$$

The perturbation term $\varepsilon_i^{(\ell-1)}$ collects the small errors arising from the alignment error $\tau_\ell$, the leakage term $\delta_\ell$, and the module-approximation term $\eta_\ell$. An upper bound for this perturbation was provided in Lemma 1.

#### I.1.2 Overall Form after Stacking $L$ Layers

We iterate the layer-wise relation from $\ell = 1$ to $\ell = L$. Since each $M^{(\ell)}$ acts on the channel dimension and each $\hat{A}^{(\ell)}$ acts on the node dimension, and since $M^{(\ell)}$ is independent of node indices, the channel-wise and node-wise effects can be separated. This yields

$$\alpha_i^{(L)} \approx G \sum_j P_{ij} \alpha_j^{(0)} + \varepsilon_i^{(0\to L)}. \tag{56}$$

Equivalently, in matrix form,

$$\alpha^{(L)} \approx \left(P\alpha^{(0)}\right) G^\top + \varepsilon^{(0\to L)}. \tag{57}$$

The two operators are defined as follows:

1. **Total channel-wise gain.** Define

$$G := M^{(L)} M^{(L-1)} \cdots M^{(1)} \in \mathbb{R}^{r\times r}. \tag{58}$$

   Its norm is bounded by

$$\|G\| = \left\| \prod_{\ell=1}^{L} M^{(\ell)} \right\| \le \left( \prod_{\ell=1}^{L} \|\Gamma^{(\ell)}\| \right) \left( \prod_{\ell=1}^{L} \|S_\ell\| \right). \tag{59}$$

   The matrix $G$ coherently multiplies or rotates the useful directions on the rank-$r$ principal subspace, such as $\mu_I$, across layers.

2. **Effective node-wise propagation operator.** Define

$$P := \hat{A}^{(L)} \hat{A}^{(L-1)} \cdots \hat{A}^{(1)}. \tag{60}$$

   If all layers share the same aggregation kernel, i.e., $\hat{A}^{(\ell)} \equiv \hat{A}$, then $P = \hat{A}^L$. Since each $\hat{A}^{(\ell)}$ is nonnegative and row-stochastic, we have $P \ge 0$ elementwise and

$$P\mathbf{1} = \mathbf{1}, \qquad \|P\|_{\infty\to\infty} = 1, \qquad \|P\|_{1\to 1} \le 1. \tag{61}$$

   Thus, node-level propagation is nonnegative and mass-preserving or non-expansive.

**Accumulated perturbation.** For compactness, denote

$$G_{a:b} := \prod_{m=a}^{b} M^{(m)}, \qquad P_{a:b} := \prod_{m=a}^{b} \hat{A}^{(m)},$$

where $G_{a:b} = I_r$ and $P_{a:b} = I$ when $a > b$. Then the accumulated perturbation satisfies

$$\left\|\varepsilon_i^{(0\to L)}\right\| \le \sum_{t=0}^{L-1} \|G_{t+1:L-1}\| \sum_j (P_{t+1:L-1})_{ij} \Big(\tau_t \left\|\Pi_{U_t} h_j^{(t)}\right\| + \delta_t \left\|\Pi_{U_t}^{\perp} h_j^{(t)}\right\| + \eta_t\Big). \tag{62}$$

Since each $P_{t+1:L-1}$ is row-stochastic, the inner sum over nodes can be further simplified into layer-wise constants and expectation-like quantities.

# J Pooled Graph Embedding and Norm Growth with $k$

## J.1 Operator Expression for Node Pooling

**Matrix notation and stacked-layer conclusion.** Since the dominant signal lies in the principal subspace, we analyze pooling operators applied to the projected representations $\alpha^{(\ell)} = U_\ell^\top h^{(\ell)}$, while noting that the orthogonal components contribute only bounded perturbations.

Assume that the graph has $n$ nodes. We stack the principal-subspace coordinates row-wise into the matrix

$$\alpha^{(\ell)} \in \mathbb{R}^{n\times r}, \qquad \text{where the } i\text{-th row is } \alpha_i^{(\ell)} \in \mathbb{R}^r. \tag{63}$$

From the layerwise factorization above, ignoring small perturbations, we have

$$\alpha^{(L)} \approx \left(P\alpha^{(0)}\right) G^\top, \tag{64}$$

where

$$P = \hat{A}^{(L-1)} \cdots \hat{A}^{(0)} \in \mathbb{R}^{n\times n}$$

is nonnegative and row-stochastic, and

$$G = \prod_\ell \Gamma_\ell S_\ell \in \mathbb{R}^{r\times r}$$

acts on the channel dimension, or equivalently on the rank-$r$ principal subspace.

### J.1.1 General Pooling Operators

We write the graph embedding as a row-wise aggregation of the node representation matrix:

$$z = \mathcal{P}_S\left(\alpha^{(L)}\right) \in \mathbb{R}^r, \tag{65}$$

where $S \subseteq \{1, \dots, n\}$ is the set of pooled nodes, and

$$\mathcal{P}_S : \mathbb{R}^{n\times r} \to \mathbb{R}^r$$

is the pooling operator. Common choices include:

1. **Sum pooling.**
$$\mathcal{P}_S^{\text{sum}}(A) = \mathbf{1}_S^\top A. \tag{66}$$
2. **Mean pooling.**
$$\mathcal{P}_S^{\text{mean}}(A) = \frac{1}{|S|}\mathbf{1}_S^\top A. \tag{67}$$
3. **Max pooling.** For coordinate-wise max pooling,
$$[\mathcal{P}_S^{\max}(A)]_c = \max_{i\in S} A_{i,c}. \tag{68}$$

### J.1.2 Operator Form of Linear Pooling

For any linear pooling operator $\mathcal{P}_S(A) = w^\top A$, where sum and mean pooling correspond to $w = \mathbf{1}_S$ and $w = \mathbf{1}_S/|S|$, respectively, substituting the stacked-layer factorization gives

$$\begin{aligned} z = w^\top \alpha^{(L)} &\approx w^\top \left(P\alpha^{(0)}\right) G^\top \\ &= \underbrace{\left(w^\top P\right)}_{\text{effective arrival distribution } \pi_S^\top} \alpha^{(0)} G^\top. \end{aligned} \tag{69}$$

Define

$$\pi_S^\top := w^\top P \in \mathbb{R}^{1\times n} \tag{70}$$

as the effective arrival distribution from the pooling perspective. Since $P$ is nonnegative and row-stochastic, $\pi_S$ is nonnegative and satisfies $\|\pi_S\|_1 = \|w\|_1$. In particular, for mean pooling, $\|\pi_S\|_1 = 1$. Therefore,

$$z \approx \left(\pi_S^\top \alpha^{(0)}\right) G^\top. \tag{71}$$

This expression decouples node propagation, represented by $\pi_S^\top$, from channel gain, represented by $G$. Specifically,

1. for sum pooling,
$$\pi_S^\top = \mathbf{1}_S^\top P;$$
2. for mean pooling,
$$\pi_S^\top = \frac{1}{|S|}\mathbf{1}_S^\top P.$$

**Interpretation for informative and non-informative nodes.** If the informative node set is $I$, then

$$\pi_S(I) := \sum_{j\in I} \pi_{S,j} \tag{72}$$

denotes the mass from $I$ that reaches the pooled region $S$. More specifically, $\pi_c(i)$ denotes the effective contribution of node $i$ along dimension $c$, and for a node set $I$, we define

$$\pi_c(I) := \sum_{i\in I} \pi_c(i). \tag{73}$$

The informative mean direction $\mu_I$ is mapped by the channel gain to $G\mu_I$. Hence, the magnitude of the principal term scales as

$$\pi_S(I) \cdot \|G\mu_I\|.$$

### J.1.3 Lower Bound for Max Pooling

If, after the effect of the $\Gamma_\ell$/ReLU processing, $\alpha^{(L)}$ is nonnegative on the principal channels, then coordinate-wise we have

$$\mathcal{P}_S^{\max}\left(\alpha^{(L)}\right) \succeq \mathcal{P}_S^{\text{mean}}\left(\alpha^{(L)}\right) \approx \left(\frac{1}{|S|}\mathbf{1}_S^\top P\alpha^{(0)}\right) G^\top, \tag{74}$$

where $\succeq$ denotes elementwise, or coordinate-wise, greater-than-or-equal-to.

### J.1.4 Propagation of Error Terms to the Pooling Output

Let $\varepsilon^{(\ell)}$ denote the small perturbation introduced at layer $\ell$. Under linear pooling, the accumulated error satisfies

$$\left\| z - \left(\pi_S^\top \alpha^{(0)}\right) G^\top \right\| \le \|w\|_1 \sum_\ell \left[ (\tau_\ell + \delta_\ell)\, \mathbb{E}\left\| h^{(\ell)} \right\| + \eta_\ell \right]. \tag{75}$$

Here $\|w\|_1 = |S|$ for sum pooling and $\|w\|_1 = 1$ for mean, or more generally weighted, pooling. This factor reflects how the pooling operator amplifies layer-wise errors.

For nonlinear pooling, we use the coordinate-wise lower bound

$$\mathcal{P}_S^{\max} \succeq \mathcal{P}_S^{\text{mean}}$$

to transfer the same-order error bound to the max-pooling case.

### J.2 Notation: Coverage Coefficients and Quadratic-mass Constant

We introduce several quantities that will be used to characterize how informative nodes contribute to the pooled graph embedding.

1. **Row coverage for sum and mean pooling.** Define

$$c_P(S, I) := \min_{i \in S} \sum_{j \in I} P_{ij}, \qquad c_P(S, I) \in [0, 1]. \tag{76}$$

   This coefficient guarantees that each receiver node $i \in S$ receives at least $c_P(S, I)$ mass from the informative set $I$. Consequently, letting

$$\pi_S(I) := \sum_{j \in I} \pi_{S,j},$$

   we have the universal lower bound

$$\pi_S(I) = w^\top P \mathbf{1}_I \geq \|w\|_1 c_P(S, I). \tag{77}$$

   This quantity is used when the principal term is written as

$$\|\mathbb{E}z\|^2 = \|G\mu_I\|^2 \pi_S(I)^2.$$

   Combining this expression with the lower bound above gives $\Theta(k^2)$ growth for sum pooling, with the appropriate proportional correction for mean or weighted pooling.

2. **Column-maximum coverage for max pooling.** Define

$$c_P^\star(S, I) := \inf_{j \in I} \max_{i \in S} P_{ij}, \qquad c_P^\star(S, I) \in [0, 1]. \tag{78}$$

   This coefficient guarantees that each critical node $j \in I$ has at least one receiver node $i \in S$ that receives it with strength at least $c_P^\star(S, I)$. It is used in the max-pooling analysis, where the Gumbel location parameter may be taken as

$$\mu_c = \mu_c^\varepsilon + (G\mu_I)_c c_P^\star(S, I),$$

   which leads to

$$\mathbb{E}[z_c] = \Theta(\log k).$$

3. **Quadratic-mass constant for variance terms.** Define

$$C_P(S) := \|\pi_S\|_\infty = \max_j \pi_{S,j}, \qquad C_P(S) \in [0, \|w\|_1]. \tag{79}$$

   Then, for any subset $T \subseteq \{1, \ldots, n\}$, we have

$$\sum_{j \in T} \pi_{S,j}^2 \leq C_P(S)^2 |T|. \tag{80}$$

   Hence, the variance terms for sum and mean pooling scale at most linearly with the number of nodes. This bound will be used later to control $\mathrm{Var}(z)$.

4. **Effective sample size for max-pooling extrema.** To quantify how many informative nodes effectively participate in the max competition, fix a threshold $\tau \in (0, 1]$ and define

$$k_{\text{eff}}(S, I; \tau) := \sum_{j \in I} \mathbf{1}\left\{ \max_{i \in S} P_{ij} \geq \tau \right\}. \tag{81}$$

   Under the covering condition $c_P^\star(S, I) > 0$, choosing a constant threshold $\tau$ yields

$$k_{\text{eff}}(S, I; \tau) = \Theta(k).$$

### J.3 Sum/Mean Pooling: $\|z\|_2^2$ Grows as $k^2$

**Starting point: layered factorization and linear pooling.** Stack the principal-subspace coordinates row-wise into the matrix $\alpha^{(0)} \in \mathbb{R}^{n\times r}$. From the layered factorization, where node propagation $P$ and channel gain $G$ are decoupled, for any linear pooling weight $w$ we have

$$\begin{aligned} z = w^\top \alpha^{(L)} &\approx w^\top \left(P\alpha^{(0)}\right) G^\top \\ &= \left(\pi_S^\top \alpha^{(0)}\right) G^\top, \qquad \pi_S^\top := w^\top P. \end{aligned} \tag{82}$$

For sum pooling, $w = \mathbf{1}_S$ and hence

$$\pi_S^\top = \mathbf{1}_S^\top P,$$

which can be interpreted as the effective arrival distribution from the whole graph to the pooled region $S$. This distribution is nonnegative and controlled by the row-stochasticity of $P$. For mean pooling, one replaces $\mathbf{1}_S$ by $\mathbf{1}_S/s$.

**Separation into informative and non-informative components.** Let the informative node set be $I$, with $|I| = k$. We decompose the initial coefficients as

$$\alpha_j^{(0)} = \begin{cases} \mu_I + \xi_j, & j \in I, \qquad \mathbb{E}[\xi_j] = 0, \quad \mathrm{Cov}(\xi_j) \preceq \Sigma_I, \\ \eta_j, & j \notin I, \qquad \mathbb{E}[\eta_j] = 0, \quad \mathrm{Cov}(\eta_j) \preceq \Sigma_N. \end{cases} \tag{83}$$

We work under weak dependence and light-tailed, sub-Gaussian-type behavior sufficient to obtain the required variance upper bounds. Introduce the coverage and upper-bound constants $c_P$ and $C_P$ such that

$$\sum_{j\in I} \pi_{S,j} \geq c_P(S,I)k, \qquad \sum_{j\in T} \pi_{S,j}^2 \leq C_P(S)^2|T|, \quad \text{for any } T \subseteq \{1,\dots,n\}. \tag{84}$$

#### J.3.1 Expectation Term Yields the Dominant $k^2$ Term

We write $z$ as an explicit sum over nodes:

$$\begin{aligned} z &= \sum_j \pi_{S,j} G \alpha_j^{(0)} \\ &= G\left(\sum_{j\in I} \pi_{S,j}\mu_I\right) + G\left(\sum_{j\in I} \pi_{S,j}\xi_j + \sum_{j\notin I} \pi_{S,j}\eta_j\right). \end{aligned} \tag{85}$$

Taking expectation gives

$$\mathbb{E}[z] = G\left(\sum_{j\in I} \pi_{S,j}\right)\mu_I = G\left(\pi_S(I)\mu_I\right). \tag{86}$$

Therefore, by direct expansion,

$$\begin{aligned} \|\mathbb{E}[z]\|_2^2 &= \|G\mu_I\|_2^2 \left(\sum_{j\in I} \pi_{S,j}\right)^2 \\ &= \|G\mu_I\|_2^2 \, \pi_S^2(I) \\ &\geq \|G\mu_I\|_2^2 \left(c_P(S,I)k\right)^2. \end{aligned} \tag{87}$$

To make the dependence on the effective arrival weights explicit, note that

$$\pi_S(I) = \sum_{j\in I} \pi_{S,j} \geq k\,\underline{\pi}_{S,I}, \qquad \underline{\pi}_{S,I} := \min_{j\in I} \pi_{S,j}. \tag{88}$$

Moreover,

$$\pi_{S,j} = \sum_{i\in S} w_i P_{ij} \geq \left(\min_{i\in S} w_i\right) \max_{i\in S} P_{ij}. \tag{89}$$

Thus,

$$\underline{\pi}_{S,I} \geq w_{\min} c_P^\star(S,I), \qquad w_{\min} := \min_{i \in S} w_i, \tag{90}$$

and consequently

$$\pi_S(I) \geq k\, w_{\min}\, c_P^\star(S,I). \tag{91}$$

This yields the $\Theta(k^2)$ leading term, corresponding to the coherent summation effect under linear pooling. In particular,

$$\|\mathbb{E}[z]\|_2^2 \geq \|G\mu_I\|_2^2 \left[k\, w_{\min}\, c_P^\star(S,I)\right]^2. \tag{92}$$

For the two standard linear pooling operators, this gives:

1. **Sum pooling.** For sum pooling, $w = \mathbf{1}_S$, and hence $w_{\min} = 1$. Therefore,

$$\pi_S(I) \geq k\, c_P^\star(S,I), \qquad \|\mathbb{E}[z]\|_2^2 \geq \|G\mu_I\|_2^2 \left[k\, c_P^\star(S,I)\right]^2. \tag{93}$$

2. **Mean pooling.** For mean pooling, $w_i = 1/|S|$, and hence $w_{\min} = 1/|S|$. Therefore,

$$\pi_S(I) \geq \frac{k}{|S|} c_P^\star(S,I), \qquad \|\mathbb{E}[z]\|_2^2 \geq \|G\mu_I\|_2^2 \left[\frac{k}{|S|} c_P^\star(S,I)\right]^2. \tag{94}$$

### J.3.2 Variance and Cross Terms Are at Most Linear

The fluctuations around the expectation are contributed by the zero-mean terms. Their covariance satisfies

$$\begin{aligned} \mathrm{Var}(z) &\preceq \|G\|^2 \left( \sum_{j \in I} \pi_{S,j}^2 \Sigma_I + \sum_{j \notin I} \pi_{S,j}^2 \Sigma_N \right) \\ &\leq \|G\|^2 C_P(S)^2 \left( |I| \Sigma_I + |I^{\mathrm{c}}| \Sigma_N \right). \end{aligned} \tag{95}$$

Consequently,

$$\mathbb{E}\, \|z - \mathbb{E}z\|_2^2 = O\big(\|G\|^2 \left(\sigma_I^2 k + \sigma_N^2 (s-k)\right)\big). \tag{96}$$

Thus, the variance grows only linearly with the number of nodes. This linear scaling is controlled by the quadratic-mass bound $C_P(S)^2$.

Moreover, cross inner-products between informative and non-informative contributions do not produce $k^2$ growth. They are either nearly orthogonal, or their overlap along the principal directions does not increase with $k$. Consequently, the cross terms scale at most as $O(k)$.

### J.3.3 Combined Growth Law for $\|z\|_2^2$

Combining the expectation term with the variance bound gives

$$\begin{aligned} \mathbb{E}\left[\|z\|_2^2\right] &= \|\mathbb{E}[z]\|_2^2 + \mathbb{E}\left[\|z - \mathbb{E}z\|_2^2\right] \\ &\geq \underbrace{\|G\mu_I\|_2^2 \left(w_{\min} c_P^\star(S,I)\right)^2}_{\text{constant}} k^2 + O(k) + O(s-k). \end{aligned} \tag{97}$$

From a geometric perspective, the $k^2$ term arises from pairwise cross terms that contribute positively. Informative nodes aligned in the same principal-subspace direction add coherently, producing $\binom{k}{2} = O(k^2)$ positive inner products and thereby pushing the squared norm to the quadratic scale. In contrast, non-aligned nodes lack this coherence and contribute at most linearly through their individual energy.

For mean pooling, both the leading term and the lower-order terms are scaled by $1/|S|^2$:

$$\mathbb{E}\left[\|z\|_2^2\right] \geq \left(\frac{\|G\mu_I\|\, c_P(S,I)}{|S|}\right)^2 k^2 + \frac{1}{|S|^2} O(k+s). \tag{98}$$

### J.3.4 Summary: From Quadratic Growth of the Squared Norm to Linear Growth of the Norm

From the preceding derivation, the leading term satisfies

$$\mathbb{E}\big[\|z\|_2^2\big] = \Theta(k^2).$$

Consequently, the growth of the embedding norm itself is linear in $k$ up to the pooling normalization:

1. **Sum pooling.** For sum pooling, we obtain
$$\mathbb{E}\left[\|z\|_2\right] = \Theta(k). \tag{99}$$
2. **Mean pooling.** For mean pooling with pool size $s$, the scaling becomes
$$\mathbb{E}\left[\|z\|_2\right] = \frac{\Theta(k)}{s}. \tag{100}$$

In other words, the embedding norm, rather than its square, grows linearly with the number of informative nodes $k$ under sum pooling. Under mean pooling, it retains the same linear dependence on $k$ up to the $1/s$ normalization factor.

## J.4 Max Pooling: $\log k$ Growth from Extreme-value Statistics

This subsection develops the logarithmic-growth regime under the tail and dependence conditions in Sec. J.4.2. The growth rate depends on the activation tail; a general light-tail or Gumbel-domain assumption alone does not fix the rate.

**Stacked layers and notation.** Let $\alpha^{(L)} \in \mathbb{R}^{n\times r}$ denote the rank-$r$ principal-subspace representation matrix at layer $L$, where the $i$-th row is $\alpha_i^{(L)}$. From the previous factorization, we have

$$\alpha^{(L)} \approx \left(P\alpha^{(0)}\right) G^\top, \qquad P = \prod_\ell \hat{A}^{(\ell)}, \qquad G = \prod_\ell \Gamma_\ell S_\ell \in \mathbb{R}^{r\times r}, \tag{101}$$

where $P$ is nonnegative and row-stochastic.

Let the pooling region be $S \subseteq \{1,\dots,n\}$. For each channel $c \in \{1,\dots,r\}$, define

$$X_{i,c} := e_c^\top \alpha_i^{(L)}, \qquad z_c := \max_{i\in S} X_{i,c}, \qquad z := (z_1,\dots,z_r)^\top. \tag{102}$$

### J.4.1 From the Operator to the Best Landing Point for Each Informative Node

Let the informative node set be $I$, with $|I| = k$. We write the initial coefficients as

$$\alpha_j^{(0)} = \mu_I + \xi_j, \qquad j \in I, \qquad \alpha_j^{(0)} = \eta_j, \qquad j \notin I, \qquad \mathbb{E}[\xi_j] = \mathbb{E}[\eta_j] = 0. \tag{103}$$

By linear superposition, for channel $c$ we have

$$X_{i,c} \approx \underbrace{(G\mu_I)_c \sum_{j\in I} P_{ij}}_{\text{aligned mean shift}} + \underbrace{\sum_{j\in I} P_{ij}(G\xi_j)_c + \sum_{j\notin I} P_{ij}(G\eta_j)_c}_{\text{noise terms}}. \tag{104}$$

Since each row of $P$ is stochastic, $\sum_j P_{ij} \le 1$. Therefore, the aligned mean shift for a single node $i$ is at most on the order of $|(G\mu_I)_c|$, which is a constant. In particular, it does not grow linearly with $k$, in contrast to the sum-pooling case.

To characterize the growth of the maximum, define the best landing strength of each informative node $j$ into the pooling region $S$ as

$$p_j^\star := \max_{i\in S} P_{ij}, \qquad i^\star(j) \in \arg\max_{i\in S} P_{ij}. \tag{105}$$

We further introduce the column-maximum coverage lower bound

$$c_P^\star(S,I) := \inf_{j\in I} p_j^\star = \inf_{j\in I} \max_{i\in S} P_{ij}. \tag{106}$$

This quantity is positive when $S$ provides stable $L$-step coverage of $I$, i.e., when $c_P^\star(S, I) > 0$.

For each $j \in I$, consider the random variable that lower-bounds the contribution from $j$ at its best landing point:

$$Y_{j,c} := (G\mu_I)_c p_j^\star + \varepsilon_{j,c}, \tag{107}$$

where $\varepsilon_{j,c}$ denotes the maximized noise contribution from node $j$ at its best receiver $i^\star(j)$. In this lower-bound analysis, we temporarily ignore terms from $j' \neq j$.

### J.4.2 Light-tailed Regime and Extremal Closure

**Tail condition for the logarithmic regime.** For the logarithmic claims below, assume that the effective channel responses have a non-degenerate exponential-type right tail, with scale bounded above and away from zero as the number of candidates grows, and sufficient independence or weak dependence for the extreme-value approximation. More explicitly, the upper-tail and informative-candidate lower-tail bounds have matching exponential order, and the effective candidate count is proportional to $k$. The i.i.d. Gumbel family below gives an exact illustrative case. Membership in the Gumbel maximum domain of attraction alone is insufficient: it also includes Gaussian tails, whose maxima have a different growth rate.

For example, with independent effective responses and regular right tails satisfying $-\log(1-F(t)) \sim ct^\alpha$ for $c, \alpha > 0$, the high quantile $b_k = F^{\leftarrow}(1-1/k)$ has order $(\log k)^{1/\alpha}$. Under the corresponding moment conditions, this is also the order of the expected maximum. Exponential-type tails ($\alpha = 1$) give logarithmic growth, whereas Gaussian-type tails ($\alpha = 2$) give square-root-logarithmic growth. A sub-Gaussian upper bound alone supplies an upper growth bound, not a matching lower bound. The norm objective itself does not prescribe either asymptotic rate, and the finite-range empirical fits do not uniquely identify the activation-tail regime.

For illustration, in the idealized i.i.d. Gumbel case, exact closure holds.

1. **Gumbel closure under the i.i.d. case.** If

   $$Y_{j,c} \sim \mathrm{Gumbel}(\mu_c, \beta_c)$$

   are i.i.d. across $j$, then

   $$M_{k,c} := \max_{j=1,\dots,k} Y_{j,c} \sim \mathrm{Gumbel}\left(\mu_c^{(k)}, \beta_c\right), \tag{108}$$

   with

   $$\mu_c^{(k)} = \underbrace{\mu_c^{\varepsilon} + (G\mu_I)_c c_P^\star(S, I)}_{:=\mu_c} + \beta_c \log k. \tag{109}$$

   This follows from the Gumbel cumulative distribution function

   $$\Lambda_{\mu,\beta}(x) = \exp\left(-e^{-(x-\mu)/\beta}\right),$$

   since

   $$\begin{aligned} \Pr(M_{k,c} \le x) &= \Lambda_{\mu,\beta}(x)^k \\ &= \exp\left(-e^{-(x-(\mu+\beta\log k))/\beta}\right) \\ &= \Lambda_{\mu+\beta\log k,\beta}(x). \end{aligned} \tag{110}$$

2. **Expectation and high quantiles.** Consequently,

   $$\mathbb{E}[M_{k,c}] = \mu_c + \beta_c \log k + \gamma\beta_c, \tag{111}$$

   and

   $$Q_{1-\delta}(M_{k,c}) = \mu_c + \beta_c \left(\log k - \log\log(1/\delta)\right), \tag{112}$$

   where $\gamma$ is the Euler constant and $Q_{1-\delta}$ denotes the $(1-\delta)$-quantile. Both quantities exhibit logarithmic growth in the sample size $k$ in this fixed-scale Gumbel model.

Since $z_c \ge M_{k,c}$, we immediately obtain the lower bound

$$\mathbb{E}[z_c] \ge \mu_c + \beta_c \log k + \gamma\beta_c, \qquad \text{and hence} \qquad \mathbb{E}[z_c] = \Omega(\log k). \tag{113}$$

### J.4.3 Upper Bound: Row-stochasticity and Light Tails Imply $\mathcal{O}(\log k)$ Growth

We now return to the full expression for $X_{i,c}$. Under the uniform exponential-moment bounds on the source terms, the nonnegative row weights with $\sum_j P_{ij} \le 1$ preserve an exponential upper-tail bound (by convexity of the exponential and a Chernoff bound). Thus, there exist constants $C$, $\tilde{\mu}_c$, and $\tilde{\beta}_c$, independent of $k$, such that, for sufficiently large $t$,

$$\Pr\{X_{i,c} > t\} \le C \exp\left(-\frac{t - \tilde{\mu}_c}{\tilde{\beta}_c}\right). \tag{114}$$

Applying a union bound, or a standard extreme-value inequality, over the pooling set $S$ yields

$$\Pr\{z_c > t\} \le |S|\, C \exp\left(-\frac{t - \tilde{\mu}_c}{\tilde{\beta}_c}\right). \tag{115}$$

Consequently, the $(1-\delta)$-quantile satisfies

$$Q_{1-\delta}(z_c) \le \tilde{\mu}_c + \tilde{\beta}_c \left(\log |S| - \log \delta\right), \tag{116}$$

where $Q_{1-\delta}(\cdot)$ denotes the $(1-\delta)$-quantile.

Under the covering and overlap-control conditions, the number of effectively independent candidates activated by informative nodes is at most a constant multiple of $k$. Equivalently,

$$k_{\text{eff}}(S, I; \tau) = \Theta(k).$$

Therefore,

$$\mathbb{E}[z_c] = \mathcal{O}\left(\log k_{\text{eff}}(S, I; \tau)\right) = \mathcal{O}(\log k). \tag{117}$$

Combining this upper bound with the lower-bound analysis in Section J.4.2 gives the matching-order result

$$\mathbb{E}[z_c] = \Theta(\log k). \tag{118}$$

### J.4.4 Conclusion for the Vector Norm

Treating each channel as a one-dimensional extremum, and noting that the per-channel extreme-value scales $\beta_c$ remain bounded with controlled inter-channel correlations, we have

$$\begin{aligned} \mathbb{E}\left[\|z\|_2^2\right] &= \sum_{c=1}^{r} \mathbb{E}\left[z_c^2\right] \\ &\approx \sum_{c=1}^{r} \left(\mu_c + \beta_c \log k\right)^2 \\ &= \Theta\big((\log k)^2\big). \end{aligned} \tag{119}$$

Consequently,

$$\mathbb{E}\left[\|z\|_2\right] = \Theta(\log k), \qquad \mathbb{E}\left[\|z\|_2^2\right] = \Theta\big((\log k)^2\big). \tag{120}$$

The constants hidden in the $\Theta(\cdot)$ notation depend on the scale parameters $\{\beta_c\}$, the location parameters $\{\mu_c\}$, which incorporate the effects of $G\mu_I$ and $c_P^\star(S, I)$, and the strength of inter-channel correlations.

### J.4.5 Extreme-value Classification and the Light-tailed Regime

Let $X_1, \ldots, X_k$ be independent and identically distributed random variables with common distribution function $F$ and density $f$, when it exists. Define the window maximum as

$$M_k := \max_{1 \le i \le k} X_i, \qquad \Pr(M_k \le x) = F(x)^k. \tag{121}$$

The Fisher–Tippett–Gnedenko extreme-value theorem (Gnedenko, 1993) states that, if there exist normalization constants $b_k \in \mathbb{R}$ and $a_k > 0$ such that

$$\frac{M_k - b_k}{a_k} \xrightarrow{d} G, \tag{122}$$

then the limiting law $G$ must belong to one of three types: Gumbel, Fréchet, or Weibull. Many regular distributions with light right tails and no finite right endpoint belong to the Gumbel maximum domain of attraction; the von Mises condition below gives a sufficient criterion. The standard Gumbel distribution satisfies

$$\Pr(G \le x) = \exp\bigl(-e^{-x}\bigr)\,. \tag{123}$$

Common families in the Gumbel domain include Gaussian, log-normal, exponential, gamma, logistic, Laplace, and Gumbel distributions. Their centering and scaling sequences need not have the same orders; in particular, log-normal distributions need not have finite positive exponential moments.

A standard normalization choice is given by a high quantile together with the von Mises scale:

$$b_k := F^{\leftarrow}\left(1 - \frac{1}{k}\right), \qquad \text{so that} \qquad F(b_k) = 1 - \frac{1}{k}, \tag{124}$$

and

$$a_k := a(b_k), \qquad a(x) := \frac{1 - F(x)}{f(x)} = \frac{1}{h(x)}, \tag{125}$$

where

$$h(x) := \frac{f(x)}{1 - F(x)}$$

is the hazard rate.

Let

$$x_F := \sup\{x : F(x) < 1\}$$

be the right endpoint, often with $x_F = \infty$. If $F$ is differentiable in the right tail and $f > 0$ for large $x$, a useful sufficient von Mises condition for the Gumbel domain is

$$\lim_{x \to x_F} a'(x) = 0, \qquad \text{equivalently} \qquad \lim_{x \to x_F} \frac{h'(x)}{h(x)^2} = 0. \tag{126}$$

Under this condition, $F$ belongs to the Gumbel maximum domain of attraction.

Consequently,

$$\Pr\left(\frac{M_k - b_k}{a_k} \le x\right) = F(b_k + a_k x)^k \longrightarrow \exp\bigl(-e^{-x}\bigr)\,. \tag{127}$$

Intuitively, near $b_k$, the tail admits the first-order approximation

$$1 - F(b_k + a_k x) \approx \frac{1}{k} e^{-x}, \tag{128}$$

and therefore

$$F(b_k + a_k x)^k \approx \left(1 - \frac{e^{-x}}{k}\right)^k \to e^{-e^{-x}}, \tag{129}$$

which yields the Gumbel limit.

Thus, for a pooling window with effective size $k$, if the approximately i.i.d. activations satisfy the stated Gumbel-domain conditions, then

$$Z_{\max} := \max_{1 \le i \le k} X_i \approx b_k + a_k G, \qquad G \sim \text{Gumbel}. \tag{130}$$

With additional moment control ensuring convergence of the first two moments, this gives the first-order moment approximations

$$\mathbb{E}[Z_{\max}] \approx b_k + \gamma a_k, \qquad \operatorname{Var}(Z_{\max}) \approx \frac{\pi^2}{6} a_k^2, \tag{131}$$

where $\gamma \approx 0.5772$ is the Euler–Mascheroni constant.

**Scope of the tail model.** The preceding calculation is conditional on the specified effective-response distribution. Finite positive exponential moments rule out power-law right tails, but neural architectures alone do not establish a particular tail regime. In particular, hard clipping, explicit norm constraints, or fixed-parameter normalization can bound the relevant responses and invalidate an unbounded-tail asymptotic model. Such cases require a separate analysis. We therefore use the exponential-type model as a sufficient-condition characterization of logarithmic growth, not as a universal consequence of message passing or ReLU. The empirical norm-growth curves support the usefulness of the signal over the observed range, without establishing a unique tail law.

### J.5 Why Non-informative Features Have a Significantly Weaker Effect

**Notation recap.** After stacking layers, we have

$$\alpha^{(L)} \approx \left(P\alpha^{(0)}\right) G^\top.$$

On the pooling region $S$, the pooled representation takes the following forms:

1. **Linear pooling.** For linear pooling,

$$z = \left(w^\top P\alpha^{(0)}\right) G^\top = \left(\pi_S^\top \alpha^{(0)}\right) G^\top, \qquad \pi_S^\top := w^\top P. \tag{132}$$

2. **Max pooling.** For channel-wise max pooling,

$$z_c = \max_{i\in S} X_{i,c}, \qquad X_{i,c} := e_c^\top \alpha_i^{(L)}. \tag{133}$$

We decompose the initial coefficients into informative and non-informative parts, supported on $I$ and $I^{\mathrm{c}}$, respectively:

$$\alpha_j^{(0)} = \begin{cases} \mu_I + \xi_j, & j \in I, \qquad \mathbb{E}[\xi_j] = 0, \quad \mathrm{Cov}(\xi_j) \preceq \Sigma_I, \\ \eta_j, & j \in I^{\mathrm{c}}, \qquad \mathbb{E}[\eta_j] = 0, \quad \mathrm{Cov}(\eta_j) \preceq \Sigma_N. \end{cases} \tag{134}$$

#### J.5.1 Sum/Mean Pooling: Non-informative Contributions Are Variance-level, whereas Informative Contributions Are Quadratic-level

From

$$z = \left(\pi_S^\top \alpha^{(0)}\right) G^\top,$$

we obtain

$$\begin{aligned} \mathbb{E}[z] &= G\left(\pi_S(I)\mu_I\right), \\ \mathrm{Var}(z) &\preceq \|G\|^2 \left(\sum_{j\in I} \pi_{S,j}^2 \Sigma_I + \sum_{j\in I^{\mathrm{c}}} \pi_{S,j}^2 \Sigma_N\right), \end{aligned} \tag{135}$$

where

$$\pi_S(I) := \sum_{j\in I} \pi_{S,j}.$$

Using the uniform bounds

$$\pi_S(I) \geq k\|w\|_1 c_P(S,I), \qquad \sum_{j\in T} \pi_{S,j}^2 \leq C_P(S)^2 |T|, \tag{136}$$

we obtain the dominant contribution from informative nodes:

$$\begin{aligned} \|\mathbb{E}[z]\|_2^2 &= \|G\mu_I\|_2^2 \left[\pi_S(I)\right]^2 \\ &\gtrsim \|G\mu_I\|_2^2 \left[k\|w\|_1 c_P(S,I)\right]^2 \\ &= \Theta(k^2). \end{aligned} \tag{137}$$

In contrast, non-informative nodes appear only through the variance term:

$$\begin{aligned} \mathbb{E}\left\|z - \mathbb{E}[z]\right\|_2^2 &\lesssim \|G\|^2 C_P(S)^2 \left(|I|\,\|\Sigma_I\|_2 + |I^{\mathrm{c}}|\,\|\Sigma_N\|_2\right) \\ &= O(k) + O(s-k). \end{aligned} \tag{138}$$

**Key point.** Non-informative components have zero mean and incoherent directions. Hence, they do not add coherently and can only accumulate at linear, variance-level order under sum or mean pooling. In contrast, informative components are aligned with the same principal directions, producing $\binom{k}{2} = O(k^2)$ positive pairwise cross terms in the squared norm of the sum. This coherent accumulation yields the quadratic leading term.

### J.5.2 Max Pooling: Non-informative Extremes Grow Only Logarithmically and Have Weaker Location and Scale

For channel $c$, we have

$$X_{i,c} \approx (G\mu_I)_c \sum_{j\in I} P_{ij} + \sum_{j\in I} P_{ij}(G\xi_j)_c + \sum_{j\in I^c} P_{ij}(G\eta_j)_c. \tag{139}$$

**Informative part.** Recall the definitions

$$p_j^\star := \max_{i\in S} P_{ij}, \qquad c_P^\star(S,I) := \inf_{j\in I} p_j^\star. \tag{140}$$

Under the exponential-type tail and effective-candidate conditions of Sec. J.4.2, the maximum over informative nodes exhibits logarithmic growth. In particular,

$$\max_{j\in I} \left((G\mu_I)_c p_j^\star + \varepsilon_{j,c}\right) \tag{141}$$

admits the asymptotic expansion

$$\mathbb{E}\left[\max_{j\in I} \left((G\mu_I)_c p_j^\star + \varepsilon_{j,c}\right)\right] = \mu_c + \beta_c \log k_{\text{eff}}(S,I;\tau) + O(\beta_c), \tag{142}$$

where

$$\mu_c = \mu_c^\varepsilon + (G\mu_I)_c c_P^\star(S,I).$$

Hence,

$$\mathbb{E}[z_c] = \Theta(\log k_{\text{eff}}(S,I;\tau)),$$

and there is a positive location shift

$$(G\mu_I)_c c_P^\star(S,I).$$

**Non-informative part.** Extremes from non-informative nodes are produced purely by noise. The row-stochasticity of $P$ restricts any deterministic shift to constant order. Under the same exponential-type tail regime, the extreme-value approximation gives

$$\mathbb{E}\left[\max_{j\in I^c} \varepsilon_{j,c}^{(N)}\right] = \mu_c^{(N)} + \beta_c^{(N)} \log(s - k_{\text{eff}}) + O\Big(\beta_c^{(N)}\Big), \tag{143}$$

where the location term $\mu_c^{(N)}$ does not contain the informative bias

$$(G\mu_I)_c c_P^\star(S,I).$$

**Combination.** Since

$$z_c = \max\{\text{informative extreme}, \text{non-informative extreme}\}, \tag{144}$$

and the informative extreme contains the additional positive location shift

$$(G\mu_I)_c c_P^\star(S,I),$$

whereas the non-informative extreme does not, the informative contribution dominates for sufficiently large $k$. Aggregating across channels yields

$$\mathbb{E}\left[\|z\|_2\right] = \Theta\left(\log k_{\text{eff}}(S,I;\tau)\right). \tag{145}$$

The non-informative contribution is at most

$$O(\log(s - k_{\text{eff}})),$$

with a smaller multiplicative scale.

# K LayerNorm under a Controllable Normalization Regime

**Setup and notation.** We analyze LayerNorm under a controllable normalization regime, focusing on its effect on the principal-subspace dynamics.

The intermediate pre-normalization quantity at layer $\ell$ is

$$\tilde{h}_i^{(\ell+1)} = \sum_j \hat{A}_{ij}^{(\ell)} W_\ell h_j^{(\ell)}, \tag{146}$$

which is then passed through LayerNorm, with learnable parameters $\gamma_\ell$ and $\beta_\ell$, followed by an activation.

Stack the principal-subspace coordinates row-wise into $\alpha^{(\ell)} \in \mathbb{R}^{n\times r}$. We retain the factorization derived above:

$$\alpha^{(L)} \approx \left(P\alpha^{(0)}\right) G^\top, \qquad P = \prod_\ell \hat{A}^{(\ell)}, \qquad G = \prod_\ell \Gamma_\ell S_\ell. \tag{147}$$

In what follows, we characterize sufficient regularity conditions under which LayerNorm preserves controllable scaling behavior in this factorized form.

### K.1 Equivalent Scaling of LayerNorm on the Principal Subspace

For each node $i$ and layer $\ell$, the Jacobian of LayerNorm admits a local linear approximation consisting of centering and channel-wise scaling. Projecting onto the output principal subspace gives

$$U_\ell^\top \mathrm{LN}_\ell\left(\tilde{h}_i^{(\ell+1)}\right) \approx \underbrace{\Gamma_\ell(i)}_{\text{node-wise channel-wise nonnegative scaling}} U_\ell^\top \tilde{h}_i^{(\ell+1)} + \epsilon_{\ell,i}^{\mathrm{LN}}, \tag{148}$$

where $\Gamma_\ell(i) \in \mathbb{R}^{r\times r}$ can be taken as a diagonal matrix with positive entries, after fixing the column signs consistently with $U_\ell$. The term $\epsilon_{\ell,i}^{\mathrm{LN}}$ collects centering, bias, and higher-order perturbation terms.

**Controllability conditions.** We characterize a stable normalization regime through the following regularity conditions:

1. **(LN-A) Scaling stability.** For the pooling region $S$, or uniformly over nodes, there exists a channel vector $g_\ell \in \mathbb{R}^r_{>0}$ such that
$$\Gamma_\ell(i) = \mathrm{diag}(g_\ell) + \Delta_\ell(i), \qquad \|\Delta_\ell(i)\| \le \varepsilon_\ell^{\mathrm{LN}}. \tag{149}$$
2. **(LN-B) Sign consistency.** By fixing the column signs of $U_\ell$, or through a positively homogeneous activation such as ReLU, the effective principal-channel scalings can be taken to be nonnegative.
3. **(LN-C) Bias absorbability.** Contributions from $\beta_\ell$ and centering residuals enter only through lower-order terms and are absorbed into the aggregated perturbation term.

Under these conditions, node-wise variability can be averaged or upper-bounded. Hence, we may write

$$\Gamma_\ell := \mathrm{diag}(g_\ell), \qquad \epsilon_{\ell,i}^{\mathrm{LN}} \text{ is absorbed into } \eta_\ell. \tag{150}$$

The matrix $\Gamma_\ell$ is then incorporated into the total gain

$$G = \prod_\ell \Gamma_\ell S_\ell.$$

### K.2 Impact of LayerNorm on the Conclusions of Section I

Within the controllable normalization regime described above, LayerNorm induces an approximately global per-channel scaling that can be absorbed into the total gain $G$. The resulting effects on the conclusions of Section I are summarized as follows:

1. **Sum and mean pooling.** The principal term remains

$$\|\mathbb{E}z\|_2^2 = \|G\mu_I\|_2^2 \left[\pi_S(I)\right]^2. \tag{151}$$

LayerNorm modifies only the constant factor $\|G\mu_I\|$ through the per-layer scaling matrices $\Gamma_\ell$. The coverage term $\pi_S(I)$ and the coverage constant $c_P(S, I)$ remain unchanged. Hence, the $\Theta(k^2)$ growth law is preserved. The variance upper bound, which depends on $\|G\|$, is likewise affected only up to constant factors.

2. **Max pooling and Gumbel-type extrema.** For each channel, the location and scale parameters become

$$\mu_c = \mu_c^{\varepsilon} + (G\mu_I)_c c_P^{\star}(S, I), \qquad \beta_c \mapsto \tilde{\beta}_c,$$

where $\tilde{\beta}_c$ is scaled proportionally by the product of the per-layer normalization factors $\Gamma_\ell$. The $\log k$ growth of extrema arises from the increasing effective sample size and is invariant under constant scalings. Therefore,

$$\mathbb{E}\left[\|z\|_2\right] = \Theta(\log k) \tag{152}$$

remains unchanged.

3. **Relative strength of informative and non-informative features.** The contrast between informative and non-informative features is preserved. Informative features retain their coherent location-and-scale advantage, whereas non-informative features remain incoherent and weak-scale. LayerNorm introduces only multiplicative constants and does not alter the ordering between informative and non-informative components.

### K.3 Aggregation and Upper Bounds of Error Terms

We combine the perturbations introduced by LayerNorm with the alignment and leakage effects into the per-layer error term:

$$\begin{aligned}\varepsilon_i^{(\ell)} := \underbrace{\tau_\ell \left\|\Pi_{U_{\ell-1}} h_i^{(\ell)}\right\| + \delta_\ell \left\|\Pi_{U_{\ell-1}}^{\perp} h_i^{(\ell)}\right\|}_{\text{alignment and leakage}} \\ + \underbrace{\varepsilon_\ell^{\mathrm{LN}} + \left\|\epsilon_{\ell,i}^{\mathrm{LN}}\right\|}_{\text{LayerNorm fluctuations}} + \eta_\ell.\end{aligned} \tag{153}$$

After stacking and pooling, these perturbations accumulate according to the error-propagation inequality established above. Since $\Gamma_\ell$ is nonnegative and $P$ is row-stochastic, the aggregated error terms contribute only through additive lower-order terms. Therefore, they do not affect the leading-order scaling behavior derived in Section J.

## L Effect of ReLU on Growth Laws

**Conclusion.** Taking the node propagation operator $P$ and the channel gain

$$G = \prod_\ell \Gamma_\ell S_\ell$$

as given, and letting the activation be $\sigma = \mathrm{ReLU}$, the leading growth laws remain unchanged.

For sum and mean pooling, the leading term remains

$$\|\mathbb{E}[z]\|_2^2 = \|G\mu_I\|_2^2 \left[\pi_S(I)\right]^2 = \Theta(k^2). \tag{154}$$

For max pooling, the leading term remains, for each channel $c$,

$$\mathbb{E}[z_c] = \Theta(\log k), \tag{155}$$

and therefore

$$\mathbb{E}\left[\|z\|_2\right] = \Theta(\log k). \tag{156}$$

ReLU preserves the right tail and acts as the identity on large positive values. Specifically, it truncates the negative half-line, acts as the identity on the positive half-line, and is positively homogeneous on the positive ray:

$$\mathrm{ReLU}(ax) = a\,\mathrm{ReLU}(x), \qquad a \geq 0.$$

Consequently, the deterministic signal amplification that grows with $k$ is preserved after ReLU: linear pooling retains its $\Theta(k)$ norm growth, and max pooling retains its $\Theta(\log k)$ extremal growth. Thus, ReLU does not disrupt the leading-order growth laws, and in practice may further reinforce them by suppressing negative or weakly aligned components.

### L.1 Three Key Properties

The ReLU activation satisfies the following properties:

1. **Monotonicity.** If $x \leq y$, then

   $$\mathrm{ReLU}(x) \leq \mathrm{ReLU}(y).$$

   Therefore, ReLU is order-preserving, and comparisons or lower bounds between maxima and means remain applicable after activation.

2. **1-Lipschitz continuity.** For any $x, y \in \mathbb{R}$,

   $$|\mathrm{ReLU}(x) - \mathrm{ReLU}(y)| \leq |x - y|.$$

   Hence, the error upper bounds from the previous analysis carry over directly through ReLU.

3. **Positive homogeneity.** For any $a \geq 0$,

   $$\mathrm{ReLU}(ax) = a\,\mathrm{ReLU}(x).$$

   Thus, ReLU does not change the growth rate of signals that scale with $k$; it only modifies multiplicative constants.

### L.2 Linear Pooling: Sum/Mean Pooling Preserves the $\Theta(k^2)$ Leading Term

Recall the decoupling formula for linear pooling, omitting lower-order error terms:

$$z \approx \left(\pi_S^\top \alpha^{(0)}\right) G^\top, \qquad \pi_S^\top := w^\top P. \tag{157}$$

Write the initial coordinates as

$$\alpha_j^{(0)} = \begin{cases} \mu_I + \xi_j, & j \in I, \qquad \mathbb{E}[\xi_j] = 0, \\ \eta_j, & j \notin I, \qquad \mathbb{E}[\eta_j] = 0, \end{cases} \tag{158}$$

and work in the light-tailed regime, such as sub-Gaussian or sub-exponential noise. For each major channel $c$, the pooled pre-activation, consisting of the aligned signal and noise, can be written as

$$Y_c = (G\mu_I)_c \underbrace{\left(\sum_{j \in I} \pi_{S,j}\right)}_{\pi_S(I)} + \zeta_c, \tag{159}$$

where $\mathbb{E}[\zeta_c] = 0$ and

$$\mathrm{Var}[\zeta_c] = O\left(\sum_j \pi_{S,j}^2\right) = O(k). \tag{160}$$

Hence, define

$$m_k := \mathbb{E}[Y_c] = (G\mu_I)_c \pi_S(I) = \Theta(k), \qquad \sigma_k := \mathrm{std}(Y_c) = O(\sqrt{k}). \tag{161}$$

The signal-to-noise ratio grows with $k$:

$$\frac{m_k}{\sigma_k} = \Theta(\sqrt{k}) \longrightarrow \infty. \tag{162}$$

After ReLU, we have

$$\mathbb{E}\left[\mathrm{ReLU}(Y_c)\right] = \mathbb{E}[Y_c] - \mathbb{E}\left[(-Y_c)^+\right]. \tag{163}$$

Using standard tail bounds for sub-Gaussian or sub-exponential noise, there exists a constant $c > 0$ such that

$$\Pr\{Y_c \le 0\} \le \exp\left(-c\frac{m_k^2}{\sigma_k^2}\right) = \exp(-c'k). \tag{164}$$

By Cauchy–Schwarz, or similar moment-probability inequalities,

$$\mathbb{E}\left[(-Y_c)^+\right] \le \sigma_k \Pr\{Y_c \le 0\}^{1/2} = O\Big(\sqrt{k}\, e^{-c''k}\Big) = o(m_k). \tag{165}$$

Therefore,

$$\mathbb{E}\left[\mathrm{ReLU}(Y_c)\right] = m_k(1 - o(1)) = \Theta(k). \tag{166}$$

Reassembling the channels into the vector $z$ and squaring the norm, the leading term remains

$$\begin{aligned} \|\mathbb{E}[z]\|_2^2 &= \|G\mu_I\|_2^2 \left[\pi_S(I)\right]^2 \\ &\ge \|G\mu_I\|_2^2 \left[c_P(S,I)k\right]^2 \\ &= \Theta(k^2). \end{aligned} \tag{167}$$

Thus, ReLU only truncates a vanishing fraction of negative noise and does not weaken the leading term.

### L.3 Max Pooling: ReLU Preserves the $\Theta(\log k)$ Extremal Growth

For any channel $c$ and pooling region $S$, since ReLU is monotone increasing, we have

$$\begin{aligned} z_c^{(\mathrm{ReLU})} &= \max_{i \in S} \mathrm{ReLU}(X_{i,c}) \\ &= \mathrm{ReLU}\left(\max_{i \in S} X_{i,c}\right) \\ &= \mathrm{ReLU}(M_{k,c}). \end{aligned} \tag{168}$$

Extreme-value theory gives the expansion

$$M_{k,c} = b_k + a_k \tilde{G}_c + o_{\mathbb{P}}(1), \tag{169}$$

where $\tilde{G}_c$ has a non-degenerate limiting distribution in the Gumbel domain, and

$$b_k \sim \mu_c + \beta_c \log k \to +\infty.$$

Consequently,

$$\Pr\{M_{k,c} \le 0\} \to 0, \qquad \mathrm{ReLU}(M_{k,c}) = M_{k,c} \quad \text{with probability tending to 1.} \tag{170}$$

Therefore,

$$\begin{aligned} \mathbb{E}\left[z_c^{(\mathrm{ReLU})}\right] &= \mathbb{E}[M_{k,c}](1 - o(1)) \\ &= \mu_c + \beta_c \log k + O(1) \\ &= \Theta(\log k). \end{aligned} \tag{171}$$

The vector norm inherits the same leading orders:

$$\mathbb{E}\left[\|z\|_2\right] = \Theta(\log k), \qquad \mathbb{E}\left[\|z\|_2^2\right] = \Theta\big((\log k)^2\big). \tag{172}$$

**Explanation.** The growth of the maximum comes from the right tail and the increase in the effective sample size $k$. ReLU acts as the identity on the positive half-line, and for large $k$, the maximum is positive with probability tending to one. Hence, ReLU does not alter the leading-order behavior.

### L.4 Interface with $G$ and $P$: ReLU Changes Only Constants, Not Orders

The main effect of ReLU is absorbed into the nonnegative channel scaling

$$\Gamma_\ell = \mathrm{diag}(g_\ell)$$

at each layer, as described in the equivalent scaling analysis for normalization and activation. As a result, ReLU modifies the constant factor in the total gain

$$G = \prod_\ell \Gamma_\ell S_\ell,$$

while leaving the node propagation operator $P$, the coverage constants $c_P(S, I)$ and $c_P^\star(S, I)$, and the effective arrival distribution $\pi_S$ unchanged.

Therefore, the leading-order growth laws are preserved:

1. **Sum and mean pooling.** The growth law
$$\Theta(k^2)$$
for the squared norm under sum and mean pooling remains unchanged under ReLU.
2. **Max pooling.** The growth law
$$\Theta(\log k)$$
for the norm under max pooling also remains unchanged under ReLU.

Thus, ReLU may change multiplicative constants through the channel gain $G$, but it does not alter the leading-order behavior.

### L.5 Summary

The effect of ReLU on the preceding growth laws can be summarized as follows:

1. **Linear pooling.** For sum or mean pooling,
$$\mathbb{E}\left[\|z\|_2^2\right] = \|G\mu_I\|_2^2 \left[\pi_S(I)\right]^2 + O(k), \qquad \pi_S(I) \geq c_P(S, I)k. \tag{173}$$
ReLU truncates only negative noise. Since the aligned signal dominates the noise as $k$ grows, we have
$$\mathbb{E}\left[\mathrm{ReLU}(Y_c)\right] = \Theta(k),$$
and the leading $\Theta(k^2)$ term in the squared norm remains unchanged.
2. **Max pooling.** Since ReLU is monotone,
$$\max_i \mathrm{ReLU}(X_i) = \mathrm{ReLU}\left(\max_i X_i\right).$$
In the exponential-type tail regime of Sec. J.4.2, the maximum $M_{k,c}$ diverges to $+\infty$ with probability tending to one. Hence,
$$\mathbb{E}[z_c] = \Theta(\log k)$$
remains unchanged after ReLU.
3. **Errors and constants.** ReLU is 1-Lipschitz, so the error upper bounds remain consistent with the previous analysis. Its positive homogeneity and monotonicity ensure that the growth laws are not disrupted. ReLU only modifies multiplicative constants through the effective channel scaling $\Gamma_\ell$.

## M GAT and GIN: Alignment-enhanced Activations

All logarithmic max-pooling statements in this section inherit the exponential-type tail conditions of Sec. J.4.2; they are conditional extensions, not general guarantees for every activation distribution.

**Review of unified notation.** After the $L$-th layer, the coordinates on the principal subspace satisfy

$$\alpha^{(L)} \approx \left(P\alpha^{(0)}\right) G^\top, \qquad P = \prod_\ell \hat{A}^{(\ell)}, \qquad G = \prod_\ell \Gamma_\ell S_\ell. \tag{174}$$

Here $P$ is nonnegative and row-stochastic, while $G$ denotes the accumulated channel-wise gain.

For linear pooling over the pooling region $S$, the graph embedding is written as

$$z = \left(w^\top P\alpha^{(0)}\right) G^\top. \tag{175}$$

The corresponding effective reach distribution is

$$\pi_S^\top := w^\top P, \tag{176}$$

and the total coverage of the informative node set $I$ is

$$\pi_S(I) := \sum_{j\in I} \pi_{S,j}. \tag{177}$$

### M.1 From Static to Dynamic Propagation Kernels

The preceding analysis is built on an operator view of message passing. After projecting node representations onto the principal subspace of the learned transformations, the $L$-layer propagation can be written as

$$\alpha^{(L)} \approx \left(P\alpha^{(0)}\right) G^\top, \qquad P = \prod_{\ell=1}^{L} \hat{A}^{(\ell)}, \qquad G = \prod_{\ell=1}^{L} \Gamma_\ell S_\ell. \tag{178}$$

Here $P$ describes node-wise propagation, while $G$ describes channel-wise amplification along model-utilized directions. This representation separates two effects: how signals are transported across the graph, and how aligned features are amplified by the learned transformations.

For GCN-type models, this separation is particularly direct because the propagation matrix $\hat{A}$ is determined by the graph topology and degree normalization. Once the input graph is given, the node-wise propagation operator is therefore a static convolution kernel independent of the hidden representations.

GIN also largely follows a static aggregation pattern at the level of neighborhood mixing. Its aggregation step sums the self representation and neighboring representations with fixed coefficients:

$$h_i^{(\ell)} = \mathrm{MLP}^{(\ell)}\left((1+\epsilon_\ell)h_i^{(\ell-1)} + \sum_{j\in\mathcal{N}(i)} h_j^{(\ell-1)}\right). \tag{179}$$

The subsequent MLP is nonlinear and can amplify aligned directions, but the neighborhood aggregation itself does not assign input-adaptive weights to different neighbors. Thus, for the purpose of analyzing signal accumulation, GIN can still be understood as applying a static sum-type propagation followed by a learnable feature transformation. Its main difference from GCN is not that the propagation kernel is dynamically recomputed from the current features, but that the unnormalized summation preserves and accumulates multiset-level additive signals more strongly.

GAT is different in a fundamental way. A GAT layer computes

$$h_i^{(\ell)} = \sum_{j\in\mathcal{N}(i)} \alpha_{ij}^{(\ell)} W^{(\ell)} h_j^{(\ell-1)}, \qquad \alpha_{ij}^{(\ell)} = \frac{\exp(e_{ij}^{(\ell)})}{\sum_{m\in\mathcal{N}(i)} \exp(e_{im}^{(\ell)})}, \tag{180}$$

where

$$e_{ij}^{(\ell)} = \mathrm{LeakyReLU}\left(a_\ell^\top \left[W^{(\ell)}h_i^{(\ell-1)} \,\|\, W^{(\ell)}h_j^{(\ell-1)}\right]\right). \tag{181}$$

Therefore, the attention matrix

$$\hat{A}_{\mathrm{GAT}}^{(\ell)}\left(H^{(\ell-1)}\right) := \left[\alpha_{ij}^{(\ell)}\right]_{i,j} \tag{182}$$

is not a fixed graph-dependent matrix, but an input-dependent propagation kernel determined by the current hidden representations. In this sense, GAT implements a dynamic convolution: the convolution weights are recomputed at each layer and for each input graph according to the node features being propagated.

This distinction is important for extending the norm-growth analysis. For GCN and GIN, the effective propagation operator can be treated as a fixed, or externally specified, operator for a given graph. For GAT, however, the corresponding operator is

$$P_{\mathrm{GAT}}(H) = \hat{A}^{(L)}_{\mathrm{GAT}}\left(H^{(L-1)}\right)\hat{A}^{(L-1)}_{\mathrm{GAT}}\left(H^{(L-2)}\right)\cdots\hat{A}^{(1)}_{\mathrm{GAT}}\left(H^{(0)}\right), \tag{183}$$

which is itself a random, representation-dependent object. Consequently, the quantities used in the previous operator analysis, such as the effective reach distribution

$$\pi_S^\top = w^\top P \tag{184}$$

and the informative coverage

$$\pi_S(I) = \sum_{j\in I}\pi_{S,j}, \tag{185}$$

should be interpreted as conditional quantities given the dynamically generated attention kernels:

$$\pi^\top_{S,\mathrm{GAT}}(H) = w^\top P_{\mathrm{GAT}}(H), \qquad \pi_{S,\mathrm{GAT}}(I;H) = \sum_{j\in I}\pi_{S,\mathrm{GAT},j}(H). \tag{186}$$

Despite this additional dependence, the dynamic nature of GAT does not invalidate the operator-based analysis. Conditional on the attention matrices generated during the forward pass, each $\hat{A}^{(\ell)}_{\mathrm{GAT}}(H^{(\ell-1)})$ is still nonnegative and row-stochastic:

$$\alpha^{(\ell)}_{ij} \geq 0, \qquad \sum_{j\in\mathcal{N}(i)}\alpha^{(\ell)}_{ij} = 1. \tag{187}$$

Hence, $P_{\mathrm{GAT}}(H)$ remains a nonnegative row-stochastic propagation operator after conditioning on $H$. The previous propagation-and-pooling analysis can therefore be applied conditionally on the realized dynamic kernel. The main change is that the coverage terms and effective sample size are no longer fixed constants determined only by the graph topology; instead, they become input-dependent quantities controlled by the attention mechanism.

This viewpoint also clarifies why GAT may strengthen the norm signal. Since the attention score $e^{(\ell)}_{ij}$ is computed from transformed node features, neighbors whose representations are better aligned with the learned directions of $(W^{(\ell)}, a_\ell)$ tend to receive larger attention weights. Thus, compared with uniform or degree-normalized aggregation, dynamic attention can increase the effective contribution of model-utilized neighbors to the pooled representation.

In the following sections, we analyze GAT from this dynamic-convolution perspective. Attention changes the effective coverage and the constants in the norm-growth law, but under non-degenerate coverage it preserves the leading growth order, including the logarithmic behavior under max pooling.

### M.2 GAT as Conditional Dynamic Convolution

We now formalize the dynamic-convolution view of GAT. The main difficulty, compared with GCN and GIN, is that the propagation kernel of GAT is not fixed before the forward pass. Instead, it is generated from the current hidden representations. Nevertheless, once the attention weights are realized, a GAT layer is still a nonnegative row-stochastic message-passing operator. Therefore, the previous operator-based norm-growth analysis can be applied conditionally on the realized dynamic kernels.

For the $\ell$-th GAT layer, let

$$\hat{A}^{(\ell)}_{\mathrm{GAT}}\left(H^{(\ell-1)}\right) = \left[\alpha^{(\ell)}_{ij}\right]_{i,j} \tag{188}$$

denote the attention matrix induced by the hidden representations $H^{(\ell-1)}$. The layer update can be written as

$$\tilde{h}^{(\ell)}_i = \sum_{j\in\mathcal{N}(i)}\alpha^{(\ell)}_{ij}W^{(\ell)}h^{(\ell-1)}_j, \qquad h^{(\ell)}_i = \sigma\left(\mathrm{LN}\left(\tilde{h}^{(\ell)}_i\right)\right), \tag{189}$$

where

$$\alpha_{ij}^{(\ell)} = \frac{\exp(e_{ij}^{(\ell)})}{\sum_{m\in\mathcal{N}(i)} \exp(e_{im}^{(\ell)})}, \qquad \sum_{j\in\mathcal{N}(i)} \alpha_{ij}^{(\ell)} = 1, \qquad \alpha_{ij}^{(\ell)} \geq 0. \tag{190}$$

Thus, conditioned on $H^{(\ell-1)}$, the matrix $\hat{A}_{\mathrm{GAT}}^{(\ell)}(H^{(\ell-1)})$ is a valid row-stochastic propagation kernel.

We next project the GAT update onto the principal subspace of the learned transformation. Following the notation in Appendix G, define

$$\alpha_j^{(\ell-1)} := U_{\ell-1}^\top h_j^{(\ell-1)} \tag{191}$$

as the principal-subspace coefficient of node $j$ at layer $\ell-1$. Under the same low-rank alignment condition

$$W^{(\ell)} = U_\ell S_\ell U_{\ell-1}^\top + E_\ell, \tag{192}$$

we have, conditional on the realized attention matrix,

$$U_\ell^\top \tilde{h}_i^{(\ell)} = S_\ell \sum_{j\in\mathcal{N}(i)} \alpha_{ij}^{(\ell)} \alpha_j^{(\ell-1)} + \varepsilon_{i,\mathrm{GAT}}^{(\ell)}. \tag{193}$$

The residual term satisfies the same type of bound as in the static-kernel case:

$$\begin{aligned} \left\|\varepsilon_{i,\mathrm{GAT}}^{(\ell)}\right\| \leq \tau_\ell \sum_{j\in\mathcal{N}(i)} \alpha_{ij}^{(\ell)} \left\|\Pi_{U_{\ell-1}} h_j^{(\ell-1)}\right\| \\ + \delta_\ell \sum_{j\in\mathcal{N}(i)} \alpha_{ij}^{(\ell)} \left\|\Pi_{U_{\ell-1}}^\perp h_j^{(\ell-1)}\right\| + \eta_\ell. \end{aligned} \tag{194}$$

Since the attention weights are nonnegative and sum to one, these error terms remain convex averages of the principal and residual components. Hence, dynamic attention does not introduce uncontrolled amplification in the node-wise propagation; instead, it changes how propagation mass is distributed over neighbors.

After incorporating the typical channel-wise effects of LayerNorm and ReLU into a nonnegative gain matrix $\Gamma_\ell$, the principal coefficients satisfy the conditional layer-wise relation

$$\alpha^{(\ell)} \approx \hat{A}_{\mathrm{GAT}}^{(\ell)}\left(H^{(\ell-1)}\right) \alpha^{(\ell-1)} \left(\Gamma_\ell S_\ell\right)^\top + \varepsilon_{\mathrm{GAT}}^{(\ell)}. \tag{195}$$

Iterating this relation over $L$ layers gives

$$\alpha^{(L)} \approx \left(P_{\mathrm{GAT}}(H)\alpha^{(0)}\right) G^\top + \varepsilon_{\mathrm{GAT}}^{(0\to L)}, \tag{196}$$

where

$$P_{\mathrm{GAT}}(H) = \hat{A}_{\mathrm{GAT}}^{(L)}\left(H^{(L-1)}\right) \cdots \hat{A}_{\mathrm{GAT}}^{(1)}\left(H^{(0)}\right), \qquad G = \prod_{\ell=1}^{L} \Gamma_\ell S_\ell. \tag{197}$$

Conditioned on the forward trajectory $H$, the operator $P_{\mathrm{GAT}}(H)$ is nonnegative and row-stochastic, because it is a product of nonnegative row-stochastic matrices. Therefore, GAT preserves the same operator structure as the previous analysis, except that the propagation operator is now input-dependent.

This conditional operator form gives a natural interpretation of the graph representation. For a linear pooling operator with weight vector $w$, we have

$$z = w^\top \alpha^{(L)} \approx \left(w^\top P_{\mathrm{GAT}}(H)\alpha^{(0)}\right) G^\top. \tag{198}$$

Define the dynamic effective reach distribution as

$$\pi_{S,\mathrm{GAT}}^\top(H) := w^\top P_{\mathrm{GAT}}(H). \tag{199}$$

For an informative node set $I$, its dynamic coverage is

$$\pi_{S,\mathrm{GAT}}(I;H) := \sum_{j\in I} \pi_{S,\mathrm{GAT},j}(H). \tag{200}$$

Then, under the same informative/non-informative feature model as before,

$$\mathbb{E}\left[z \mid P_{\mathrm{GAT}}(H)\right] \approx \pi_{S,\mathrm{GAT}}(I;H)\mu_I^\top G^\top + \text{lower-order terms}. \tag{201}$$

Thus, the role of dynamic attention is to replace the static coverage term $\pi_S(I)$ with the input-dependent coverage term $\pi_{S,\mathrm{GAT}}(I;H)$.

The attention mechanism can further strengthen the contribution of informative nodes. If an informative neighbor $j \in I$ is better aligned with the learned attention direction than a non-informative neighbor $m \notin I$, then its attention score is larger. In particular, whenever

$$e_{ij}^{(\ell)} - e_{im}^{(\ell)} \geq \Delta_\ell > 0, \tag{202}$$

the corresponding attention weights satisfy

$$\frac{\alpha_{ij}^{(\ell)}}{\alpha_{im}^{(\ell)}} = \exp\left(e_{ij}^{(\ell)} - e_{im}^{(\ell)}\right) \geq \exp(\Delta_\ell). \tag{203}$$

Therefore, aligned informative neighbors can receive exponentially larger propagation mass than weakly aligned neighbors. This explains why GAT may increase the effective coverage of model-utilized nodes compared with uniform or degree-normalized aggregation.

However, this effect should be interpreted through the dynamic coverage term, rather than as an unconditional increase in all cases. If the attention distribution is moderately spread over multiple informative nodes, then $\pi_{S,\mathrm{GAT}}(I;H)$ grows as more informative nodes are selected. If the attention becomes extremely sharp, it may instead behave like a local winner-takes-all selector and concentrate on only a few informative nodes. In that case, the leading growth law is governed by the number of informative nodes that remain effectively reachable under the dynamic attention kernel, rather than by the raw cardinality $|I|$ alone.

Consequently, the dynamic-convolution view leads to the following principle: GAT does not change the form of the norm-growth mechanism, but it changes the effective coverage through which informative nodes contribute to the pooled representation. The next subsection makes this explicit for max pooling by introducing a dynamic effective number of informative candidates and showing that the logarithmic growth law is preserved whenever this effective number scales linearly with the number of selected informative nodes.

### M.3 Max Pooling under Dynamic Attention

We now analyze how the dynamic-convolution view affects the max-pooling conclusion. The key point is that max pooling is governed not by the total amount of attention mass assigned to informative nodes, but by how many informative candidates can effectively enter the coordinate-wise maximum. Therefore, for GAT, the raw number of informative nodes $k = |I|$ should be replaced by an input-dependent effective number of informative candidates.

Recall from the previous subsection that, conditional on the realized attention trajectory, the principal-subspace representation of GAT can be written as

$$\alpha^{(L)} \approx \left(P_{\mathrm{GAT}}(H)\alpha^{(0)}\right) G^\top + \varepsilon_{\mathrm{GAT}}^{(0\to L)}, \tag{204}$$

where

$$P_{\mathrm{GAT}}(H) = \hat{A}_{\mathrm{GAT}}^{(L)}\left(H^{(L-1)}\right)\cdots\hat{A}_{\mathrm{GAT}}^{(1)}\left(H^{(0)}\right) \tag{205}$$

is a nonnegative row-stochastic propagation operator after conditioning on the forward trajectory $H$.

For max pooling over a node set $S$, the $c$-th coordinate of the graph representation is

$$z_c = \max_{i\in S} \alpha_{i,c}^{(L)}. \tag{206}$$

Substituting the conditional operator form gives

$$z_c \approx \max_{i\in S}\left[\sum_j P_{\mathrm{GAT},ij}(H)\left(\alpha_j^{(0)}G^\top\right)_c + \varepsilon_{i,c}\right]. \tag{207}$$

For sum or mean pooling, the informative contribution is naturally measured by the total arrival mass

$$\pi_{S,\mathrm{GAT}}(I;H)=\sum_{j\in I}\pi_{S,\mathrm{GAT},j}(H). \tag{208}$$

For max pooling, however, a more relevant quantity is whether each informative node can reach at least one pooled node with non-negligible strength. We therefore define the dynamic effective sample size

$$k_{\mathrm{eff}}(S,I;H,\tau):=\sum_{j\in I}\mathbf{1}\left\{\max_{i\in S}P_{\mathrm{GAT},ij}(H)\geq\tau\right\},\qquad\tau\in(0,1]. \tag{209}$$

This quantity counts the number of informative nodes that effectively participate in the max-pooling competition under the dynamically generated attention kernel. Unlike the static case, $k_{\mathrm{eff}}$ is input-dependent because $P_{\mathrm{GAT}}(H)$ depends on the hidden representations.

We assume a non-degenerate dynamic coverage condition: there exists a constant threshold $\tau>0$ and constants $0<c_1\leq c_2\leq 1$ such that, with high probability over the forward trajectory,

$$c_1k\leq k_{\mathrm{eff}}(S,I;H,\tau)\leq c_2k. \tag{210}$$

This condition rules out the degenerate case where attention always collapses onto only a constant number of informative nodes, regardless of how many informative nodes are selected. It still allows attention to be highly non-uniform and input-dependent; it only requires that the number of informative candidates reaching the pooled region grows proportionally with $k$.

Under this condition, the max-pooling analysis follows the same extreme-value mechanism as before, but with $k_{\mathrm{eff}}$ replacing $k$. To see this, define the effectively reachable informative set

$$I_{\mathrm{eff}}(H,\tau):=\left\{j\in I:\max_{i\in S}P_{\mathrm{GAT},ij}(H)\geq\tau\right\}. \tag{211}$$

For every $j\in I_{\mathrm{eff}}(H,\tau)$, there exists at least one receiver $i_j\in S$ such that

$$P_{\mathrm{GAT},i_jj}(H)\geq\tau. \tag{212}$$

Thus, along an informative principal channel $c$, the received signal at $i_j$ contains a non-vanishing aligned component:

$$\begin{aligned}\alpha^{(L)}_{i_j,c}&\approx P_{\mathrm{GAT},i_jj}(H)\left(\alpha_j^{(0)}G^\top\right)_c+\text{other terms}\\&\geq\tau\left(\alpha_j^{(0)}G^\top\right)_c+\text{lower-order terms.}\end{aligned} \tag{213}$$

Therefore, the coordinate-wise maximum over $S$ is lower bounded, up to lower-order perturbations, by the maximum over the $k_{\mathrm{eff}}$ effectively reachable informative candidates:

$$z_c\gtrsim\max_{j\in I_{\mathrm{eff}}(H,\tau)}\left[\tau\left(\alpha_j^{(0)}G^\top\right)_c+\xi_{j,c}\right], \tag{214}$$

where $\xi_{j,c}$ denotes the residual fluctuation induced by non-informative components, weak dependence, and layer-wise approximation errors.

Under the same exponential-type tail condition of Sec. J.4.2, the maximum over $k_{\mathrm{eff}}$ effective candidates satisfies

$$\mathbb{E}\left[z_c\mid P_{\mathrm{GAT}}(H)\right]=\mu_c(H)+\beta_c(H)\log k_{\mathrm{eff}}(S,I;H,\tau)+O(1), \tag{215}$$

where $\mu_c(H)$ and $\beta_c(H)>0$ absorb the dynamic attention weights, the channel-wise gain $G$, and constant-order perturbations. Equivalently,

$$\mathbb{E}\left[z_c\mid P_{\mathrm{GAT}}(H)\right]=\Theta\left(\log k_{\mathrm{eff}}(S,I;H,\tau)\right). \tag{216}$$

Since the graph embedding norm is obtained from finitely many principal channels, the same leading order carries over to the vector norm:

$$\mathbb{E}\left[\|z\|_2\mid P_{\mathrm{GAT}}(H)\right]=\Theta\left(\log k_{\mathrm{eff}}(S,I;H,\tau)\right). \tag{217}$$

Finally, under the non-degenerate dynamic coverage condition

$$k_{\mathrm{eff}}(S,I;H,\tau)=\Theta(k), \tag{218}$$

we obtain

$$\mathbb{E}\left[\|z\|_2 \mid P_{\mathrm{GAT}}(H)\right] = \Theta(\log k). \tag{219}$$

Taking expectation over the forward trajectory preserves the same leading order, as long as the dynamic coverage condition holds with high probability:

$$\mathbb{E}\left[\|z\|_2\right] = \Theta(\log k). \tag{220}$$

This result clarifies the role of GAT's dynamic attention. Attention does not change the extreme-value nature of max pooling. Instead, it changes which informative nodes effectively enter the max competition. If attention assigns non-negligible mass to a growing number of informative nodes, then $k_{\mathrm{eff}}$ grows proportionally with $k$, and the original logarithmic growth law is preserved. If attention becomes extremely sharp and repeatedly selects only a constant number of informative nodes, then the growth may saturate at $\Theta(\log k_{\mathrm{eff}})$ with $k_{\mathrm{eff}} \ll k$.

Thus, the most precise statement for GAT is

$$\mathbb{E}\left[\|z\|_2\right] = \Theta(\log k_{\mathrm{eff}}), \tag{221}$$

and the usual

$$\mathbb{E}\left[\|z\|_2\right] = \Theta(\log k) \tag{222}$$

follows whenever the dynamic attention kernel provides non-degenerate coverage of the selected informative nodes.

This dynamic effective-sample-size view explains why the conclusion for max-pooling GAT is consistent with that for GCN and GIN. GAT modifies the constants, the attention sharpness, and the effective coverage through $P_{\mathrm{GAT}}(H)$, but under non-degenerate dynamic coverage it does not change the leading logarithmic growth order induced by max pooling.

### M.4 Sum and Mean Pooling under Dynamic Attention

We next briefly discuss sum and mean pooling for GAT. Although our empirical validation for GAT focuses on max pooling, the dynamic-convolution perspective also gives a direct extension of the linear growth result for sum and mean pooling.

Conditioned on the realized attention trajectory, the principal-subspace representation of GAT satisfies

$$\alpha^{(L)} \approx \left(P_{\mathrm{GAT}}(H)\alpha^{(0)}\right) G^\top + \varepsilon_{\mathrm{GAT}}^{(0\to L)}, \tag{223}$$

where $P_{\mathrm{GAT}}(H)$ is a nonnegative row-stochastic operator. For a linear pooling operator with weight vector $w$, including sum and mean pooling, the graph representation is

$$z = w^\top \alpha^{(L)} \approx \left(w^\top P_{\mathrm{GAT}}(H)\alpha^{(0)}\right) G^\top. \tag{224}$$

Define the dynamic effective reach distribution as

$$\pi_{S,\mathrm{GAT}}^\top(H) := w^\top P_{\mathrm{GAT}}(H). \tag{225}$$

For the informative node set $I$, define the dynamic informative mass as

$$\pi_{S,\mathrm{GAT}}(I;H) := \sum_{j\in I} \pi_{S,\mathrm{GAT},j}(H). \tag{226}$$

Under the same informative/non-informative feature model used in the previous analysis, informative nodes have a coherent mean component along the principal directions, whereas non-informative nodes are approximately zero-mean in those directions. Therefore, conditional on $P_{\mathrm{GAT}}(H)$,

$$\mathbb{E}\left[z \mid P_{\mathrm{GAT}}(H)\right] \approx \pi_{S,\mathrm{GAT}}(I;H)\mu_I^\top G^\top + \text{lower-order terms}. \tag{227}$$

Thus, unlike max pooling, whose leading behavior is governed by an extreme-value effect over the effective number of candidates, sum and mean pooling are governed by the total dynamic informative mass.

For sum pooling, $w = \mathbf{1}_S$, and hence

$$\pi_{S,\mathrm{GAT}}^\top(H) = \mathbf{1}_S^\top P_{\mathrm{GAT}}(H). \tag{228}$$

If the dynamic attention kernel provides non-degenerate mass coverage of the informative nodes, namely

$$\pi_{S,\mathrm{GAT}}(I;H) = \Theta(k), \tag{229}$$

then the coherent informative component in the pooled representation grows linearly with $k$. Since the variance and leakage terms are lower order under the same bounded-noise and small-leakage assumptions as before, we obtain

$$\mathbb{E}\left[\|z_{\mathrm{sum}}\|_2 \mid P_{\mathrm{GAT}}(H)\right] = \Theta(k). \tag{230}$$

Taking expectation over the forward trajectory preserves the same order when the non-degenerate mass coverage condition holds with high probability:

$$\mathbb{E}\left[\|z_{\mathrm{sum}}\|_2\right] = \Theta(k). \tag{231}$$

For mean pooling, $w = \mathbf{1}_S/|S|$, and therefore

$$\pi^{\mathrm{mean}}_{S,\mathrm{GAT}}(I;H) = \frac{1}{|S|}\pi^{\mathrm{sum}}_{S,\mathrm{GAT}}(I;H). \tag{232}$$

Consequently,

$$\mathbb{E}\left[\|z_{\mathrm{mean}}\|_2\right] = \Theta\left(\frac{k}{|S|}\right). \tag{233}$$

In the controlled setting considered in our norm-growth analysis, the selected subgraph size $|S|$ is fixed while the number of selected informative nodes varies. Thus, the factor $1/|S|$ is constant, and mean pooling exhibits the same linear dependence on $k$:

$$\mathbb{E}\left[\|z_{\mathrm{mean}}\|_2\right] = \Theta(k), \qquad \text{up to the constant factor } \frac{1}{|S|}. \tag{234}$$

This result shows that the dynamic nature of GAT does not change the leading-order behavior of sum and mean pooling. Attention modifies the effective informative mass $\pi_{S,\mathrm{GAT}}(I;H)$ by redistributing propagation weights over neighbors. If this mass grows proportionally with the number of selected informative nodes, then the pooled aligned signal grows linearly.

Therefore, for GAT, the refined pooling-dependent conclusions are:

1. **Sum and mean pooling.** Under non-degenerate dynamic mass coverage and fixed selected subgraph size,
$$\mathbb{E}\left[\|z_{\mathrm{sum/mean}}\|_2\right] = \Theta(k). \tag{235}$$
2. **Max pooling.** Max pooling satisfies
$$\mathbb{E}\left[\|z_{\max}\|_2\right] = \Theta(\log k_{\mathrm{eff}}), \tag{236}$$
which reduces to $\Theta(\log k)$ when the dynamic effective sample size satisfies
$$k_{\mathrm{eff}} = \Theta(k).$$

### M.5 Comparison with GIN and Final Conclusion

We finally summarize how the above arguments extend the GCN-based analysis to GIN and GAT. The main distinction is whether the neighborhood propagation kernel is static or dynamically generated from the current hidden representations.

For GCN, the propagation kernel is static once the input graph is fixed. The normalized adjacency matrix defines a graph-dependent but feature-independent convolution operator. Hence, the node-wise propagation operator $P$ is fixed for a given graph, and the previous analysis directly applies: informative nodes contribute coherent principal-subspace signals, whereas non-informative nodes mainly contribute zero-mean fluctuations. Under non-degenerate coverage, sum and mean pooling accumulate the coherent informative component linearly in the number of informative nodes $k$, whereas max pooling follows an extreme-value mechanism and grows logarithmically:

$$\mathbb{E}\left[\|z_{\mathrm{sum/mean}}\|_2\right] = \Theta(k), \qquad \mathbb{E}\left[\|z_{\max}\|_2\right] = \Theta(\log k). \tag{237}$$

GIN belongs to the same static-propagation category for the purpose of our norm-growth analysis. A standard GIN layer has the form

$$h_i^{(\ell)} = \mathrm{MLP}^{(\ell)}\left((1+\epsilon_\ell)h_i^{(\ell-1)} + \sum_{j\in\mathcal{N}(i)} h_j^{(\ell-1)}\right). \tag{238}$$

The key point is that the coefficients in the neighborhood aggregation are not recomputed from the current hidden representations. In contrast to GAT, GIN does not assign input-dependent attention weights to different neighbors. Its aggregation matrix can be written as a static nonnegative operator

$$\tilde{A}_{\mathrm{GIN}}^{(\ell)} = A + (1+\epsilon_\ell)I, \tag{239}$$

followed by a learnable nonlinear transformation.

Under bounded-degree or bounded-aggregation-scale conditions, the row-scale factors of $\tilde{A}_{\mathrm{GIN}}^{(\ell)}$ only change constants. Equivalently, one may work with a normalized static operator $\hat{A}_{\mathrm{GIN}}^{(\ell)}$ and absorb the remaining scale factors into the layer-wise gain term. Thus, after projecting onto the principal subspace, the GIN update can be written in the same operator form:

$$\alpha^{(\ell)} \approx \left(\hat{A}_{\mathrm{GIN}}^{(\ell)}\alpha^{(\ell-1)}\right)\left(\Gamma_\ell S_\ell\right)^\top + \varepsilon_{\mathrm{GIN}}^{(\ell)}. \tag{240}$$

Iterating over layers gives

$$\alpha^{(L)} \approx \left(P_{\mathrm{GIN}}\alpha^{(0)}\right)G^\top + \varepsilon_{\mathrm{GIN}}^{(0\to L)}, \qquad P_{\mathrm{GIN}} = \prod_{\ell=1}^{L}\hat{A}_{\mathrm{GIN}}^{(\ell)}. \tag{241}$$

Therefore, GIN differs from GCN mainly in the strength of additive aggregation and the subsequent MLP transformation, rather than in the static nature of the propagation kernel.

This static-operator view allows the GCN norm-growth argument to be reused for GIN. For linear pooling, including sum and mean pooling, we have

$$z = w^\top\alpha^{(L)} \approx \left(w^\top P_{\mathrm{GIN}}\alpha^{(0)}\right)G^\top. \tag{242}$$

Define

$$\pi_{S,\mathrm{GIN}}^\top := w^\top P_{\mathrm{GIN}}, \qquad \pi_{S,\mathrm{GIN}}(I) := \sum_{j\in I}\pi_{S,\mathrm{GIN},j}. \tag{243}$$

Under the same informative/non-informative feature model, the mean of the pooled representation satisfies

$$\mathbb{E}[z] \approx \pi_{S,\mathrm{GIN}}(I)\mu_I^\top G^\top + \text{lower-order terms}. \tag{244}$$

If the static GIN propagation operator provides non-degenerate coverage of informative nodes, namely

$$\pi_{S,\mathrm{GIN}}(I) = \Theta(k), \tag{245}$$

then the coherent informative component grows linearly with $k$. The variance and leakage terms remain lower order under the same bounded-noise and small-leakage assumptions. Therefore,

$$\mathbb{E}\left[\|z_{\mathrm{sum}}\|_2\right] = \Theta(k). \tag{246}$$

For mean pooling, the pooling weight is scaled by $1/|S|$. In our controlled setting, the selected subgraph size $|S|$ is fixed. Hence, the same linear dependence holds up to the constant factor $1/|S|$:

$$\mathbb{E}\left[\|z_{\mathrm{mean}}\|_2\right] = \Theta(k). \tag{247}$$

For max pooling, GIN also inherits the same extreme-value argument as GCN. Since the propagation operator is static, the number of informative candidates that effectively reach the pooled region can be defined as

$$k_{\mathrm{eff}}^{\mathrm{GIN}}(S, I; \tau) := \sum_{j\in I}\mathbf{1}\left\{\max_{i\in S} P_{\mathrm{GIN},ij} \geq \tau\right\}. \tag{248}$$

Under the static coverage condition

$$k_{\mathrm{eff}}^{\mathrm{GIN}}(S, I; \tau) = \Theta(k), \tag{249}$$

the coordinate-wise maximum is taken over a number of informative candidates proportional to $k$. Therefore, under the same exponential-type tail condition of Sec. J.4.2,

$$\mathbb{E}\left[\|z_{\max}\|_2\right] = \Theta(\log k). \tag{250}$$

The MLP in GIN can amplify aligned directions and modify constants through the gain matrix $G$, but it does not change the pooling-induced leading order.

GAT, by contrast, requires a dynamic-convolution treatment because its propagation kernel depends on the current hidden representations:

$$P_{\mathrm{GAT}}(H) = \hat{A}_{\mathrm{GAT}}^{(L)}\left(H^{(L-1)}\right) \cdots \hat{A}_{\mathrm{GAT}}^{(1)}\left(H^{(0)}\right). \tag{251}$$

As shown above, once we condition on the realized attention kernels, $P_{\mathrm{GAT}}(H)$ is still nonnegative and row-stochastic. Thus, the same operator analysis applies with dynamic coverage quantities. For sum and mean pooling, the leading term is governed by the dynamic informative mass $\pi_{S,\mathrm{GAT}}(I; H)$; when this mass grows proportionally with $k$, the growth remains linear. For max pooling, the leading term is governed by the dynamic effective sample size $k_{\mathrm{eff}}(S, I; H, \tau)$; when $k_{\mathrm{eff}} = \Theta(k)$, the growth remains logarithmic:

$$\mathbb{E}\left[\|z_{\mathrm{sum/mean}}\|_2\right] = \Theta(k), \qquad \mathbb{E}\left[\|z_{\max}\|_2\right] = \Theta(\log k). \tag{252}$$

Overall, GCN, GIN, and GAT differ in how informative signals are propagated:

1. **GCN.** GCN uses static degree-normalized convolution.
2. **GIN.** GIN uses static sum aggregation followed by learnable MLP transformations.
3. **GAT.** GAT uses dynamic attention-based convolution with input-dependent kernels.

These differences affect constants, effective coverage, and the strength of aligned feature amplification. However, once informative nodes reach the pooled representation in a non-degenerate way, the leading-order growth is determined by the pooling operator. Sum and mean pooling accumulate aligned informative signals additively and therefore yield linear growth in $k$, while max pooling follows an extreme-value mechanism and yields logarithmic growth in the effective number of informative candidates. Thus, the conclusions derived for GCN extend to GIN through the static-operator view, and to GAT through the conditional dynamic-convolution view.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes. The main claims are stated in the abstract and introduction: the paper identifies optimization difficulties of MMI-based GNN explainers, proposes NORMA as a norm-guided post-hoc explainer, and supports the method with theoretical analysis and empirical evaluation. The scope of the claims is aligned with the theoretical assumptions and experiments, including benchmark evaluations and the stated post-hoc explanation setting.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes. Section 6 discusses that high representation norm does not imply causal correctness: if the target GNN relies on spurious shortcuts, NORMA may also highlight them. This is not a NORMA-specific limitation, but a general challenge for faithful post-hoc explainers, including MI/CE-based methods, because they reflect the evidence used by the target model.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Yes. The paper states the assumptions for the norm-growth analysis in Section 3.2, presents Proposition 1 in Section 3.3, and provides the complete theoretical support in Appendix F–M, including the GCN setup, low-rank alignment assumptions, propagation analysis, pooling norm-growth proof, and discussions for LayerNorm, ReLU, GIN, and GAT.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes. The paper specifies the NORMA objective and mask-generation procedure, reports datasets, target GNN architectures, train/test split, optimizer settings, evaluation metrics, and baseline methods. Additional implementation details, including dataset statistics, GNN training settings, explainer hyperparameters, batch sizes, and mask-to-edge-score conversion, are provided in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Yes. The code for reproducing the main experiments is provided in the supplementary material.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes. Section 5.1 specifies the datasets, target GNN architectures, data split, optimizer, learning rate, weight decay, evaluation metrics, and baselines. Appendix D further provides dataset/model statistics, GNN training details, explainer hyperparameters, batch sizes, mask-generation details, and the unified inverse-fidelity evaluation protocol.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Yes. The reported AUROC results are presented as mean ± standard deviation over independent runs with different random seeds.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Yes. We report the computational platform used for the experiments in Appendix D.4, including the operating system, GPU and CPU configuration, and the software stack.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: Yes. We have reviewed the NeurIPS Code of Ethics and ensured that this research conforms to it.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: N/A. The paper proposes a post-hoc GNN explanation method evaluated on standard benchmarks, and we do not identify a direct path to negative societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: N/A. The paper does not release high-risk models or scraped datasets that require misuse safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Yes. We properly cite the original sources of the datasets, benchmark models, and baseline methods used in the paper, and follow their license and terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: N/A. This work does not introduce new datasets, models, or other assets.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: N/A. This work does not involve human subjects, user studies, or human data collection.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: N/A. This work does not use any datasets containing personally identifiable or sensitive information.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: N/A. LLMs were used only for writing assistance, not as part of the core research method.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.