BOTTLENECKMLP: GRAPH EXPLANATION VIA IMPLICIT INFORMATION BOTTLENECK

Anonymous authors
Paper under double-blind review

ABSTRACT

The success of Graph Neural Networks (GNNs) in modeling unstructured data has heightened the demand for explainable AI (XAI) methods that provide transparent, interpretable rationales for their predictions. A prominent line of work leverages the Information Bottleneck (IB) principle, which frames explanation as optimizing for representations that maximize predictive information $I(Z;Y)$ while minimizing input dependence $I(X;Z)$. We show that explicit IB-based losses in GNN explainers provide little benefit beyond standard training: the fitting and compression phases of IB emerge naturally, whereas the variational bounds used in explicit objectives are too loose to meaningfully constrain mutual information. To address this, we propose BottleneckMLP, an architectural module that implicitly enforces the IB principle. By injecting Gaussian noise inversely scaled by node importance, followed by architectural compression, BottleneckMLP amplifies the reduction of $I(X;Z)$ while increasing $I(Z;Y)$. This yields embeddings where important nodes remain structured and clustered, while unimportant nodes drift toward Gaussianized, high-entropy distributions, consistent with progressive information loss under IB. BottleneckMLP integrates seamlessly with current explainers, as well as subgraph recognition tasks, replacing explicit IB terms and consistently improving predictive performance and explanation quality across diverse datasets.

1 INTRODUCTION

Graph-structured data appears in a wide range of domains, including drug design Liu et al. (2023), healthcare Zitnik et al. (2018), social networks Bian et al. (2020), and recommendation systems Chen et al. (2022). Graph Neural Networks (GNNs) have emerged as powerful models for learning from such data, achieving state-of-the-art results in tasks such as node/graph classification Bacciu et al. (2019); Kipf & Welling (2017), link prediction Zhang & Chen (2018), and graph regression Zhang et al. (2024); yet, GNNs remain black-box models. Recent research has focused on developing explainability methods Dai et al. (2024); Li et al. (2023b); Yuan et al. (2022), which are essential to build trust and ensure reliability in sensitive applications such as healthcare and scientific discovery.

Most GNN explainability methods are post-hoc Bajaj et al. (2021a); Baldassarre & Azizpour (2019); Luo et al. (2020a), applying an explainer to a pre-trained black-box model. Recent work instead explores ante-hoc approaches Luong et al. (2024); Miao et al. (2022); Seo et al. (2024a), which train the explainer and classifier jointly to avoid spurious correlations. These intrinsically interpretable GNNs aim to balance accuracy with interpretability, encouraging reliance on ground-truth explanatory features. We focus on ante-hoc graph classification explainers, and additionally show that BottleneckMLP generalizes to post-hoc node classification and subgraph recognition tasks.

The Information Bottleneck (IB) principle Tishby et al. (2000); Tishby & Zaslavsky (2015) formalizes the following tradeoff: the optimal representation Z should capture minimal but sufficient information from X to predict Y. The IB principle is pertinent to graph data, where rich structure and feature dependencies make learning compact, task-relevant representations challenging. Previous work in GNN explainers (ante-hoc Miao et al. (2022), post-hoc Chen et al. (2024), prototype-based Seo et al. (2024b)) have included information bottleneck losses to encourage the learned representation to be sufficient yet minimal.

In this work, we argue that explicitly minimizing $I(X;Z)$ via auxiliary explicit IB losses ('IB Loss') is ineffective. We formalize GNN explanation as an IB problem, noting that cross-entropy training alone induces the information curve: deeper layers naturally compress X (input graph G), retaining only the information in Z (explanatory subgraph G_S) needed for predicting Y. We reconstruct the information plane for GNNs and show consistency with IB theory and empirical findings in representation learning Tishby & Zaslavsky (2015). Our method surpasses IB-based explainers without such terms, improving accuracy and explanation quality. Our key contributions are fourfold:

(1) **Ineffectiveness of Explicit IB in Graph Settings** We show that explicit IB loss terms are ineffective for explanation in the graph setting, as structural dependencies violate the i.i.d. assumptions underlying IB theory.

(2) **BottleneckMLP: A General Implicit Architectural IB Module** We propose BottleneckMLP, a model-agnostic architectural primitive that implicitly enforces the IB principle without variational bounds or auxiliary losses. By injecting importance-scaled Gaussian noise and applying architectural compression (via an MLP), it drives unimportant nodes toward Gaussianized high-entropy embeddings while preserving structured clusters of important nodes, yielding compact, interpretable representations.

(3) **Gaussianization Encourages Forgetting** We show that the Gaussianization effect induced by BottleneckMLP serves as a natural mechanism for forgetting task-irrelevant information. By injecting noise inversely proportional to node importance, uninformative node representations are progressively pushed toward high-entropy, Gaussian-like embeddings, removing spurious correlations. Relevant information is retained in clustered, low-entropy embeddings, while irrelevant information is forgotten through Gaussianization.

(4) **Empirical Validation** In comprehensive experiments on ante-hoc graph explainers, post-hoc node classification explainers, and subgraph recognition models, BottleneckMLP consistently outperforms explicit IB losses in both explanation quality and model performance, demonstrating that implicit IB is more effective for interpretable graph learning.

2 PRELIMINARIES

Mutual information (MI) is a symmetric measure of how much one random variable reduces the uncertainty of another random variable:

$$I(X;Z) = H(X) - H(X|Z), \tag{1}$$

where $H(X)$ denotes the entropy of random variable X. (A complete list of all symbols and notation used in the paper appears in Appendix A.)

Information Bottleneck, proposed in Tishby et al. (2000), provides a framework to learn a minimal sufficient representation Z that preserves only the aspects of X relevant to Y, quantified by the mutual information $I(X;Y)$. Assuming X and Y are statistically dependent, with Y implicitly distinguishing between relevant and irrelevant features in X, the goal is for Z to retain all information needed to predict Y while discarding irrelevant details. The general IB objective is:

$$\min_{p(z|x)} \left[-I(Z;Y) + \beta I(X;Z)\right], \quad \beta \geq 0 \tag{2}$$

where the Lagrange multiplier β balances relevant information $I(Z;Y)$ and compression $I(X;Z)$.

IB in Deep Learning. Tishby & Zaslavsky (2015) interpret DNN training via the IB framework, where Shwartz-Ziv & Tishby (2017) identify two distinct phases in MI dynamics:

(1) **Fitting Phase**: Cross-entropy minimization increases $I(X;Z)$ and $I(Z;Y)$ as the network fits the data.

(2) **Compression Phase**: Layers discard task-irrelevant information, reducing $I(X;Z)$ while preserving or increasing $I(Z;Y)$. Proceeds slowly and without explicit regularization.

The two-phase dynamic of DNN training emerges in our graph experiments due to BottleneckMLP, marking the first such observation in the graph setting. In graph learning, this behavior is especially desirable as embeddings must encode rich structural, node, and edge-level features into compact, low-dimensional representations. We leverage this insight in our architectural mechanism that

achieves stronger implicit compression while improving both predictive accuracy and explanatory subgraph quality.

3 Related Work

3.1 IB-Based Explainers

A key challenge in generating explanatory subgraphs is their varying size, making fixed-size constraints ineffective Kakkad et al. (2023). To address this, these information-constrained methods adopt the IB principle Tishby et al. (2000), which limits retained information rather than subgraph size. Ante-hoc explainers such as GSAT Miao et al. (2022), PGIB Seo et al. (2024b), and TGIB Seo et al. (2024a) incorporate IB objectives to encourage minimal sufficient representations:

$$\min_{\phi} -I(G_S; Y) + \beta I(G_S; G), \quad \text{s.t. } G_S \sim g_\phi(G). \tag{3}$$

GSAT learns edge attention weights to suppress irrelevant features, sampling G_S from $P_\phi(G_S|G)$. PGIB introduces prototypes G_p and modifies the objective to include $I(Y; G_S, G_p)$ and $I(Y; G_p|G_S) + \beta I(G; G_S)$. TGIB, a temporal variant, extracts bottleneck subgraphs R_k from temporal neighborhoods, using an analogous IB objective. While these methods rely on variational upper bounds or contrastive loss to constrain $I(X; Z)$, we show these bounds are too loose to enforce meaningful compression. In contrast, our BottleneckMLP achieves the compression phases of the IB curve effectively, without requiring explicit IB loss terms.

3.2 Compression in Deep Learning

Several studies have shown that DNNs naturally undergo an implicit compression phase during supervised training. Scabini & Bruno (2023) use complex network theory to show that emergent motifs arise during training without explicit regularization, supporting the IB perspective Tishby & Zaslavsky (2015). Similarly, simple fully connected layers improve CNN generalization Basha et al. (2020); Kocsis et al. (2022) even without explicit compression losses. These results suggest that architectural biases alone can induce compact, informative representations. Our BottleneckMLP demonstrates this in GNN explainability; compression can emerge directly from architecture and cross-entropy training without IB loss terms, bridging implicit IB dynamics with graph explanations.

3.3 Gaussianization of Representations

Eftekhari & Papyan (2025) show that Gaussian distributions are both the most efficient signal representation and the worst-case noise. They propose a mechanism that enforces Gaussianity in neural representations, where injecting Gaussian noise and normalization improve generalization and robustness across architectures. Agrawal et al. (2020) extend infinite-width theory to bottleneck neural network Gaussian processes (NNGPs), showing that unlike deep ReLU NNGPs which lose discriminative power, bottleneck layers preserve task-relevant information by acting as information-preserving compression points. These results suggest that structured architectures (e.g., bottlenecks, Gaussianized activations) naturally promote robustness and generalization, supporting our view that compression and explainability need not rely on explicit IB constraints. Our BottleneckMLP follows this principle, achieving effective compression and explanation purely through architectural design.

4 BottleneckMLP

We begin by identifying critical limitations of explicit IB-based approaches in graph explainability (Section 4.1), then introduce our BottleneckMLP module as an architectural solution (Section 4.2). We provide theory for why this approach is more effective than explicit IB losses (Section 4.3), followed by empirical validation of our theoretical results (Sections 4.3.3 and 5).

4.1 Limitations of Explicit Information Bottleneck in Graph Explainability

While some post-hoc explainers Bajaj et al. (2021b); Luo et al. (2020b) impose sparsity, budget, or connectivity constraints on explanatory subgraphs, the predominant strategy across IB-based meth-

ods remains the use of variational upper bounds. However, these approaches face fundamental limitations when applied to graph-structured data. We describe the general approach used by these ante-hoc IB explainers below, and give explainer-specific details in Appendix I.

Variational Upper Bounds are Loose The terms of the Lagrangian in Equation 2 cannot be computed directly, and require the integrals for $I(X;Z)$ and $I(Z;Y)$. The marginal $p(z)$ and the true posterior $p(y|z)$ are intractable, and variational bounds for mutual information Poole et al. (2019) are used in machine learning Du et al. (2020); Dai et al. (2018); Bao (2021); Li et al. (2023a). Maximizing $I(Z;Y)$ reduces to the usual cross-entropy (CE) loss:

$$I(Z;Y) = \mathbb{E}_{p(y,z)}\left[\log p(y|z)\right] - H(Y). \tag{4}$$

Almost all ante-hoc graph explainers Lee et al. (2023); Miao et al. (2022; 2023); Seo et al. (2024a;b) rely on variational approximations or naive priors which lead to loose or ineffective bounds. This results in insufficient enforcement of the information constraint, allowing the learned subgraphs to retain excessive or redundant information from the input. The variational upper bound loss $\mathcal{L}_{VUB}$ for graph explainers is defined as:

$$\mathcal{L}_{VUB} := \mathrm{KL}(P_\phi(G_S \mid G) \,\|\, Q(G_S)), \tag{5}$$

where edges e_{ij} in $Q(G_S)$ are parameterized by $e_{ij} \sim \text{Bernoulli}(r)$ edges $\hat{e}_{ij}$ are parametrized by $\hat{e}_{ij} \sim Bernoulli(\phi_{ij})$. We refer to this as simply 'IB Loss' throughout the paper. PGIB uses a contrastive loss (Appendix I to minimize $I(X;Z)$, which we again refer to as 'IB Loss'.

Structural Dependencies Break Node-Level Independence Assumptions GNNs compute node embeddings via recursive message passing: $h_v^{(k)} = \texttt{UPDATE}^{(k)}(h_v^{(k-1)}, \texttt{AGGREGATE}^{(k)}(\{h_u^{(k-1)} : u \in N(v)\}))$. Since each $h_v^{(k)}$ aggregates information from multi-hop neighborhoods, node features are interdependent. This violates the i.i.d assumptions underlying variational IB bounds, making KL-based regularizers over node distributions ineffective in capturing structural dependencies. We elaborate on this in Section 4.3.1.

Empirical Evidence of Ineffectiveness Sections 4.3.3 and 5 provide empirical results showing that explicit IB constraints fail to regulate information effectively, motivating our implicit, architecture-driven alternative based on selective forgetting of information.

4.2 BottleneckMLP Architecture

We propose BottleneckMLP, a two-component architectural module that implicitly enforces the IB principle without requiring explicit IB Loss terms.

Component 1: Importance-Weighted Gaussian Noise For each node embedding $Z_i \in \mathbb{R}^d$ with importance scores $\alpha_i \in (0,1)$, we inject noise as:

$$Z_i = f(X) + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}\Big(0, (\frac{\sigma^2}{\alpha_i})I_d\Big), \tag{6}$$

where σ is a tunable hyperparameter, $f(X)$ is the GNN embedder, and I_d is the d-dimensional identity matrix. This mechanism scales the variance of injected Gaussian noise inversely with importance, so that nodes deemed unimportant are perturbed more heavily, thereby pushing them toward high-entropy representations. We present our theoretical results for the component in Section 4.3.2

Component 2: Progressive Compression via MLP Layers Following Tishby & Zaslavsky (2015), we use an MLP to compress representations and filter task-irrelevant noise. The default configuration is $h \longrightarrow \frac{h}{4} \longrightarrow h$, with ReLU activations between layers. We note that determining the optimal architecture of the BottleneckMLP for a given dataset and explainer model is analogous to hyperparameter tuning. This compression retains salient information while discarding noise, promoting compact, meaningful representations.

Integration with Existing Explainers BottleneckMLP integrates seamlessly into existing ante-hoc explanation pipelines (Figure 1), operating directly on the GNN embeddings before subgraph extraction. This modular design replaces explicit IB loss terms across different explainer architectures.

Importance Score Computation Node importance scores are computed using the explainer's existing attention or selection mechanism. GSAT uses attention weights from the stochastic attention mechanism, PGIB uses prototype similarity scores, and TGIB uses temporal neighborhood relevance scores. This ensures compatibility while leveraging each method's inherent importance estimation.

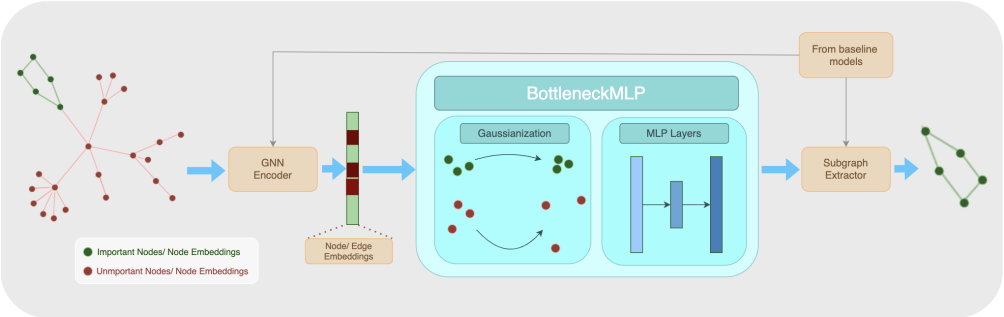


Figure 1: BottleneckMLP method for a general ante-hoc (intrinsically interpretable) GNN pipeline. Our general module acts on the embeddings produced by the explainer's GNN module, and the transformed embeddings are directed do the subsequent subgraph extractor component.

4.3 Theory

In the following subsections, we provide theoretical justification for why explicit IB methods fail on graphs (Section 4.3.1), and grounding for how BottleneckMLP effectively (and implicitly) enforces IB by effecting latent space representation dynamics (Section 4.3.2). We empirically validate the effectiveness of BottleneckMLP and current approaches failing to reduce $I(X;Z)$ (Section 4.3.3). We provide further empirical evidence on theory of latent space dynamics in Section 5.

4.3.1 Why Explicit IB Fails on Graphs

Explicit IB relies on the *Asymptotic Equipartition Property (AEP)*, which holds when data is generated from a stationary, ergodic process with primarily local dependencies. In images or speech, each variable (e.g., a pixel or phoneme) depends mostly on its local neighborhood and is conditionally independent of distant variables given that neighborhood (low global dependence). The AEP theorem Breiman (1957) states that for a sequence $X_1, X_2, \ldots$ from distribution $p(x_1, x_2, \ldots)$

$$\lim_{n\to\infty} -\tfrac{1}{n}\log p(x_1,\ldots,x_n) = H(X). \tag{7}$$

Thus, for large n, almost all patterns are *typical*. Under local dependence, the joint distribution can be approximated by products of localized conditionals:

$$p(x_1,\ldots,x_n \mid P) \approx 2^{-nH(X|P)} \quad \text{for typical partitions } P. \tag{8}$$

In systems with low global connectivity and primarily local dependencies, conditional probabilities can be factorized locally and averaged via the Central Limit Theorem. Under the *typicality* assumption, a similar effect holds in information theory, causing $I(X;Z)$ and $I(Z;Y)$ to concentrate and enabling reliable estimation from partitions of $p(X,Y)$.

$$I(X;Z) = \mathbb{E}_{X,Z}\left[\log \frac{p(x\mid z)}{p(x)}\right] = \mathbb{E}_{X,Z}\left[\sum_i \log \frac{p(x_i \mid \mathcal{N}(x_i), z)}{p(x_i \mid \mathcal{N}(x_i))}\right], \tag{9}$$

$$I(Z;Y) = \mathbb{E}_{Z,Y}\left[\log \frac{p(y\mid z)}{p(y)}\right] = \mathbb{E}_{Z,Y}\left[\sum_x p(y\mid x)\prod_i p(x_i \mid \mathcal{N}(x_i), z) - \log p(y)\right], \tag{10}$$

were $\mathcal{N}(x_i)$ denotes the neighborhood of x_i. These assumptions justify the use of variational bounds in estimating $I(X;Z)$ and $I(Z;Y)$. However, they break down in graphs, where features are not i.i.d. and nodes are structurally entangled, and where the patterns are not large enough to be *typical*.

Graphs Break AEP Assumptions Graphs violate the requisite conditions for AEP. Structural entanglement creates strong global dependencies, the distribution $P(G)$ is not factorized over nodes or edges (also addressed in Wu et al. (2020)), and most graphs are too small to exhibit typicality. As a result, variational estimates of $I(X;Z)$ collapse to KL terms that assume i.i.d. sampling, systematically underestimating dependencies. Empirical evidence of these dependencies is provided in Appendix E.3. Thus, explicit IB losses cannot reliably control information flow in GNNs.

4.3.2 IMPLICIT IB THROUGH GAUSSIANIZATION

Implicit Realization of the IB Lagrangian BottleneckMLP implicitly optimizes the IB objective without requiring variational bounds. Concretely, we obtain:

$$\min_{p(z|x)} -I(Z;Y)+\beta I(X;Z) \rightsquigarrow \begin{cases} I(Z;Y) & \text{is preserved via low-noise important nodes,} \\ I(X;Z) & \text{is reduced via Gaussianization of unimportant nodes.} \end{cases}$$

Decomposing the input representation, $I(X;Z) = I(X;Z_{imp}) + I(X;Z_{unimp})$, and noting that unimportant nodes are conditionally independent of the target given the important nodes, $I(Z_{unimp};Y \mid Z_{imp}) = 0$, reducing $I(X;Z_{unimp})$ via noise does not hurt prediction. Thus BottleneckMLP achieves the minimal sufficient representation postulated by the IB principle, while avoiding the weaknesses of explicit IB loss functions in graphs.

Selective Forgetting of Information Our approach achieves compression by selectively forgetting information about unimportant nodes ($I(X;Z)$ minimization) while preserving structure around important ones. Gaussianization serves as a natural mechanism for information loss. By the Central Limit Theorem, repeated independent noise-injection drives convergence towards Gaussianity (since the embeddings aggregate many independent perturbations).

For each unimportant node i we define

$$Z_i = f_i(X) + t_i\,\epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, I_d), \quad t_i := \frac{\sigma}{\alpha_i},\ \alpha_i \in (0,1), \tag{11}$$

where $f_i(X) \in \mathbb{R}^d$ is the output of the encoder, and σ is a fixed hyperparameter, and and the noise terms $\{\epsilon_i\}$ are sampled independently at each forward pass. We assume $Z_i \mid X$ admits a smooth density with finite Fisher information.

Below we formalize the theoretical groundings of BottleneckMLP. We refer the reader to B for the proofs of Lemma 1 and Lemma 2.

Lemma 1 (Monotonicity of Conditional Entropy) *Let $Z_i = f_i(X) + \sqrt{t_i}\,N_i$ as above. Then the conditional entropy of Z_i given X satisfies the multivariate De Bruijn identity:*

$$\frac{d}{dt_i} H(Z_i \mid X) = \frac{1}{2}\mathrm{Tr}(J(Z_i \mid X)) \geq 0,$$

where $J(Z_i \mid X)$ is the Fisher information matrix. Therefore conditional entropy $H(Z_i \mid X)$ monotonically increases.

Lemma 2 (Bounded Marginal Entropy) *The marginal entropy of Z_i is bounded by:*

$$H(Z_i) \leq \frac{1}{2}\log\Big((2\pi e)^d \det(\mathrm{Cov}(f_i(X)) + t_i I_d)\Big).$$

Theorem 1 (Mutual Information Reduction) *As noise variances $\{t_i\}_{i\in U}$ increase: (1) $I(X;Z_{\mathrm{unimp}})$ decreases monotonically, and (2) $\lim_{\min_{i\in\mathrm{unimp}} t_i\to\infty} I(X;Z_{\mathrm{unimp}}) = 0$.*

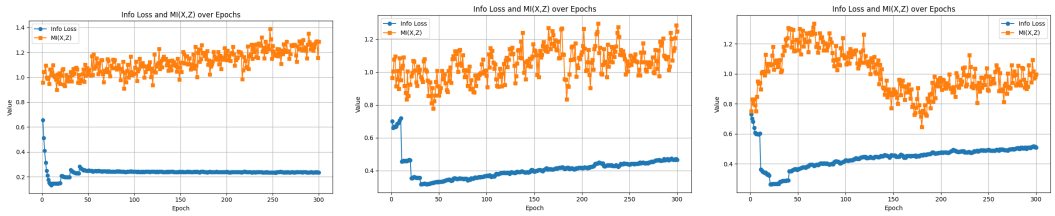
Proof

(1) By the chain rule for MI, $I(X;Z) = I(X;Z_{\mathrm{unimp}}) + I(X;Z_{\mathrm{imp}} \mid Z_{\mathrm{unimp}})$. For the unimportant term, $I(X;Z_{\mathrm{unimp}}) = H(Z_{\mathrm{unimp}}) - H(Z_{\mathrm{unimp}} \mid X)$. By Lemma 1, $H(Z_{\mathrm{unimp}} \mid X)$ increases monotonically with $\{t_i\}$; by Lemma 2 the marginal entropy $H(Z_{\mathrm{unimp}})$ is bounded above. Consequently $I(X;Z_{\mathrm{unimp}})$ decreases as the noise variances increase.

(2) As $t_i \to \infty$ for $i \in$ unimp, the noise dominates the signal: $Z_i \approx t_i\epsilon_i$. Since ϵ_i is independent of X, we get $I(X;Z_i) \to 0$ for each $i \in$ unimp. For the joint MI, note that while embeddings $\{f_i(X)\}$ may be correlated, the noise terms $\{\epsilon_i\}$ are independent. As $t_i \to \infty$, we have $Z_{\mathrm{unimp}} \approx [t_1\epsilon_1, \ldots, t_{|\mathrm{unimp}|}\epsilon_{|\mathrm{unimp}|}]$ where the ϵ_i are independent of X, giving $I(X;Z_{\mathrm{unimp}}) \to 0$.

Stochastic Relaxation For DNNs, after initial fitting, gradient noise induces stochastic relaxation where the network implicitly maximizes conditional entropy $H(Z|X)$, minimizing $I(X;Z) = H(Z) - H(Z|X)$ without explicit regularization Tishby & Zaslavsky (2015); Tishby et al. (2000). In the graph setting, BottleneckMLP is the key architectural component that elicits this phenomenon.

4.3.3 BOTTLENECKMLP EFFECTIVELY ENFORCES IB DYNAMICS



(a) Original (b) w/o IB loss (c) w/o IB loss + BottleneckMLP

Figure 2: Mutual information $I(X;Z)$ (orange) and IB Loss (blue) (Equation 5) visualizations over epochs (GSAT on MUTAG). BottleneckMLP enforces compression effectively, whereas explicit IB Loss terms do not. (a) and (b) do not exhibit $I(X;Z)$ minimization. (c) exhibits both the fitting ($I(X;Z)\uparrow$) and compression ($I(Z;Y)\downarrow$) phases consistent with Tishby & Zaslavsky (2015). Note that minimization of IB Loss (blue) does not correlate with $I(X;Z)$ (orange).

In Figure 2c, $I(X;Z)$ rises early as task-relevant features are captured, then declines in later epochs, reflecting effective compression. This shows that our architecture encourages forgetting irrelevant details and produces IB dynamics absent in prior methods. HSIC analysis further confirms reduced dependence between X and Z across layers (Appendix E.4). By contrast, explicit IB losses fail: in 2a (GSAT) and 2b (GSAT w/o IB loss), $I(X;Z)$ grows monotonically and IB Loss curves misalign with actual dynamics, highlighting the limits of explicit IB and the advantage of our implicit, architecture-driven approach.

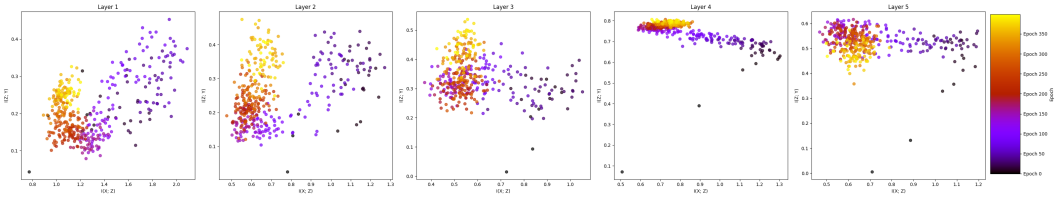


Figure 3: BottleneckMLP enforces IB on graph embeddings in the information plane $I(X;Z_i)$ vs. $I(Z_i;Y)$ over layers, i in range 1–5, and epochs (purple to yellow). Information curve Tishby & Zaslavsky (2015) naturally appears with CE minimization. Compression and concentration are more apparent in the later layers, consistent with the theory in Section 4.3.1.

BottleneckMLP effectively reproduces the information plane dynamics originally observed in DNNs by Tishby & Zaslavsky (2015), now extended to the graph domain with GNNs, as shown in Figure 3. Both the characteristic fitting and compression phases are clearly visible, demonstrating that BottleneckMLP successfully induces the expected IB behavior in graph representations.

5 EXPERIMENTS

Section 5.1 analyzes the distributional properties of node embeddings over training via LNSA and embedding drift to validate our theory on Selective Forgetting of Information given in Section 4.3.2, Section 5.2 reports the improved performance of BottleneckMLP versus baselines and ablations. We focus on ante-hoc IB-based explainers. For generalizability of BottleneckMLP, we report improved performance on subgraph recognition and post-hoc node classification tasks in Appendix H.

5.1 REPRESENTATION DYNAMICS OF BOTTLENECKMLP REFLECT IB

We provide strong mechanistic and empirical evidence that BottleneckMLP induces representational dynamics aligned with the goals of the IB framework, selectively reducing $I(X;Z)$ while preserving $I(Z;Y)$. All experiments corroborate that important node embeddings maintain lower entropy and more structured distributions, while unimportant node embeddings progressively approach Gaussian-like high-entropy distributions. This results in the desired $I(X;Z)$ minimization

(4.3.2). We present LNSA and node drift results, and refer the reader to Appendix for convex hull volume (E.2), node linkage (E.3), HSIC (E.4), and additional node drift experiments (F)

Results on Localized Normalized Space Alignment (LNSA) Following Ebadulla et al. (2025), we track how representation geometry evolves across epochs using LNSA, which measures neighborhood alignment across epochs (higher values indicate greater instability) (details in Appendix K). Nodes are grouped by importance: Category 1 (important with important neighbors), Category 2 (important with mixed neighbors), and Category 3 (unimportant with unimportant neighbors). In Figure 4, we observe that Categories 1 and 2 maintain low LNSA over epochs, while Category 3 exhibits high values, reflecting drift toward high-entropy distributions under BottleneckMLP.

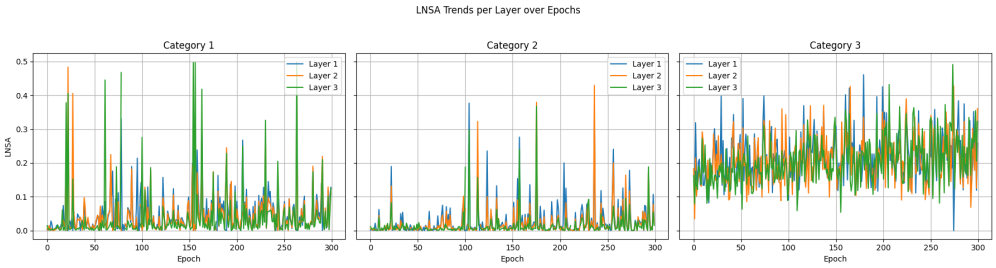
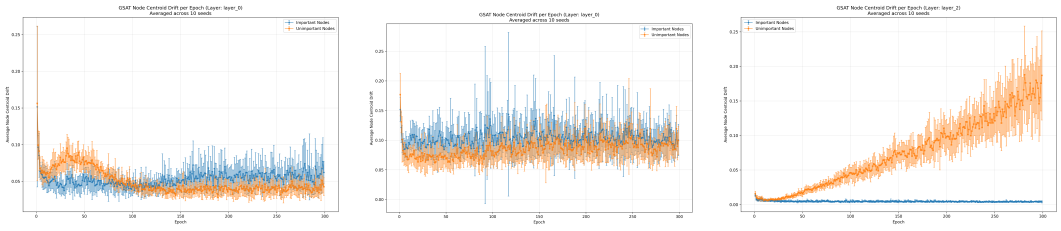


Figure 4: LNSA values for PGIB + BottleneckMLP. Cat 1: important nodes. Cat 2: important nodes with unimportant neighbors. Cat 3: unimportant nodes. Cat 1 and Cat 2 embeddings remain similar, while Cat 3 and has a higher mean LNSA value as embedding structure changes over epochs.

Results on Embedding Drift Figure 5 shows embedding drift under different GSAT configurations. Baseline GSAT yields little separation between node types, and removing the IB Loss term causes uniformly unstable drift. With BottleneckMLP, we achieve the intended IB effect in alignment with out theory: important nodes stabilize with low drift, while unimportant nodes continue drifting toward noisier, Gaussian-like distributions. This pattern holds across all datasets (see Appendix E.1). Notably, these effects occur without an explicit IB loss, demonstrating that BottleneckMLP introduces a powerful implicit bottleneck via architectural constraints alone.



(a) GSAT (b) GSAT w/o IB Loss (c) GSAT + BottleneckMLP

Figure 5: Drift of important (blue) vs. unimportant (orange) nodes across epochs for: (a) GSAT, (b) GSAT w/o IB Loss, and (c) GSAT with BottleneckMLP. BottleneckMLP alone correctly affects representation dynamics. We see the same plots across models and datasets (in Appendix E).

5.2 BottleneckMLP Improves Classification and Explanation

We evaluate if IB Loss adds value beyond supervised training by comparing baseline explainers with/ without the IB Loss, and BottleneckMLP (experimental setup in Appendix C). We evaluate performance on benchmarks using accuracy, explanation AUC-ROC, and Fidelity± (Appendix J).

5.2.1 Graph Task Performance

Table 1 shows the ineffectiveness of IB Loss in GSAT and PGIB, where adding BottleneckMLP consistently improves performance. On BA-2Motifs, where IB Loss removal hurts performance, BottleneckMLP recovers and increases accuracy, replicating and surpassing the role of IB Loss.

Table 1: PGIB/ GSAT Classifier Accuracy. BottleneckMLP increases performance over the original explainer method, and when IB Loss is removed.

	MUTAG	BA-2Motifs	NCI1	PROTEINS
GSAT	0.909±0.033	0.994±0.006	0.689±0.017	0.681±0.036
GSAT w/o IB Loss	0.935±0.043	0.770±0.224	0.743±0.024	0.706±0.040
GSAT w/o IB Loss + BottleneckMLP	**0.949**±0.010	**1.000**	**0.802**±0.019	**0.745**±0.048
PGIB	0.904 ± 0.010	0.628 ± 0.171	0.729 ± 0.024	0.729 ± 0.024
PGIB w/o IB Loss	0.916 ± 0.011	0.896 ± 0.145	**0.774** ± 0.014	0.763 ± 0.022
PGIB w/o IB Loss + BottleneckMLP	**0.925** ± 0.009	**0.963** ± 0.016	0.753 ± 0.019	**0.792** ± 0.018

Table 2: TGIB Link Prediction AP and Explanation AUC/ROC. BottleneckMLP improves performance across datasets, where removal of IB Loss also increases performance

	Link Prediction (AP)			**Explanation AUC/ROC**		
Model	CanParl	USLegis	Wikipedia	CanParl	USLegis	Wikipedia
TGIB	0.789	0.828	0.991	0.588	0.673	0.983
TGIB w/o IB Loss	0.814	0.763	0.994	0.629	0.541	0.989
TGIB w/o IB Loss + Bottleneck MLP	**0.82**	**0.843**	**0.994**	**0.632**	**0.703**	**0.989**

In Table 2, BottleneckMLP improves Average Precision (main metric) and AUC-ROC for TGIB. Performance gain is largest on USLegis, while Wikipedia and CanParl have comparable or slightly better performance. Removing IB Loss boosts performance on Wikipedia and CanParl but reduces it on USLegis; adding BottleneckMLP recovers this loss and exceeds the baseline. Classifier accuracy is not reported on original TGIB paper, we include it in the Appendix H.

5.2.2 Explanatory Subgraph Quality

Table 3: PGIB Fidelity. BottleneckMLP improves explanation quality (Fid^+, Fid^-) across datasets

Method	**MUTAG**		**BA-2Motifs**		**NCI1**		**PROTEINS**	
	Fid+↑	Fid-↓	Fid+↑	Fid-↓	Fid+↑	Fid-↓	Fid+↑	Fid-↓
PGIB	0.750 ± 0.079	0.588 ± 0.204	0.825 ± 0.159	0.492 ± 0.149	0.451 ± 0.124	0.523 ± 0.156	0.639 ± 0.024	0.602 ± 0.125
PGIB w/o IB Loss	0.719 ± 0.083	0.516 ± 0.192	0.829 ± 0.160	0.479 ± 0.143	0.524 ± 0.136	0.546 ± 0.125	0.654 ± 0.028	0.604 ± 0.035
PGIB + BottleneckMLP	**0.762** ± 0.071	**0.383** ± 0.254	**0.975** ± 0.079	**0.400** ± 0.242	**0.771** ± 0.162	**0.478** ± 0.214	**0.658** ± 0.018	**0.592** ± 0.087

Table 3 reports fidelity metrics for PGIB on MUTAG and BA-2Motifs (the only datasets with ground-truth explanations). Adding BottleneckMLP consistently improves Fid^+ and reduces Fid^-, yielding higher-quality explanations. Table 2 reports improved AUC-ROC scores across datasets for TGIB with BottleneckMLP. AUC-ROC scores and information planes for GSAT variants are provided in Appendix F. Visual comparisons of explanatory subgraphs appear in Appendix D. We further extend BottleneckMLP to node classification Luo et al. (2020b) and subgraph recognition Yu et al. (2022), where results in Appendix H confirm its generality and effectiveness.

6 Discussion and Future Directions

BottleneckMLP is an architectural module that implicitly enforces the IB principle without IB loss terms. Our model-agnostic method relies on two components: Importance-Weighted Gaussianization and Progressive Compression via MLP layers. We both theoretically and empirically demonstrate the effectiveness of these components, and show that explicit IB losses are less effective.

Formalizing the conditions under which implicit bottlenecks arise remains an open theoretical challenge. A deeper information-theoretic analysis of graph message passing could further unify our observations with learning theory. We unite the representation geometry of explainers with IB theory; future work can study how representation dynamics impact generalization in graph learning. Future work may explore how inducing disentanglement Pan et al. (2020) may amplify IB effects and promote stable, explainable representations in GNNs.

7 Reproducibility Statement

All datasets used are available for download from the original source, linked in Appendix C. All code needed to reproduce experiments and generate all figures within the paper is available in the supplementary material, and experimental setup information is also in Appendix C. Follow the guidelines on the supplementary material README pages to reproduce the performance results, as well as the plots presented throughout the paper.

References

Devanshu Agrawal, Theodore Papamarkou, and Jacob Hinkle. Wide neural networks with bottlenecks are deep gaussian processes, 2020. URL https://arxiv.org/abs/2001.00921.

Davide Bacciu, Federico Errica, and Alessio Micheli. Contextual graph markov model: A deep and generative approach to graph processing, 2019. URL https://arxiv.org/abs/1805.10636.

Mohit Bajaj, Lingyang Chu, Zi Yu Xue, Jian Pei, Lanjun Wang, Peter Cho-Ho Lam, and Yong Zhang. Robust counterfactual explanations on graph neural networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 5644–5655. Curran Associates, Inc., 2021a. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/2c8c3a57383c63caef6724343eb62257-Paper.pdf.

Mohit Bajaj, Lingyang Chu, Zi Yu Xue, Jian Pei, Lanjun Wang, Peter Cho-Ho Lam, and Yong Zhang. Robust counterfactual explanations on graph neural networks. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021b. URL https://openreview.net/forum?id=Uq_tGs7N54M.

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

Feng Bao. Disentangled variational information bottleneck for multiview representation learning. *arXiv preprint arXiv:2105.07599*, 2021. URL https://arxiv.org/abs/2105.07599.

S.H. Shabbeer Basha, Shiv Ram Dubey, Viswanath Pulabaigari, and Snehasis Mukherjee. Impact of fully connected layers on performance of convolutional neural networks for image classification. *Neurocomputing*, 378:112–119, 2020. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2019.10.008. URL https://www.sciencedirect.com/science/article/pii/S0925231219313803.

Tian Bian, Xi Xiao, Tingyang Xu, Peilin Zhao, Wenbing Huang, Yu Rong, and Junzhou Huang. Rumor detection on social media with bi-directional graph convolutional networks, 2020. URL https://arxiv.org/abs/2001.06362.

Leo Breiman. The individual ergodic theorem of information theory. *Annals of Mathematical Statistics*, 28(3):809–811, 1957. doi: 10.1214/aoms/1177707032.

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, 2024. URL https://arxiv.org/abs/2402.02036.

Ziheng Chen, Fabrizio Silvestri, Jia Wang, Yongfeng Zhang, Zhenhua Huang, Hongshik Ahn, and Gabriele Tolomei. Grease: Generate factual and counterfactual explanations for gnn-based recommendations, 2022. URL https://arxiv.org/abs/2208.04222.

Bin Dai, Chen Zhu, and David Wipf. Compressing neural networks using the variational information bottleneck. *arXiv preprint arXiv:1802.10399*, 2018. URL https://arxiv.org/abs/1802.10399.

Enyan Dai, Tianxiang Zhao, Huaisheng Zhu, Junjie Xu, Zhimeng Guo, Hui Liu, Jiliang Tang, and Suhang Wang. A comprehensive survey on trustworthy graph neural networks: Privacy, robustness, fairness, and explainability. *Machine Intelligence Research*, 21(6):1011–1061, September 2024. ISSN 2731-5398. doi: 10.1007/s11633-024-1510-8. URL http://dx.doi.org/10.1007/s11633-024-1510-8.

Asim Kumar Debnath, Rosa L. Lopez de Compadre, Gargi Debnath, Alan J. Shusterman, and Corwin Hansch. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of Medicinal Chemistry*, 34(2):786–797, 1991. doi: 10.1021/jm00106a046. URL https://doi.org/10.1021/jm00106a046.

Yingjun Du, Jun Xu, Huan Xiong, Qiang Qiu, Xiantong Zhen, Cees G. M. Snoek, and Ling Shao. Learning to learn with variational information bottleneck for domain generalization. *arXiv preprint arXiv:2007.07645*, 2020. URL https://arxiv.org/abs/2007.07645.

Danish Ebadulla, Aditya Gulati, and Ambuj Singh. Normalized space alignment: A versatile metric for representation analysis. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, pp. 555–566, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400714542. doi: 10.1145/3711896.3737065. URL https://doi.org/10.1145/3711896.3737065.

Daniel Eftekhari and Vardan Papyan. On the importance of gaussianizing representations, 2025. URL https://arxiv.org/abs/2505.00685.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International Conference on Algorithmic Learning Theory*, pp. 63–77. Springer, 2005.

Shenyang Huang, Yasmeen Hitti, Guillaume Rabusseau, and Reihaneh Rabbany. Laplacian change point detection for dynamic graphs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '20. ACM, August 2020. doi: 10.1145/3394486.3403077. URL http://dx.doi.org/10.1145/3394486.3403077.

Jaykumar Kakkad, Jaspal Jannu, Kartik Sharma, Charu Aggarwal, and Sourav Medya. A survey on explainability of graph neural networks, 2023. URL https://arxiv.org/abs/2306.01958.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017. URL https://arxiv.org/abs/1609.02907.

Peter Kocsis, Peter Súkeník, Guillem Brasó, Matthias Nießner, Laura Leal-Taixé, and Ismail Elezi. The unreasonable effectiveness of fully-connected layers for low-data regimes, 2022. URL https://arxiv.org/abs/2210.05657.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '19, pp. 1269–1278. ACM, July 2019. doi: 10.1145/3292500.3330895. URL http://dx.doi.org/10.1145/3292500.3330895.

Namkyeong Lee, Dongmin Hyun, Gyoung S. Na, Sungwon Kim, Junseok Lee, and Chanyoung Park. Conditional graph information bottleneck for molecular relational learning, 2023. URL https://arxiv.org/abs/2305.01520.

Honglin Li, Chenglu Zhu, Yunlong Zhang, Yuxuan Sun, Zhongyi Shui, Wenwei Kuang, Sunyi Zheng, and Lin Yang. Task-specific fine-tuning via variational information bottleneck for weakly-supervised pathology whole slide image classification. In *Proceedings*

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7454–7463, 2023a. URL https://openaccess.thecvf.com/content/CVPR2023/papers/Li_Task-Specific_Fine-Tuning_via_Variational_Information_Bottleneck_for_Weakly-Supervised_Pathology_Whole_CVPR_2023_paper.pdf.

Yiqiao Li, Jianlong Zhou, Sunny Verma, and Fang Chen. A survey of explainable graph neural networks: Taxonomy and evaluation metrics, 2023b. URL https://arxiv.org/abs/2207.12599.

Yunchao (Lance) Liu, Yu Wang, Oanh Vu, Rocco Moretti, Bobby Bodenheimer, Jens Meiler, and Tyler Derr. Interpretable chirality-aware graph neural network for quantitative structure activity relationship modeling in drug discovery. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(12):14356–14364, Jun. 2023. doi: 10.1609/aaai.v37i12.26679. URL https://ojs.aaai.org/index.php/AAAI/article/view/26679.

Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 19620–19631. Curran Associates, Inc., 2020a. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/e37b08dd3015330dcbb5d6663667b8b8-Paper.pdf.

Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network, 2020b. URL https://arxiv.org/abs/2011.04573.

Kha-Dinh Luong, Mert Kosan, Arlei Lopes Da Silva, and Ambuj Singh. Robust ante-hoc graph explainer using bilevel optimization, 2024. URL https://arxiv.org/abs/2305.15745.

Siqi Miao, Miaoyuan Liu, and Pan Li. Interpretable and generalizable graph learning via stochastic attention mechanism, 2022. URL https://arxiv.org/abs/2201.12987.

Siqi Miao, Yunan Luo, Mia Liu, and Pan Li. Interpretable geometric deep learning via learnable randomness injection, 2023. URL https://arxiv.org/abs/2210.16966.

Ari Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/a7a3d70c6d17a73140918996d03c014f-Paper.pdf.

Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020. URL www.graphlearning.io.

Ziqi Pan, Li Niu, Jianfu Zhang, and Liqing Zhang. Disentangled information bottleneck, 2020. URL https://arxiv.org/abs/2012.07372.

Ben Poole, Sherjil Ozair, Aaron van den Oord, Alexander A. Alemi, and George Tucker. On variational bounds of mutual information, 2019. URL https://arxiv.org/abs/1905.06922.

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 (CVPR)*, June 2019.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability, 2017. URL https://arxiv.org/abs/1706.05806.

Evgenia Rusak, Patrik Reizinger, Attila Juhos, Oliver Bringmann, Roland S. Zimmermann, and Wieland Brendel. Infonce: Identifying the gap between theory and practice, 2025. URL https://arxiv.org/abs/2407.00143.

Leonardo F.S. Scabini and Odemir M. Bruno. Structure and performance of fully connected neural networks: Emerging complex network properties. *Physica A: Statistical Mechanics and its Applications*, 615:128585, 2023. ISSN 0378-4371. doi: https://doi.org/10.1016/j.physa.2023.128585. URL https://www.sciencedirect.com/science/article/pii/S0378437123001401.

Sangwoo Seo, Sungwon Kim, Jihyeong Jung, Yoonho Lee, and Chanyoung Park. Self-explainable temporal graph networks based on graph information bottleneck, 2024a. URL https://arxiv.org/abs/2406.13214.

Sangwoo Seo, Sungwon Kim, and Chanyoung Park. Interpretable prototype-based graph information bottleneck, 2024b. URL https://arxiv.org/abs/2310.19906.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information, 2017. URL https://arxiv.org/abs/1703.00810.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle, 2015. URL https://arxiv.org/abs/1503.02406.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method, 2000. URL https://arxiv.org/abs/physics/0004057.

Tailin Wu, Hongyu Ren, Pan Li, and Jure Leskovec. Graph information bottleneck, 2020. URL https://arxiv.org/abs/2010.12811.

Junchi Yu, Jie Cao, and Ran He. Improving subgraph recognition with variational graph information bottleneck, 2022. URL https://arxiv.org/abs/2112.09899.

Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. Explainability in graph neural networks: A taxonomic survey, 2022. URL https://arxiv.org/abs/2012.15445.

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, 2023. doi: 10.1109/TPAMI.2022.3204236.

Jiaxing Zhang, Zhuomin Chen, Hao Mei, Longchao Da, Dongsheng Luo, and Hua Wei. Reg-explainer: Generating explanations for graph neural networks in regression tasks, 2024. URL https://arxiv.org/abs/2307.07840.

Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks, 2018. URL https://arxiv.org/abs/1802.09691.

Marinka Zitnik, Monica Agrawal, and Jure Leskovec. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466, 06 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty294. URL https://doi.org/10.1093/bioinformatics/bty294.

APPENDIX: TECHNICAL DETAILS AND SUPPLEMENTARY MATERIAL

TABLE OF CONTENTS

A LIST OF SYMBOLS

B PROOFS OF LEMMA 1 AND LEMMA 2

Proof of Lemma 1 *$J(\cdot)$ is positive semi-definite, therefore trace is nonnegative. Thus $H(Z_i \mid X)$ is monotonically increasing in the noise variance t_i. The same argument applies jointly to the vector Z_{unimp}, since independent Gaussian noise is injected into each unimportant node.*

Proof of Lemma 2 *By the law of total covariance,*

$$\text{Cov}(Z_i) = \mathbb{E}[\text{Cov}(Z_i \mid X)] + \text{Cov}(\mathbb{E}[Z_i \mid X]) = t_i I_d + \text{Cov}(f_i(X)).$$

The Gaussian distribution maximizes entropy among all distributions with fixed covariance, Σ_{Z_i}, which yields the bound.

C EXPERIMENTAL SETUP

All experiments were implemented using `PyTorch Geometric` and run on either CPU or NVIDIA H200 GPUs. Unless otherwise stated, the following hyperparameters were used:

Tables 6 and Table 7 give an overview of the dataset statistics used within the paper.

Symbol	Description
G	Input graph
G_S	Explanation subgraph
X	Input feature vector (node embeddings of G after graph layers)
Z	Learned hidden representation
Z_{imp}	Learned hidden representations for important nodes
Z_{unimp}	Learned hidden representations for unimportant nodes
Z_i	Noisy hidden representation for node i
Y	Ground truth label
$H(X)$	Entropy of r.v. X
$I(X;Z)$	Mutual information between X and Z
β	Lagrangian multiplier for the IB functional
$g_\phi(G)$	Subgraph extractor
$P_\phi(G_S\|G)$	Distribution of subgraph G_S outputted by $g_\phi(G)$
$\mathcal{N}(x_i)$	Neighborhood of x_i
f_i	Encoder yielding node/ edge embeddings
t_i	Noise magnitude, inversely proportional to importance
ϵ_i	Gaussian noise vector
I_d	d $\times$ d identity matrix
α_i	Importance score of node i
σ	Fixed noise scale; hyperparameter
$J(\cdot)$	Fischer information matrix
$Tr(\cdot)$	Trace of a matrix
det	Matrix determiant
Cov	Covariance
d	Dimension of embedding space
G_p	Prototype graph in PGIB
$Q(G_S)$	Assumed prior for G_S for the KL variational upper bound
$p(x_1, x_2, ..., x_n)$	Joint distribution of random variables
$\mathcal{N}(x_i)$	The local neighborhood on which r.v. x_i depends
ΔI_X^k	Partial compression at layer k
h_v^k	Embedding of node v in kth layer
'IB Loss'	Loss term explicitly which explicitly enforces IB principle

Table 4: Source code links of baseline models

Method	Source code
GSATMiao et al. (2022)	https://github.com/Graph-COM/GSAT/tree/main
PGIBSeo et al. (2024b)	https://github.com/sang-woo-seo/PGIB
TGIBSeo et al. (2024a)	https://github.com/sang-woo-seo/TGIB

Table 5: Summary of model hyperparameters.

Model	Learning Rate	Weight Decay	Batch Size	# Epochs	# Random Seeds
GSATMiao et al. (2022)	10^{-3}	0.0	128	100	10
PGIBSeo et al. (2024b)	5^{-3}	0.0	128	300	10
TGIBSeo et al. (2024a)	10^{-5}	0.0	200	10	10

Table 6: Overview of graph classification datasets used in experiments on GSAT and PGIB.

Dataset	#Graphs	#Classes	Avg. # Nodes	Avg. # Edges	Node Labels	Edge Labels	Node Attr. (Dim.)	Edge Attr. (Dim.)
MUTAG Debnath et al. (1991)	188	2	17.93	19.79	+	+	-	-
NCI1 Morris et al. (2020)	4110	2	29.87	32.30	+	-	-	-
PROTEINS Morris et al. (2020)	1113	2	39.06	72.82	+	-	+	-
BA_2Motifs Luo et al. (2020b)	1000	2	25	51.39	-	-	-	-

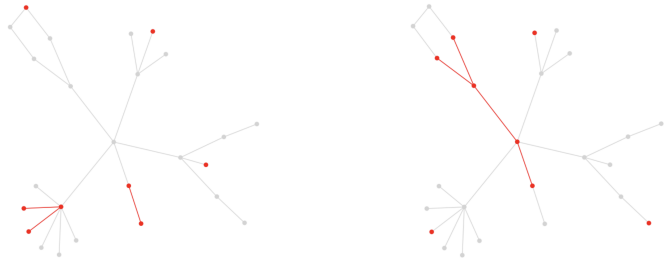
Table 7: Overview of node classification datasets used in experiments on TGIB.

Dataset	Domain	#Nodes	#Edges	#Edge Features	Duration
WikipediaKumar et al. (2019)	Social	9,227	157,474	172	1 month
CanParlHuang et al. (2020)	Politics	734	74,478	1	14 years
USLegisHuang et al. (2020)	Politics	225	60,396	1	12 terms

D EXPLANATORY SUBGRAPH VISUALIZATION

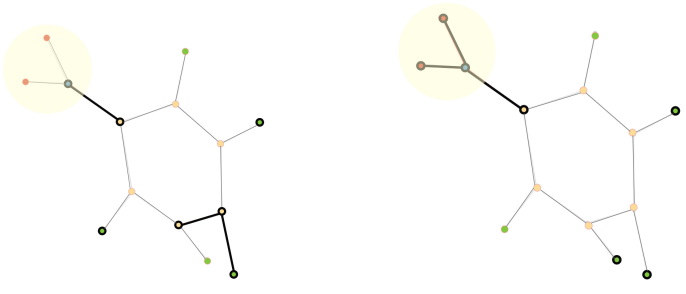
(a) GSAT (b) GSAT + BottleneckMLP

Figure 6: Visualization of explanatory subgraphs for MUTAG dataset. (a) GSAT baseline, and (b) GSAT enhanced with BottleneckMLP. BottleneckMLP correctly identifies the carbon ring for non-mutagenic molecules.



(a) PGIB (b) PGIB + BottleneckMLP

Figure 7: Visualization of explanatory subgraphs for BA-2Motifs dataset. (a) PGIB baseline, and (b) PGIB enhanced with BottleneckMLP. BottleneckMLP correctly identifies the cycle motif.



(a) PGIB (b) PGIB + BottleneckMLP

Figure 8: Visualization of explanatory subgraphs for MUTAG dataset. (a) PGIB baseline, and (b) PGIB enhanced with BottleneckMLP. BottleneckMLP successfully identifies the NO2 group (yellow shaded region), which is the ground truth explanation for mutagenic molecules in MUTAG.

E Representation Dynamics Across Layers

Supplementary to the NSA and node drift results in Section 5.1, in this section we include these results for additional models, and we study the representation dynamics of the important versus unimportant node embeddings across training epochs from several lenses including (1) node drift, (2) convex hull volume, (3) average linkage distance. Each of these further corroborates the effectiveness of BottleneckMLP in implicitly enforcing IB as an architectural primitive which effects the representation dynamics over training.

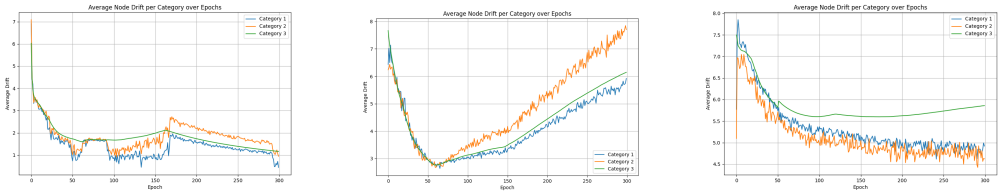
E.1 Node Drift

Architecture / Configuration	MUTAG	BA-2Motifs	NCI1	PROTEINS
GSAT	-0.02 ± 0.007	0.006 ± 0.006	-0.004 ± 0.006	0.002 ± 0.001
GSAT w/o IB Loss	-0.018 ± 0.007	-0.003 ± 0.006	0.028 ± 0.011	$\mathbf{0.004 \pm 0.013}$
GSAT w/o IB Loss + BottleneckMLP	$\mathbf{0.07 \pm 0.009}$	$\mathbf{0.02 \pm 0.006}$	$\mathbf{0.04 \pm 0.054}$	-0.003 ± 0.015

Table 8: Difference between unimportant and important node drift (Unimp − Imp) for GSAT. With BottleneckMLP, we validate across datasets that important nodes stabilize and unimportant nodes drift to higher-entropy representations.

Architecture / Configuration	MUTAG	BA-2Motifs	NCI1	PROTEINS
PGIB	0.087 ± 1.014	0.126 ± 0.240	0.227 ± 0.127	0.129 ± 0.0531
PGIB w/o IB Loss	-0.205 ± 0.169	0.186 ± 0.457	-0.116 ± 0.128	0.294 ± 0.095
PGIB w/o IB Loss + BottleneckMLP	$\mathbf{0.546 \pm 0.111}$	$\mathbf{0.214 \pm 0.049}$	$\mathbf{0.415 \pm 0.272}$	$\mathbf{0.937 \pm 0.563}$

Table 9: Difference between unimportant and important node drift (Unimp − Imp) for PGIB. With BottleneckMLP, we validate across datasets that important nodes stabilize and unimportant nodes drift to higher-entropy representations.



(a) PGIB (b) PGIB w/o IB Loss (c) PGIB + BottleneckMLP

Figure 9: Drift of important vs. unimportant nodes across models: (a) PGIB baseline, (b) PGIB without Information Loss, and (c) PGIB with BottleneckMLP. The blue line represents the average drift over epochs of the category 1 nodes, the orange line is that of category 2 nodes, and the green is that of category 3 nodes. We see that BottleneckMLP effectively enforces drift/ forgetting of unimportant nodes, as evident by their increased drift in latent space across epochs.

E.2 Convex Hull

To further support our hypothesis that the BottleneckMLP influences representation structure locally—particularly around important nodes—we analyze the convex hull volumes and average linkage distances of node embeddings over training.

As described earlier, we project node embeddings into 3D using PCA and compute the convex hull volume for each category across training epochs. Our findings, visualized in Figure 10, indicate a clear trend: Category 3 nodes (unimportant) occupy significantly larger and more variable convex hulls compared to Category 1 nodes (important). This increasing spatial dispersion suggests that unimportant nodes drift more in embedding space as training progresses, while important nodes remain compact and tightly clustered—consistent with more stable local structure.

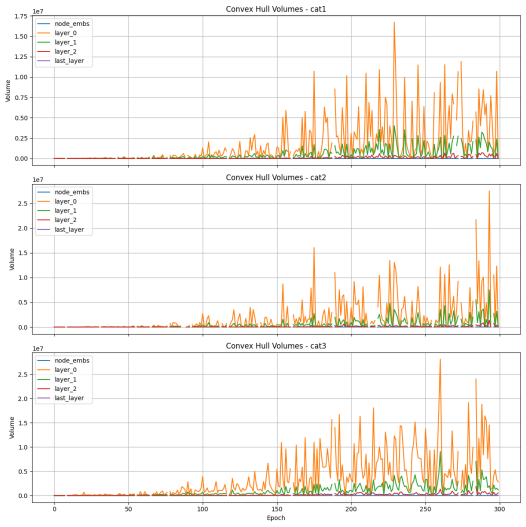
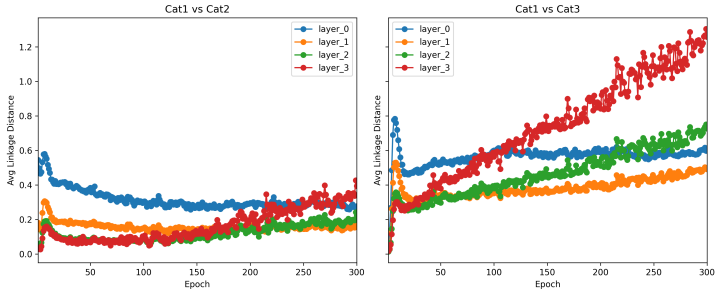
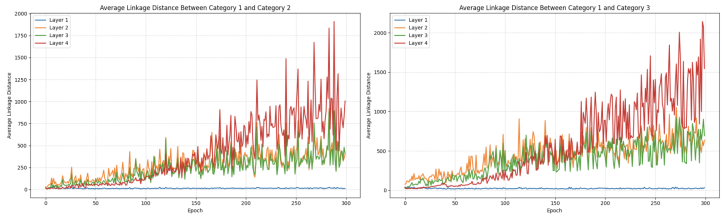


Figure 10: Convex hull volumes increase across categories.



(a) GSAT node linkage visualization



(b) PGIB node linkage visualization

Figure 11: Comparison of distances from Category 1 nodes to Category 3 nodes. Higher drift in the embedding space of unimportant nodes increases these distances. We also present the importance of node dependencies in graphs, Category 3 nodes drift far more than Category 2 nodes, even though they are both unimportant

E.3 Node linkage distance

To complement this analysis, we compute the average linkage distance between node embeddings within each category. This metric quantifies the average pairwise distance between points in a cluster, offering an alternative view of intra- and inter-category embedding dynamics. The results, summarized in Table 11, closely mirror the trends observed in the convex hull and LNSA analyses. Category 1 nodes consistently exhibit low average intra-category linkage distances, confirming strong internal cohesion. Category 3 nodes show a significant increase in linkage distance, especially at deeper layers and later epochs, reinforcing the notion of representational drift in less important regions. Category 2 nodes, which represent the neighbors of important nodes, exhibit average linkage values that lie between those of Categories 1 and 3, but are closer to Category 1. This suggests

that these nodes remain structurally and representationally aligned with important nodes due to their direct connections and dependency.

We also report the average inter-category linkage distances in Table 10. Notably, the largest inter-cluster distances are observed between Category 1 and Category 3, and between Category 2 and Category 3. In contrast, the average distance between Category 1 and Category 2 embeddings remains relatively low, highlighting their continued proximity in latent space. These patterns further confirm the embedding drift of unimportant nodes away from critical substructures.

Table 10: Inter-Cluster Average Linkage for Categories and Layers

Layer	Cat1 - Cat2	Cat1 - Cat3	Cat2 - Cat3
Layer 1	14.955	22.934	23.155
Layer 2	285.214	450.272	451.345
Layer 3	236.287	383.437	383.836
Layer 4	409.867	568.589	568.324

Table 11: Intra-Cluster Average Linkage for Categories and Layers

Layer	Cat1	Cat2	Cat3
Layer 1	12.734	14.277	23.000
Layer 2	234.417	256.448	450.396
Layer 3	197.862	215.450	383.092
Layer 4	351.402	387.511	563.660

The results reported in Table 11 for the average linkage distance between a Category 1 node embedding, and the closest node embedding from Category 2, and Category 3, respectively, are visualized in Figure 11.

Together, these metrics validate our claim that important nodes form stable neighborhoods in embedding space, while unimportant nodes undergo more drift. Importantly, this localized drift is not easily observable through global metrics, but becomes clearly evident through localized geometric and clustering analyses. Overall, this set of geometric and clustering-based analyses underscores the model's localized influence on representation learning, and validates our claim: the BottleneckMLP selectively shapes local representations, stabilizing meaningful substructures while allowing greater flexibility and dispersion in the remainder of the graph.

E.4 HSIC Results

Hilbert-Schmidt Independence Criterion (HSIC) is a kernel-based method for measuring statistical dependence between variables Gretton et al. (2005). We use HSIC to quantify the dependence between the output of GCN layers (X) and the final node embeddings before classification (Z), as an alternative to mutual information. Lower HSIC values indicate greater independence, helping us assess how much information Z retains from X.

We observe that lowest values of dependence between input graph G and learned latent representation of the explanation G_S are achieved with BottleneckMLP. (Table 12)

Table 12: HSIC values across datasets for GSAT. Lower values indicate more independence

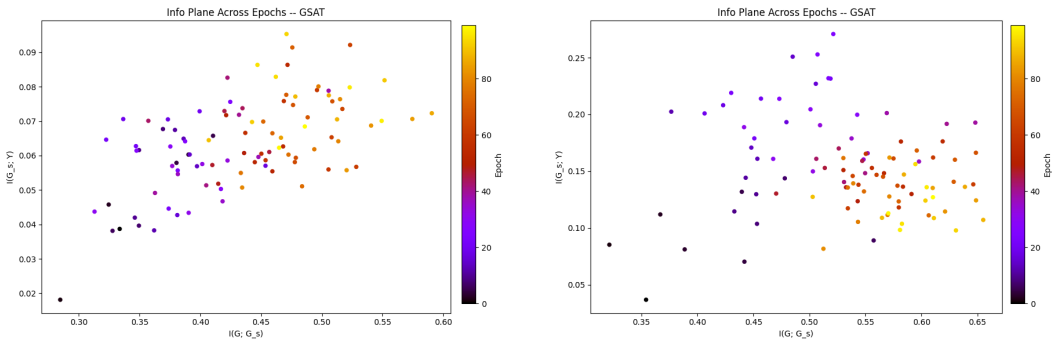
Architecture / Configuration	**BA-2Motifs**	**MUTAG**	**PROTEINS**	**NCI1**
GSAT	3.39×10^{-3}	6.23×10^{-3}	2.95×10^{-3}	6.98×10^{-3}
GSAT w/o IB Loss	$\mathbf{1.11 \times 10^{-3}}$	1.01×10^{-2}	2.89×10^{-3}	6.87×10^{-3}
GSAT w/o IB Loss + BottleneckMLP	1.48×10^{-3}	$\mathbf{3.89 \times 10^{-3}}$	$\mathbf{2.53 \times 10^{-3}}$	$\mathbf{6.37 \times 10^{-3}}$

Table 13: GSAT AUC-ROC

	MUTAG	BA-2Motifs
GSAT	0.995 ±0.03	0.988±0.01
GSAT w/o IB Loss	0.998±0.04	0.996±0.00
GSAT w/o IB Loss + Bottleneck MLP	0.987±0.01	0.982±0.01

F AUC-ROC

Table 13 shows that explanation AUC-ROC remains consistently high across all GSAT variants, even when the information loss term is removed. The differences are not statistically significant, suggesting that explicit information loss is not essential for generating high-quality subgraphs. Notably, our BottleneckMLP matches this performance, preserving explanatory power without additional loss terms. Figure 12 further reinforces this conclusion. In the information plane, GSAT with BottleneckMLP (Figure 12b) achieves higher mutual information between the explanatory subgraph and the label, $I(G_s, y)$, without increasing $I(G, G_s)$. This demonstrates that our approach improves both label relevance and compression.



(a) GSAT

(b) GSAT w/o IB Loss +BottleneckMLP

Figure 12: Comparison of $I(G, G_s)$ vs. $I(G_s, Y)$ across training epochs for the original GSAT model, and our model with a BottleneckMLP with fully connected layers. BottleneckMLP elicits both decreased $I(X; Z)$ and increased $I(Z; Y)$, as compared to original GSAT.

G Testing BottleneckMLP on a Variety of Architectures

We tested multiple fully-connected architectures inserted after the GNN layers of the existing explainer models. All BottleneckMLP architectures use ReLU activations. Determining the optimal architecture of the BottleneckMLP for a given dataset/ explainer model is analogous to hyperparameter tuning.

Table 14: PGIB/ GSAT Test Accuracy

	MUTAG	BA-2Motifs	NCI1	PROTEINS
GSAT w/o IB Loss + Bottleneck MLP (64-64-64)	0.943 ± 0.019	0.959 ± 0.066	0.799 ± 0.018	0.750 ± 0.048
GSAT w/o IB Loss + Bottleneck MLP (64-48-32)	0.959±0.02	0.995 ±0.18	0.798±0.01	0.749 ±0.058
GSAT w/o IB Loss + Bottleneck MLP (64-32)	0.942 ± 0.010	0.930 ± 0.131	0.797 ± 0.018	0.729 ± 0.056
GSAT w/o IB Loss + Bottleneck MLP (64-32-16)	0.942 ± 0.013	0.854 ± 0.162	0.796 ± 0.019	0.747 ± 0.043
GSAT w/o IB Loss + Bottleneck MLP (64-128)	0.926 ± 0.037	0.904 ± 0.152	0.788 ± 0.025	0.732 ± 0.067
PGIB w/o IB Loss + Bottleneck MLP (128-128-128)	0.928 ± 0.011	0.967 ± 0.009	0.771 ± 0.009	0.769 ± 0.015
PGIB w/o IB Loss + Bottleneck MLP (128-96-72)	0.918 ± 0.016	0.947 ± 0.033	0.769 ± 0.010	0.784 ± 0.015
PGIB w/o IB Loss + Bottleneck MLP (128-64)	0.923 ± 0.026	0.939 ± 0.055	0.763 ± 0.017	0.0766 ± 0.036
PGIB w/o IB Loss + Bottleneck MLP (128-64-32)	0.923 ± 0.014	0.940 ± 0.040	0.766 ± 0.010	0.782 ± 0.014
PGIB w/o IB Loss + Bottleneck MLP (128-256)	0.919 ± 0.019	0.959 ± 0.028	0.753 ± 0.016	0.779 ± 0.018

H GENERALIZATION OF BOTTLENECKMLP ACROSS GRAPH TASKS

We used all existing ante-hoc graph explainers that use variational IB objectives as our baselines. For generalizability, we tested BottleneckMLP on subgraph recognition Yu et al. (2022) and post-hoc node classification. For VGIB Yu et al. (2022), we removed the MI loss penalty term, and added BottleneckMLP. For VGIB in Table 15, the default model hidden dimension $h = 16$.

Table 15: VGIB Test Prediction Accuracy ± Std Dev.

Model Variant	**MUTAG**	**PROTEINS**	**AIDS**	**NCI1**
VGIB (normal)	0.423 ± 0.147	0.573 ± 0.036	0.418 ± 0.153	0.496 ± 0.069
VGIB (noinfo)	0.573 ± 0.167	0.569 ± 0.031	0.335 ± 0.207	0.494 ± 0.043
VGIB + BottleneckMLP ($h - \frac{h}{4} - h$)	**0.607** ± 0.232	0.556 ± 0.075	0.530 ± 0.345	$\underline{0.500} \pm 0.062$
VGIB + BottleneckMLP (128-32)	0.556 ± 0.231	**0.593** ± 0.002	**0.679** ± 0.313	0.474 ± 0.019
VGIB + BottleneckMLP (64-48-32)	$\underline{0.594} \pm 0.233$	$\underline{0.592} \pm 0.002$	$\underline{0.576} \pm 0.238$	**0.561** ± 0.062

For PGExplainer Luo et al. (2020b), we removed size and entropy losses, kept only cross-entropy, and added BottleneckMLP to the node classifier for implicit compression. Table 16 shows that we achieve better generalization and accuracy in node classifiaction when BottleneckMLP is added.

	BAShapes	**TreeCycles**	**TreeGrids**
GCN	$\underline{0.954} \pm 0.009$	0.916 ± 0.037	$\underline{0.806} \pm 0.075$
GCN + BottleneckMLP (48-30)	**0.981** ± 0.013	$\underline{0.959} \pm 0.013$	**0.815** ± 0.111
GCN + BottleneckMLP (30)	0.979 ± 0.013	**0.961** ± 0.008	0.782 ± 0.162

Table 16: GNN Node Classification Accuracy

Table 16 shows that without the entropy loss, PGExplainer suffers a tremendous drop in explanation quality (AUC-ROC). However, BottleneckMLP acts as such entropy regularizer, and the AUC-ROC goes right back up, reaching or exceeding the initial PGExplainer performance.

	BA-Shapes	**Tree-Cycles**	**Tree-Grids**
PGExplainer	$\underline{0.993} \pm 0.006$	**0.941** ± 0.002	$\underline{0.676} \pm 0.003$
PGExplainer w/o Entropy Loss	0.033 ± 0.021	0.058 ± 0.002	0.628 ± 0.028
PGExplainer w/o E. Loss + BottleneckMLP	**0.999** ± 0.0001	$\underline{0.938} \pm 0.031$	**0.732** ± 0.001

Table 17: PGExplainer Explanation AUC-ROC

	Classifier Accuracy		
Model	Wikipedia	CanParl	USLegis
TGIB	0.947	0.528	**0.642**
TGIB w/o Info Loss	**0.960**	**0.588**	0.544
TGIB w/o Info Loss + Bottleneck MLP	$\underline{0.959}$	$\underline{0.586}$	$\underline{0.607}$

Table 18: TGIB Classifier Accuracy.

I EXPLICIT IB METHODS

We demonstrate ineffectiveness across multiple methods:

GSAT Miao et al. (2022) adopts a variational IB framework to extract explanatory subgraphs. It introduces a lower bound on $I(G_S; Y)$ by employing a variational approximation to the joint distribution $P(Y, G_S)$, and an upper bound on $I(G; G_S)$ (what we refer to as info-loss) using a variational approximation to the marginal $P(G_S) = \sum_G P_\phi(G_S|G)P_G(G)$.

The resulting KL-divergence term between $P_\phi(G_S|G)$ and $Q(G_S)$ simplifies to a sum of KL divergences between individual Bernoulli distributions per edge. While this formulation is differentiable, it is not tight. The prior $Q(G_S)$ acts only as a weak regularizer since it assumes i.i.d. edge inclusion and ignores structural dependencies. Consequently, the KL term provides an upper bound on $I(G; G_S)$ which is unable to sufficiently constrain the learned explainer.

PGIB Seo et al. (2024b) introduces a prototype-based framework that explains GNN predictions by selecting a subgraph G_s and a prototype G_p that together retain information about the label Y. It maximizes a lower bound on $I(Y; G_s, G_p)$ via a variational classifier $q_\theta(Y|\gamma(G_s, G_p))$ where γ is a similarity function. To minimize $I(G_s; G_p)$ (what we refer to as info-loss), PGIB uses a variational upper bound $\mathbb{E}[-\log q_\phi(G_p|G_s)]$ similar to GSAT, or the variant of their method, $\text{PGIB}_{\text{CONT}}$, leverages a contrastive loss approach proposed in Rusak et al. (2025) to minimize $I(X, Z)$, see Appendix M.

Despite being grounded in the IB framework, these approximations are weak. The variational classifier q_θ is task-specific and does not tightly control information flow similar to GSAT. The contrastive loss in $\text{PGIB}_{\text{CONT}}$ is indirect, sensitive to sampling and hyperparameters, and lacks clear control over mutual information. As such, $\text{PGIB}_{\text{CONT}}$ does not impose a tight constraint on $I(G; G_S)$, as it does not explicitly regularize the information retained from the original graph G and enforce compression.

TGIB integrates temporal graph learning with the IB principle to improve both link prediction and model explainability. It extracts a bottleneck code R_k for each target edge e_k from its L-hop neighborhood G_k. This code is a subgraph of the L-hop computation graph around e_k and acts as a compressed representation used for prediction. By limiting the information flow through this bottleneck, the model highlights the relevant parts of the neighborhood for edge prediction.

The objective function is defined as: $\min_{R_k} -I(Y_k; R_k) + \beta I(R_k; e_k, G_k)$ where the first term encourages the model to preserve information relevant to the label Y_k, while the second term penalizes encoding unnecessary information from the edge e_k and its surrounding graph G_k. We find that the IB Loss is detrimental to the performance of TGIB.

J Fidelity Metric

Following Seo et al. (2024b), we use Fidelity metrics Pope et al. (2019); Yuan et al. (2023) to quantify the quality of the explanations. Let y_i denote the ground-truth values and $\hat{y}_i$ denote the predicted values for the i-th input graph. Let k be the sparsity score, denoting the $(k \times 100)\%$ of important nodes of the original graph which are used to construct the explanatory subgraph. The prediction of the explanatory subgraph for a sparsity score of k is denoted by $\hat{y}_i^k$. The prediction of the non-explanatory subgraph for a sparsity score of k is denoted by $\hat{y}^{1-k_i}$. The equations for the Fidelity metrics are given by:

$$F^- = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(y_i = \hat{y}_i) - \mathbb{I}(y_i = \hat{y}_i^k), F^+ = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(y_i = \hat{y}_i) - \mathbb{I}(y_i = \hat{y}_i^{1-k}), \tag{12}$$

where the binary indicator $\mathbb{I}(y_i = \hat{y}_i)$ returns 1 if $y_i = \hat{y}_i$, and 0, otherwise. Higher values for F^+, and lower values for F^- indicate that the explanatory subgraphs produced by the model are better.

K Normalized Space alignment (NSA)

To evaluate the similarity between learned representations in neural networks, Ebadulla et al. propose Normalized Space Alignment (NSA) as a manifold analysis technique for neural network representations which provides a robust similarity metric, and loss function, for comparing vector spaces across architectures, layers, or training regimes Ebadulla et al. (2025). NSA builds upon previous methods including Canonical Correlation Analysis (CCA) Morcos et al. (2018); Raghu et al. (2017) and Centered Kernel Alignment (CKA) Kornblith et al. (2019), but addresses key limitations such as scale sensitivity and confounding dimensionality effects.

The central idea is to treat representations as subspaces, and then measure alignment via projections onto the normalized Grassmann manifold, where vector directions are invariant to orthogonal transformations and global scaling. Ebadulla et al. introduce Global NSA (GNSA), which compares the entire representation spaces holistically, based on normalized projection matrices and Frobenius norms, and Local NSA (LNSA), which focuses on fine-grained, per-sample neighborhood structure preservation via rank-correlated similarity matrices, emphasizing local geometric alignment. NSA provides a scale- and dimension-agnostic framework to evaluate whether our BottleneckMLP layers yield more robust and aligned task-relevant representations.

L DERIVATIONS

$$
\begin{aligned}
I(X;Z) &= \mathbb{E}_{X,Z}\left[\log \frac{p(x \mid z)}{p(x)}\right] \\
&= \mathbb{E}_{X,Z}\left[\log \prod_i \frac{p(x_i \mid \text{Patch}(x_i), z)}{p(x_i \mid \text{Patch}(x_i))}\right] = \mathbb{E}_{X,Z}\left[\sum_i \log \frac{p(x_i \mid \text{Patch}(x_i), z)}{p(x_i \mid \text{Patch}(x_i))}\right]
\end{aligned} \tag{13}
$$

M PGIB

The aforementioned contrastive loss approach used in PGIB is given as follows:

$$
\mathcal{L}_{\text{CONT}} := -\frac{1}{n}\sum_{i=1}^{n} \log \frac{\exp(g(z_{G_i}, z_{G_j})/\tau)}{\sum_{k:z_k \notin P_{\text{sub}}} \exp(g(z_{G_i}, z_{G_k})/\tau)}. \tag{14}
$$